



Universidade do Minho
Licenciatura em Ciências da Computação

Unidade Curricular de Bases de Dados

Ano Letivo de 2024/2025

SuperKarts Cup

Ivo Sousa	João Matos	Ricardo Cerqueira
a102935	a102528	a102878

janeiro de 2025

BD

Data de Recepção	
Responsável	
Avaliação	
Observações	

SuperKarts Cup

Ivo Sousa	João Matos	Ricardo Cerqueira
a102935	a102528	a102878

janeiro de 2025

Resumo

Ao longo do trabalho de desenvolvimento de um sistema de gestão de base de dados para o evento **SuperKarts Cup**, o principal objetivo foi melhorar a organização da competição, tornando-a mais segura, eficiente e capaz de prevenir problemas. Este esforço visou assegurar a satisfação tanto dos espectadores como dos principais intervenientes no evento, além de reforçar a imagem profissional da competição.

O projeto foi desenvolvido em várias etapas, começando pela definição do sistema, que consistiu na delimitação dos objetivos e funcionalidades principais. Seguiu-se o levantamento e análise de requisitos, fase em que foram identificadas as necessidades específicas do evento. Posteriormente, realizou-se a modelação conceptual e lógica, etapa dedicada à estruturação dos dados e à organização das informações de forma clara e funcional. Por fim, foi feita a implementação física, que envolveu a configuração e o desenvolvimento da base de dados em ambiente prático.

As ferramentas utilizadas neste processo foram o “BrModelo” para a modelação conceptual, permitindo a estruturação inicial dos dados e “MySQL Workbench” para a modelação lógica e a implementação física, garantindo um sistema robusto e funcional.

Área de Aplicação: Análise, planeamento, modelação e implementação de um sistema de gestão de base de dados;

Palavras-Chave: Sistema de Gestão de Base de Dados, Casos, Requisitos, Modelação Concetual, Modelação Lógica, Implementação Física, SuperKarts Cup, IJR;

Índice

Resumo	i
Índice	ii
Índice de Figuras	iv
Índice de Tabelas	vii
1. Definição do Sistema	1
1.1. Contextualização	1
1.2. Fundamentação	2
1.3. Objetivos	3
1.4. Viabilidade	3
1.5. Recursos a Utilizar	4
1.6. Equipa de Trabalho	5
1.7. Plano de execução do trabalho	5
2. Definição de Requisitos	8
2.1. Método de levantamento e de análise de requisitos adotado	8
2.2. Organização dos requisitos levantados	9
2.2.1. Requisitos de descrição	9
2.2.2. Requisitos de exploração	11
2.2.3. Requisitos de controlo	12
2.3. Análise e validação geral dos requisitos	13
3. Modelação Concetual	14
3.1. Apresentação da abordagem de modelação realizada	14
3.2. Identificação e caracterização das entidades	14
3.3. Identificação e caracterização dos relacionamentos	16
3.4. Identificação e caracterização da associação dos atributos com as entidades e relacionamentos	17
3.5. Apresentação e explicação do diagrama ER produzido	28
4. Modelação Lógica	30
4.1. Construção e validação do modelo de dados lógico	30
4.2. Normalização de Dados	30
4.3. Apresentação e explicação do modelo lógico produzido	31
4.4. Validação do modelo com interrogações do utilizador	38
5. Implementação Física	39

5.1. Apresentação do processo de criação da base de dados	39
5.2. Cálculo do espaço da base de dados	63
5.3. Realização do povoamento da base de dados	72
5.4. Tradução das interrogações do utilizador em SQL	81
5.5. Definição e caracterização das vistas de utilização em SQL	84
5.6. Definição dos perfis de utilização para cada utilizador da base de dados.	88
5.7. Indexação do sistema de dados	91
5.8. Procedimentos implementados	92
5.9. Plano de segurança e recuperação de dados	94
Lista de Siglas e Acrónimos	97

Índice de Figuras

Figura 1: Primeira parte do Diagrama de GANTT	7
Figura 2: Segunda Parte do Diagrama de GANTT	8
Figura 3: Ilustração do diagrama ER	28
Figura 4: Ilustração do Modelo Lógico	38
Figura 5: Instruções para a criação e utilização da base de dados	39
Figura 6: Tabela lógica do Evento	40
Figura 7: Implementação do Evento	40
Figura 8: Tabela lógica de Funcionarios	41
Figura 9: Implementação de Funcionarios	41
Figura 10: Tabela lógica de Espectadores	42
Figura 11: Implementação de Espectadores	42
Figura 12: Tabela lógica de Lembrancas	43
Figura 13: Implementação de Lembrancas	43
Figura 14: Tabela lógica da Venda	44
Figura 15: Implementação da Venda	44
Figura 16: Tabela lógica do Fornecedor	45
Figura 17: Implementação do Fornecedor	45
Figura 18: Tabela lógica da Encomenda	46
Figura 19: Implementação da Encomenda	46
Figura 20: Tabela lógica da Corrida	47
Figura 21: Implementação da Corrida	47
Figura 22: Tabela lógica de Equipas	48
Figura 23: Implementação de Equipas	48
Figura 24: Tabela lógica de Pilotos	49
Figura 25: Implementação de Pilotos	49
Figura 26: Tabela lógica da Classificacao	50
Figura 27: Implementação da Classificacao	50
Figura 28: Tabela lógica de Funcionarios da Equipa	51
Figura 29: Implementação de Funcionarios da Equipa	51
Figura 30: Tabela lógica da Protecao Civil	52
Figura 31: Implementação da Protecao Civil	53

Figura 32: Tabela lógica do Arbitro	54
Figura 33: Implementação do Arbitro	54
Figura 34: Tabela lógica de Comissarios	55
Figura 35: Implementação de Comissarios	56
Figura 36: Tabela lógica de Material de Corrida	56
Figura 37: Implementação de Material de Corrida	57
Figura 38: Tabela lógica de Patrocinadores	57
Figura 39: Implementação de Patrocinadores	58
Figura 40: Tabela lógica da Comunica	58
Figura 41: Implementação de Comunica	59
Figura 42: Tabela lógica de gereF	59
Figura 43: Implementação de gereF	59
Figura 44: Tabela lógica do Turno	60
Figura 45: Implementação do Turno	60
Figura 46: Tabela lógica de Notificacao	61
Figura 47: Implementação de Notificacao	61
Figura 48: Tabela lógica de Contacto_Espectador	61
Figura 49: Implementação de Contacto_Espectador	62
Figura 50: Tabela lógica de Oferecido	62
Figura 51: Implementação de Oferecido	62
Figura 52: Povoamento da tabela Evento	72
Figura 53: Povoamento da tabela Protecao Civil	73
Figura 54: Povoamento da tabela Corrida	73
Figura 55: Povoamento da tabela Arbitro	73
Figura 56: Povoamento da tabela Comissarios	74
Figura 57: Povoamento da tabela Funcionarios	74
Figura 58: Povoamento da tabela Turno	75
Figura 59: Povoamento da tabela Espectadores	75
Figura 60: Povoamento da tabela Contacto_Espectador	75
Figura 61: Povoamento da tabela Equipas	76
Figura 62: Povoamento da tabela Funcionarios da Equipa	76
Figura 63: Povoamento da tabela Pilotos	76
Figura 64: Povoamento da tabela Classificacao	77
Figura 65: Povoamento da tabela Material de Corrida	78
Figura 66: Povoamento da tabela Patrocinadores	78
Figura 67: Povoamento da tabela Lembrancas	78
Figura 68: Povoamento da tabela Venda	79
Figura 69: Povoamento da tabela Fornecedor	79
Figura 70: Povoamento da tabela Encomenda	79
Figura 71: Povoamento da tabela Oferecido	80

Figura 72: Povoamento da tabela Comunica	80
Figura 73: Povoamento da tabela gereF	81
Figura 74: Mostrar as equipas registadas para um evento específico	81
Figura 75: Mostrar o histórico de vendas realizadas por um funcionário	82
Figura 76: Listar todos os artigos em stock e suas respectivas quantidades de um evento	82
Figura 77: Mostrar os árbitros designados para uma prova específica de um evento	83
Figura 78: Listar todos os fornecedores com encomendas pendentes	84
Figura 79: Vista que mostra todas as informações de uma determinada equipa	85
Figura 80: Vista que mostra o funcionário responsável pela comunicação com uma determinada equipa	85
Figura 81: Vista que mostra toda a informação de uma encomenda	86
Figura 82: Vista que mostra toda a informação de um piloto	87
Figura 83: Vista que mostra a equipa vencedora de um determinado evento	88
Figura 84: Criação de utilizadores "Gerente" e devidas permissões	89
Figura 85: Criação de utilizadores "Funcionario" e devidas permissões	89
Figura 86: Criação de utilizadores "Arbitro" e devidas permissões	89
Figura 87: Criação de utilizadores "Comissario" e devidas permissões	90
Figura 88: Criação de utilizadores "Chefe de Equipa" e devidas permissões	90
Figura 89: Criação de utilizadores "Piloto" e devidas permissões	91
Figura 90: Criação de utilizadores "Funcionario de Equipa" e devidas permissões	91
Figura 91: Aplicação de indexação na SuperKarts Cup	92
Figura 92: Gatilho responsável por criar uma notificação quando uma encomenda é criada	92
Figura 93: Gatilho responsável por criar uma notificação quando uma encomenda é atualizada	93
Figura 94: Procedimento responsável por atribuir/atualizar os pontos das equipas	93

Índice de Tabelas

Tabela 1: Requisitos de descrição	11
Tabela 2: Requisitos de Manipulação	12
Tabela 3: Requisitos de controlo	13
Tabela 4: Caracterização das entidades	16
Tabela 5: Caracterização dos relacionamentos	17
Tabela 6: Caracterização dos atributos da entidade Evento	17
Tabela 7: Caracterização dos atributos da entidade Funcionários	18
Tabela 8: Caracterização dos atributos da entidade Proteção Cível	19
Tabela 9: Caracterização dos atributos da entidade Corrida	19
Tabela 10: Caracterização dos atributos da entidade Venda	20
Tabela 11: Caracterização dos atributos da entidade Espectadores	20
Tabela 12: Caracterização dos atributos da entidade Contacto_Espectadores	21
Tabela 13: Caracterização dos atributos da entidade Lembranças	21
Tabela 14: Caracterização dos atributos da entidade Encomenda	21
Tabela 15: Caracterização dos atributos da entidade Notificação	21
Tabela 16: Caracterização dos atributos da entidade Fornecedores	22
Tabela 17: Caracterização dos atributos da entidade Turno	22
Tabela 18: Caracterização dos atributos da entidade Equipa	23
Tabela 19: Caracterização dos atributos da entidade Pilotos	24
Tabela 20: Caracterização dos atributos da entidade Funcionários da Equipa	24
Tabela 21: Caracterização dos atributos da entidade Material de Corrida	25
Tabela 22: Caracterização dos atributos da entidade Patrocinadores	26
Tabela 23: Caracterização dos atributos da entidade Comissários	27
Tabela 24: Caracterização dos atributos da entidade Árbitro	27
Tabela 25: Caracterização dos atributos da entidade Classificação	28
Tabela 26: Caracterização da tabela lógica do Evento	31
Tabela 27: Caracterização da tabela lógica do Funcionarios	31
Tabela 28: Caracterização da tabela lógica do GereF	32
Tabela 29: Caracterização da tabela lógica do Venda	32
Tabela 30: Caracterização da tabela lógica do Lembrancas	32
Tabela 31: Caracterização da tabela lógica do Espectadores	33

Tabela 32: Caracterização da tabela lógica do Encomenda	33
Tabela 33: Caracterização da tabela lógica do Fornecedor	33
Tabela 34: Caracterização da tabela lógica do Comunica	33
Tabela 35: Caracterização da tabela lógica do Protecao Civil	34
Tabela 36: Caracterização da tabela lógica do Equipas	34
Tabela 37: Caracterização da tabela lógica do Pilotos	35
Tabela 38: Caracterização da tabela lógica do Funcionarios da Equipa	35
Tabela 39: Caracterização da tabela lógica do Corrida	35
Tabela 40: Caracterização da tabela lógica do Arbitro	36
Tabela 41: Caracterização da tabela lógica do Comissarios	36
Tabela 42: Caracterização da tabela lógica do Material de Corrida	37
Tabela 43: Caracterização da tabela lógica do Patrocinadores	37
Tabela 44: Caracterização da tabela lógica do participa	37
Tabela 45: Caracterização da tabela lógica do oferecido	37
Tabela 46: Caracterização da tabela lógica da Notificacao	38
Tabela 47: Tipos de dados e seus respectivos tamanhos em bytes	63
Tabela 48: Espaço ocupado por Arbitros com um registo	63
Tabela 49: Espaço ocupado por Comissarios com um registo	64
Tabela 50: Espaço ocupado por Comunica com um registo	64
Tabela 51: Espaço ocupado por Contacto_Espectador com um registo	64
Tabela 52: Espaço ocupado por Corrida com um registo	65
Tabela 53: Espaço ocupado por Encomenda com um registo	65
Tabela 54: Espaço ocupado por Equipa com um registo	65
Tabela 55: Espaço ocupado por Espectadores com um registo	66
Tabela 56: Espaço ocupado por Evento com um registo	66
Tabela 57: Espaço ocupado por Fornecedor com um registo	66
Tabela 58: Espaço ocupado por Funcionarios da Equipa com um registo	67
Tabela 59: Espaço ocupado por Funcionarios com um registo	67
Tabela 60: Espaço ocupado por gereF com um registo	67
Tabela 61: Espaço ocupado por Lembrancas com um registo	68
Tabela 62: Espaço ocupado por Material de Corrida com um registo	68
Tabela 63: Espaço ocupado por Notificacao com um registo	68
Tabela 64: Espaço ocupado por Oferecido com um registo	68
Tabela 65: Espaço ocupado por Patrocinadores com um registo	69
Tabela 66: Espaço ocupado por Pilotos com um registo	69
Tabela 67: Espaço ocupado por Protecao Civil com um registo	69
Tabela 68: Espaço ocupado por Turnos com um registo	70
Tabela 69: Espaço ocupado por Venda com um registo	70
Tabela 70: Espaço ocupado por Classificacao com um registo	70
Tabela 71: Espaço ocupado pelas tabelas após o povoamento inicial	71

1. Definição do Sistema

1.1. Contextualização

A **SuperKarts Cup** é um evento automotivo de superkarts, como o próprio nome indica, realizado em **Portimão, Portugal**. Este evento ocorre tradicionalmente na **primeira semana de outono** e, neste ano, decorreu de **22 a 29 de setembro**, com o objetivo de promover e expandir o mundo dos superkarts.

Fundada por **Roberto Serafim**, a SuperKarts Cup tornou-se uma importante atração turística, atingindo já a sua **21.ª edição**, tendo estreado a 29 de setembro de 2003.

Esta competição é considerada a mais completa do mundo no universo dos superkarts. Para além da clássica corrida no **Autódromo Internacional do Algarve**, conta com provas adicionais que tornam o evento único e diversificado. A programação é rica e bem estruturada:

- Na **segunda-feira**, realizam-se as voltas de treino;
- Na **terça-feira**, decorrem as voltas de qualificação;
- Na **quarta-feira**, acontece a corrida estilo clássico.
- **Quinta-feira** é reservada às emocionantes **Drag Races**, uma prova de 400 metros disputada no formato “play-off”. Nesta competição, composta por 7 corridas, as equipas vencedoras passam para a fase seguinte. Com 10 equipas participantes, os dois primeiros quartos de final são disputados entre 3 equipas, sendo estas sorteadas aleatoriamente.
- Na **sexta-feira** e no **sábado**, realiza-se a desafiante **corrida de resistência de 24 horas**, cujo objetivo é simples: a equipa que completar o maior número de voltas ao autódromo em 24 horas vence.
- No **domingo**, último dia do evento, acontece a prova de **Rally de superkarts**, onde cada equipa tem direito a uma volta de reconhecimento da pista durante a manhã. A prova começa às 14:00 e inicia-se no Autódromo Internacional do Algarve, terminando no centro de Portimão.

Cada edição da SuperKarts Cup conta com a participação de **10 equipas**, constituídas por **2 a 4 pilotos** e **1 a 5 elementos de apoio técnico**. Cada equipa só pode ter um superkart em pista em qualquer prova, e cada piloto está limitado a competir em **duas categorias por evento**. Por exemplo, um piloto que participe na corrida clássica e nas Drag Races não poderá competir noutras categorias.

A equipa vencedora do evento é aquela que acumular o maior número de pontos ao longo de todas as provas. A atribuição de pontos varia consoante a categoria:

- **Corrida estilo clássico:**
 - O vencedor recebe **9 pontos**, o segundo lugar recebe **8**, e assim sucessivamente até ao décimo classificado, que não pontua.
 - A equipa que registar a volta mais rápida recebe um **bónus um ponto**.
 - Equipas que não terminem a prova não recebem qualquer ponto.
- **Drag Races:**
 - O vencedor da prova recebe **9 pontos**, o finalista **6 pontos**, e os eliminados nas meias-finais **3 pontos**.
 - Equipas que não passam dos quartos de final não recebem pontos.
 - A equipa com o melhor tempo nas Drag Races recebe **um ponto adicional**.
- **Corrida de resistência de 24 horas:**
 - Os pontos são atribuídos da mesma forma que na corrida estilo clássico, incluindo o bónus de meio ponto para a volta mais rápida.
- **Rally:**
 - Os pontos seguem o mesmo critério da corrida clássica e da corrida de resistência, mas **não há pontos extra atribuídos** nesta prova.

Com este formato diversificado e desafiante, a SuperKarts Cup continua a ser um evento de referência no desporto automóvel, atraindo fãs e equipas de todo o mundo para Portimão.

1.2. Fundamentação

Ao longo dos anos, a SuperKarts Cup começou a enfrentar duras críticas pela sua forma antiquada de organização. Este cenário tornou-se motivo de preocupação para muitos, incluindo o próprio Sr. Roberto Serafim, que frequentemente se queixava da perda de fundos monetários associados às despesas das equipas, uma consequência direta da falta de eficiência e controlo na gestão do evento.

Reconhecendo a necessidade de modernizar a competição e melhorar a sua estrutura organizativa, o Sr. Roberto Serafim tomou a decisão de implementar um Sistema de Gestão de Base de Dados. Este sistema surgiu como uma solução indispensável para transformar a forma como o evento é gerido, assegurando maior precisão no controlo de recursos e na gestão financeira, enquanto eleva a SuperKarts Cup a um novo patamar de profissionalismo e inovação.

1.3. Objetivos

O Sistema de Gestão de Base de Dados, pedido pelo Sr. Roberto Serafim, deve ser capaz de:

- **Registrar as equipas presentes no evento de forma detalhada:** O sistema deve permitir o registo completo de cada equipa, incluindo informações sobre os pilotos, membros da equipa, e pontos adquiridos durante as provas. As atualizações devem ser automáticas após cada corrida, garantindo que a classificação seja precisa e em tempo real;
- **Monitorizar o stock de pneus e de combustível usado por cada equipa em cada prova:** O sistema deve ter um módulo específico para o controle de inventário, permitindo que as equipas registem a quantidade de pneus e combustível utilizados. Deve emitir alertas quando estiver no último jogo de pneus ou no último depósito de combustível;
- **Gerar relatórios detalhados sobre as vendas dos bilhetes e dos artigos de marketing:** O sistema deve incluir funcionalidades para registar e analisar as vendas de bilhetes e produtos de marketing, sendo possível gerar relatórios sobre o desempenho financeiro do evento. Estes relatórios devem ser acessíveis em diferentes formatos, desde gráficos a tabelas, para facilitar a visualização dos dados;
- **Facilitar a comunicação entre equipas e a administração do evento:** O sistema deve incluir uma forma de comunicação entre as equipas e a administração do evento, para que seja possível reportar problemas, pedir informações ou até mesmo esclarecer dúvidas durante a Superkarts Cup;
- **Implementar um sistema de “backup” e segurança de dados:** O sistema deve garantir que todos os dados sejam guardados de forma a respeitar os protocolos de segurança para proteger as informações sensíveis, com isto, evita-se perdas de informação e acessos não autorizados;
- **Fornecer suporte técnico durante o evento:** O sistema deve ter uma interface amigável e suporte técnico disponível para auxiliar os administradores e as equipas em caso de problemas técnicos durante a competição;
- **Gerir a logística da Superkarts Cup:** O sistema deve permitir o planeamento e a coordenação de aspetos logísticos, desde a horários de prova a prevenções da proteção civil, garantindo um funcionamento fluído de todas as atividades;

1.4. Viabilidade

O Sr. Roberto Serafim defende que, em termos de viabilidade financeira, a implementação de um **Sistema de Gestão de Base de Dados (SGBD)** representa uma solução significativamente mais eficiente do que a utilização de folhas de papel. Este sistema permitirá o registo detalhado de todos os movimentos relacionados com o evento, desde a gestão dos pneus utilizados até ao controlo do combustível consumido pelas equipas.

Com esta inovação, a **SuperKarts Cup** será capaz de alcançar resultados tangíveis, como:

- **Recuperar aproximadamente 10% do valor perdido** em edições anteriores devido a falhas no controlo e na gestão de recursos;
- **Gerir eficazmente os valores monetários** provenientes das inscrições das equipas, garantindo maior transparência e precisão.

Além dos benefícios financeiros, o **SGBD** contribui de forma relevante para a sustentabilidade ambiental da SuperKarts Cup. A utilização do sistema reduz o consumo desnecessário de pneus e de combustível pelas equipas participantes, promovendo práticas mais conscientes. Paralelamente, a eliminação do uso extensivo de folhas de papel na organização do evento reforça o compromisso com a preservação do meio ambiente, alinhando a competição com princípios modernos de sustentabilidade.

Esta abordagem, além de eficiente, posiciona a SuperKarts Cup como um exemplo de inovação e responsabilidade ambiental no sector desportivo.

1.5. Recursos a Utilizar

Em termos de recursos humanos, o evento **SuperKarts Cup** conta com a colaboração de diferentes grupos essenciais para o seu sucesso. Estes incluem o pessoal do **STAFF**, que assegura o bom funcionamento das diversas áreas do evento, as equipas participantes, que são a alma da competição, e os profissionais envolvidos no desenvolvimento do **Sistema de Gestão de Base de Dados**, responsáveis por garantir a eficiência tecnológica e organizacional.

Relativamente aos recursos materiais, destacam-se três categorias principais:

- **Hardware:** Inclui um servidor central e 10 pontos de controlo distribuídos estrategicamente, permitindo que cada equipa tenha a capacidade de atualizar a base de dados em tempo real, garantindo agilidade e precisão na gestão das informações.
- **Software:** O sistema de gestão de base de dados desempenha um papel crucial ao fornecer as ferramentas necessárias para a organização e manipulação eficiente dos dados.
- **Recursos documentais:** Este grupo engloba as especificações e requisitos previamente levantados, que servem como guias fundamentais para o desenvolvimento e a implementação do sistema, assegurando que todos os elementos do evento estejam alinhados com as necessidades identificadas.

Assim, a conjugação equilibrada destes recursos humanos e materiais é a base para o funcionamento harmonioso e eficaz da SuperKarts Cup, garantindo tanto a satisfação dos intervenientes como a excelência do evento.

1.6. Equipa de Trabalho

A equipa de trabalho é composta pelos profissionais da *IJR*, reconhecidos pela sua competência e elevado nível de profissionalismo em todas as tarefas que desempenham. A equipa é formada por Ivo Sousa, João Matos e Ricardo Cerqueira, que, em conjunto, foram responsáveis pelo levantamento de requisitos, pela modelação e pela implementação do sistema.

1.7. Plano de execução do trabalho

O projeto SuperKarts Cup foi cuidadosamente planeado e estruturado em quatro fases principais: Definição do Sistema, Definição de Requisitos, Modelação Conceptual e Modelação Lógica. Cada uma destas etapas foi desenhada para garantir um desenvolvimento consistente e alinhado com os objetivos propostos.

Na Definição do Sistema, foram estabelecidas as bases para o projeto, começando pela contextualização, conduzida por Ivo Sousa, entre 16 e 18 de setembro de 2024, onde se analisou o enquadramento do projeto e os seus fundamentos. Esta tarefa foi complementada pela fundamentação teórica e prática, também realizada por Ivo Sousa, no dia 18 de setembro de 2024. Posteriormente, Ricardo Cerqueira definiu os objetivos a serem alcançados, numa tarefa concluída entre 19 e 20 de setembro de 2024. A avaliação da viabilidade do projeto esteve a cargo de João Matos, entre 20 e 23 de setembro de 2024, onde foram analisados os recursos disponíveis e os desafios potenciais. Ricardo Cerqueira identificou os recursos necessários para o projeto, num trabalho realizado de 20 a 25 de setembro de 2024. A estruturação da equipa de trabalho foi concluída por João Matos em 26 de setembro de 2024, definindo papéis e responsabilidades. Finalmente, a elaboração de um plano de execução detalhado, realizada por toda a equipa, foi concluída entre 26 e 30 de setembro de 2024.

A segunda fase, Definição de Requisitos, teve como foco o levantamento e organização das necessidades do sistema. Entre 30 de setembro e 3 de outubro de 2024, a equipa, adotou um método sistemático para identificar os requisitos essenciais. A organização e estruturação dos requisitos levantados foi realizada por João Matos entre 3 e 10 de outubro de 2024. Paralelamente, foram documentados os requisitos de descrição e explorados os aspetos funcionais, sob responsabilidade de João Matos e Ricardo Cerqueira, respetivamente, no mesmo período. As necessidades relacionadas com o controlo do sistema foram identificadas por Ivo Sousa, também no intervalo de 3 a 10 de outubro de 2024. Após estas atividades, Ivo Sousa conduziu a análise e validação geral dos requisitos levantados, garantindo a sua consistência, numa tarefa finalizada entre 10 e 14 de outubro de 2024.

Seguindo para a Modelação Conceptual, o objetivo desta fase foi criar uma representação abstrata das entidades e relações do sistema. Entre 14 e 16 de outubro de 2024, a equipa definiu e apresentou a abordagem metodológica para a modelação. João Matos dedicou-se, entre 16 e 18 de outubro de 2024, à identificação e caracterização das entidades relevantes do sistema. Entre 18 e 20 de outubro de 2024, Ricardo Cerqueira mapeou e definiu as relações entre essas entidades. No período de 20 a 24 de outubro de 2024, Ivo Sousa especificou os atributos associados às entidades e relacionamentos. Por fim, *IJR* elaborou e explicou o diagrama entidade-relacionamento (ER), concluindo esta etapa entre 24 e 28 de outubro de 2024.

A última fase, Modelação Lógica, concentrou-se na tradução do modelo conceptual para um formato técnico adequado à implementação do sistema de gestão de bases de dados. Entre 28 de outubro e 3 de novembro de 2024, Ricardo Cerqueira liderou a construção e validação do modelo lógico. Seguindo esta tarefa, João Matos normalizou o modelo, eliminando redundâncias e melhorando a sua integridade, entre 3 e 7 de novembro de 2024. No mesmo período, Ivo Sousa apresentou e explicou o modelo lógico produzido, garantindo a compreensão por toda a equipa. A fase foi concluída com a validação do modelo, incluindo interações com utilizadores, conduzida por UR entre 9 e 11 de novembro de 2024.

Este cronograma detalhado reflete o rigor e empenho da equipa em todas as etapas do projeto, assegurando que os objetivos definidos foram alcançados dentro dos prazos estabelecidos. Cada tarefa foi cumprida com sucesso, garantindo a base sólida necessária para o desenvolvimento de um sistema robusto e eficiente.

Tarefa	Designado a :	Progresso	Começo	Fim
Definição do Sistema				
Contextualização	Ivo Sousa	100%	16/09/2024	18/09/2024
Fundamentação	Ivo Sousa	100%	16/09/2024	18/09/2024
Objetivos	Ricardo Cerqueira	100%	19/09/2024	20/09/2024
Viabilidade	João Matos	100%	20/09/2024	25/09/2024
Recursos a utilizar	Ricardo Cerqueira	100%	20/09/2024	25/09/2024
Equipa de trabalho	João Matos	100%	25/09/2024	26/09/2024
Plano de execução do trabalho	IJR	100%	26/09/2024	30/09/2024
Definição de Requisitos				
Método de levantamento e de análise de requisitos adotado	IJR	100%	30/09/2024	03/10/2024
Organização dos requisitos levantados	IJR	100%	03/10/2024	10/10/2024
Requisitos de descrição	João Matos	100%	03/10/2024	10/10/2024
Requisitos de exploração	Ricardo Cerqueira	100%	03/10/2024	10/10/2024
Requisitos de controlo	Ivo Sousa	100%	03/10/2024	10/10/2024
Análise e validação geral dos requisitos	IJR	100%	10/10/2024	14/10/2024
Modelação Concetual				
Apresentação da abordagem de modelação realizada	IJR	100%	14/10/2024	16/10/2024
Identificação e caracterização das entidades	João Matos	100%	16/10/2024	18/10/2024
Identificação e caracterização dos relacionamentos	Ricardo Cerqueira	100%	18/10/2024	20/10/2024
Identificação e caracterização da associação dos atributos com as entidades e relacionamentos	Ivo Sousa	100%	20/10/2024	24/10/2024
Apresentação e explicação do diagrama ER produzido	IJR	100%	24/10/2024	28/10/2024
Modelação Lógica				
Construção e validação do modelo de dados lógico	Ricardo Cerqueira	100%	28/10/2024	03/11/2024
Normalização de Dados	Ivo Sousa	100%	03/11/2024	07/11/2024
Apresentação e explicação do modelo lógico produzido	IJR	100%	07/11/2024	09/11/2024
Validação do modelo com interrogações do utilizador	IJR	100%	09/11/2024	11/11/2024

Figura 1: Primeira parte do Diagrama de GANTT

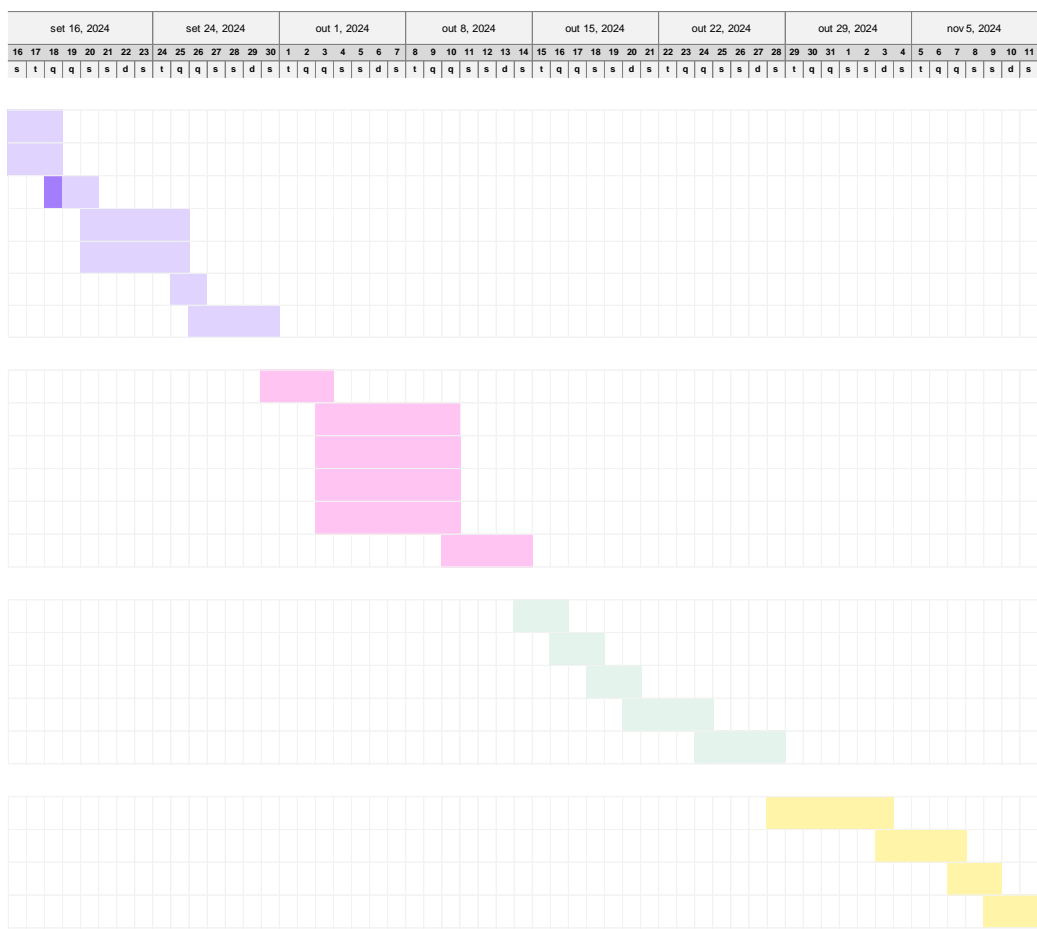


Figura 2: Segunda Parte do Diagrama de GANTT

2. Definição de Requisitos

Ao longo das diferentes etapas de desenvolvimento, tornou-se necessário proceder ao levantamento e à análise dos requisitos que sustentaram o sistema de gestão de bases de dados.

2.1. Método de levantamento e de análise de requisitos adotado

Para garantir um desenvolvimento eficiente do sistema de gestão da base de dados para a SuperKarts Cup, a *IJR*, equipa composta por Ivo Sousa, João Matos e Ricardo Eusébio dedicou-se a registar e documentar todas as condições e especificações do evento,

procurando a melhor forma de estruturar o modelo que abrange o problema identificado. Para tal, recorreram aos seguintes métodos de levantamento de requisitos:

- **Reuniões:** Foram realizadas reuniões com os principais responsáveis pelo evento e profissionais do *IJR*, com o intuito de compreender a forma mais adequada de gerir os dados relacionados com o evento.
- **Documentos:** A equipa reuniu diversos documentos, incluindo registos de eventos anteriores, bem como regulamentos e normas de segurança.

2.2. Organização dos requisitos levantados

Após a definição do método para o levantamento e análise de requisitos, e a respetiva aplicação, procedeu-se à organização e disposição tabelada da informação, conforme detalhado nos anexos deste relatório.

2.2.1. Requisitos de descrição

Nº	Data Hora	Descrição	Fonte	Analista
RD1	21/10/2024 15:34	O Evento tem um nome, uma data de início, um contacto telefónico, um email, uma localização onde se realiza e gastos	Documentacao	IJR
RD2	21/10/2024 15:34	A localização é uma morada constituída por rua, número de Porta, localidade e código postal	Documentacao	IJR
RD3	21/10/2024 15:34	O Evento tem vários funcionários	Documentacao	IJR
RD4	21/10/2024 15:34	O Funcionário do Evento é descrito por nome, número, data de nascimento, morada, contacto e tipo	Documentacao	IJR
RD5	21/10/2024 15:38	A Morada é constituída por rua, número de porta, localidade e código postal	Documentacao	IJR
RD6	21/10/2024 15:35	O Evento tem 10 Equipas	Documentacao	IJR

RD7	21/10/2024 15:35	As Equipas são caracterizadas pelo seu número, nome, elementos e contacto	Documentacao	IJR
RD8	21/10/2024 15:35	Os pilotos são caracterizados por número de pista, nome, data de nascimento e contacto	Documentacao	IJR
RD9	21/10/2024 15:35	Os funcionários das equipas são caracterizados por número, nome, data de nascimento, contacto e tipo	Documentacao	IJR
RD10	21/10/2024 15:35	As provas são distinguidas através do tipo, classificação e pista	Documentacao	IJR
RD11	21/10/2024 15:36	O material de corrida oferecido pelos patrocinadores do evento às equipas são caracterizados por nome, id e quantidade	Documentacao	IJR
RD12	21/10/2024 15:36	Os patrocinadores do evento são caracterizados por número, nome, morada, contacto, quantia e tipo de material	Documentacao	IJR
RD13	21/10/2024 15:36	O Evento necessita de ter presente a proteção civil para ocorrer, sendo ela identificada por número do elemento, nome, tipo, contacto	Documentacao	IJR
RD14	21/10/2024 15:36	O Evento realiza vendas de artigos turísticos sendo que essas vendas são definidas por funcionário que vendeu, data, valor e artigo	Documentacao	IJR
RD15	21/10/2024 15:36	As lembranças são definidas através de um nome, id, quantidade e preço	Documentacao	IJR
RD16	21/10/2024 15:36	As lembranças têm um fornecedor	Documentacao	IJR
RD17	21/10/2024 15:37	Os fornecedores são definidos por id, nome, morada, contacto, estado	Documentacao	IJR

RD18	21/10/2024 15:40	Os fornecedores recebem encomendas	Documentacao	IJR
RD19	21/10/2024 15:41	Os espectadores na compra do bilhete ficam registados com um número, nome, data de nascimento, contacto e morada	Documentacao	IJR
RD20	21/10/2024 15:41	Cada espectador pode estar associado a vários contactos	Documentacao	IJR
RD21	21/10/2024 15:43	As provas necessitam de árbitros	Documentacao	IJR
RD22	21/10/2024 15:43	As provas necessitam de comissários	Documentacao	IJR
RD23	21/10/2024 15:43	O Árbitro é descrito por nome, número, data de nascimento, contacto e morada	Documentacao	IJR
RD24	21/10/2024 15:43	O comissário é descrito por nome, número, data de nascimento, morada, contacto e tipo	Documentacao	IJR
RD25	21/10/2024 15:43	O Evento organiza as corridas que são caracterizadas por tipo, classificação, pista	Documentacao	IJR
RD26	21/10/2024 15:44	As vendas ficam registadas com um número, valor e data, funcionário e cliente	Documentacao	IJR

Tabela 1: Requisitos de descrição

2.2.2. Requisitos de exploração

Nº	Data Hora	Descrição	Fonte	Analista
RM1	21/10/2024 15:39	Deve ser possível remover e adicionar os artigos de venda	Observação do Evento	IJR
RM2	21/10/2024 15:39	Deve ser possível visualizar o histórico das vendas	Entrevista Sr. Roberto	IJR

RM3	21/10/2024 15:40	Deve ser possível gerir funcionários	Entrevista Sr. Roberto	IJR
RM4	21/10/2024 15:40	Deve ser possível gerir o stock dos artigos	Entrevista Funcionário	IJR
RM5	21/10/2024 15:40	Quando o stock de pneus e combustível está no mínimo deve ser gerado um alerta para a equipa	Observação do Evento	IJR
RM6	21/10/2024 15:40	Funcionários deviam poder visualizar suas informações pessoais e turnos	Entrevista Funcionário	IJR
RM7	21/10/2024 15:41	Gerir as encomendas de artigos, pneus e combustível feitas aos fornecedores	Entrevista Funcionário	IJR
RM8	21/10/2024 15:41	Registar a data de entrega das encomendas	Entrevista Sr. Roberto	IJR
RM9	21/10/2024 15:42	Deve ser possível as equipas comunicarem com o STAFF do Evento	Entrevistas as Equipas	IJR

Tabela 2: Requisitos de Manipulação

2.2.3. Requisitos de controlo

Nº	Data Hora	Descrição	Fonte	Analista
RC1	21/10/2024 15:39	Apenas o Sr. Roberto pode gerir os artigos de venda	Entrevista Sr. Roberto	IJR
RC2	21/10/2024 15:39	A base de dados encontra-se operacional 24/7	Observação do Evento	IJR
RC3	21/10/2024 15:39	Apenas o gerente do Evento pode adicionar e remover funcionários	Entrevista Sr. Roberto	IJR
RC4	21/10/2024 15:40	Registo de turnos e horários de cada funcionário	Entrevista Sr. Roberto	IJR
RC5	21/10/2024 15:41	Apenas o gerente pode adicionar, editar e remover os fornecedores	Entrevista Sr. Roberto	IJR

RC6	21/10/2024 15:42	Notificações sobre atrasos ou encomendas pendentes	Entrevista Sr. Roberto	IJR
-----	---------------------	--	---------------------------	-----

Tabela 3: Requisitos de controlo

2.3. Análise e validação geral dos requisitos

Após a conclusão do levantamento e da organização dos requisitos, foi realizada uma reunião final para identificar possíveis discordâncias. Com todos os pontos aprovados de forma unânime, a equipa confirmou que os requisitos estavam em total conformidade com as expectativas do Sr. Roberto. Assim, foi possível garantir as condições necessárias para a futura implementação de um sistema de gestão de bases de dados e avançar para o desenvolvimento do modelo conceptual.

3. Modelação Concetual

3.1. Apresentação da abordagem de modelação realizada

Apos uma pequena discussão sobre qual ferramenta se iria usar para o desenvolvimento do modelo concetual, os elementos do *IJR* chegaram a um acordo e decidiram utilizar o “BrModelo”, devido a sua simplicidade, pois o simples é bonito.

Para se proceder à modelação concetual é necessário, numa fase inicial, identificar todas as entidades únicas e relevantes destacadas pelos requisitos levantados, os relacionamentos entre as mesmas e os atributos tanto das entidades, como, possivelmente, dos relacionamentos. Identificadas estas componentes, será então possível proceder às identificações e caracterizações detalhadas de cada uma destas.

3.2. Identificação e caracterização das entidades

Designação	Descrição	Sinónimos	Ocorrências
Evento	Entidade que representa a organização	Competição	Um evento é responsável por gerar tudo
Funcionários	Entidade que representa os trabalhadores do Evento	Trabalhadores	Um funcionário desempenha 1 ou mais funções e gere 1 ou mais funcionários. Também comunicam com as equipas
Venda	Entidade que representa a transação das lembranças	Transação	Uma venda é realizada por um funcionário, por uma lembrança e por um espectador
Turno	entidade que representa o turno de trabalho dos funcionários	Trabalho	Um turno é desempenhado por um 1 ou mais funcionários

Espectadores	entidade que representa os fãs que estão presentes no evento	Fã	Um espectador desempenha 1 função
Contacto espectadores	Entidade que represente os vários contactos que cada espectador pode ter	Contacto	Cada espectador pode ter 1 ou mais contactos
Lembranças	entidade que representa as bugigangas para os espectadores	<i>Souvenires</i>	Cada lembrança relaciona se com um espectador, um funcionário e com a encomenda
Encomenda	Entidade que representa a transição das lembranças com os fornecedores	Compra	Cada encomenda é feita por um funcionário, e precisa de um fornecedor e de uma lembrança
Notificação	entidade responsável por avisar como está a encomenda	Aviso	Cada notificação depende da encomenda
Fornecedores	Entidade que representa a organização responsável por fornecer as lembranças ao evento	Abastecedor	Cada fornecedor fornece uma lembrança por encomenda
Proteção civil	Entidade que representa os bombeiros e GNR responsáveis por assegurar o evento	Bombeiros/GNR	A proteção civil assegura o evento
Corrida	Entidade que representa as várias corridas organizadas pelo evento	Prova	As corridas são organizadas pelo evento, corridas pelas equipas e asseguradas pelos árbitros e comissários
Equipa	Entidade que representa cada equipa que corre nas corridas	Grupo	As equipas para além de participarem nas corridas são constituídas pelos pilotos e seus respetivos funcionários. Também comunicam com os funcionários do evento
Pilotos	Entidade que representa os pilotos de cada equipa	Corredores	Os pilotos pertencem suas respetivas equipas
Funcionários da equipa	Entidade que representa cada funcionário de cada equipa	Trabalhadores	Os funcionários da equipa pertencem suas respetivas equipas
Material de Corrida	Entidade que representa os pneus e o combustível fornecido pelo evento	Pneus/Combustível	O material de corrida oferecidos as equipas têm um patrocinador

Patrocinadores	Entidade que representa quem fornecer o material de corrida	Patrono	Patrocina o material para as equipas
Comissários	Entidade que representa todos os comissários que asseguram cada corrida	<i>Marshall</i>	Asseguram as corridas
Arbitro	Entidade que representa todos os árbitros que asseguram cada corrida	Juiz	Ajuízam as corridas
Classificação	Entidade responsável pela classificação de cada corrida	Posição	A Classificação é responsável por atribuir pontos aos pilotos pela posição que ficou na corrida

Tabela 4:Caracterização das entidades

3.3. Identificação e caracterização dos relacionamentos

Entidade	Multiplicidade	Relacionamento	Multiplicidade	Entidade(relacionada)
Evento	1	contrata	N	Funcionários
Evento	1	possui	N	Lembranças
Evento	1	assegurado	N	Proteção civil
Evento	1	organiza	N	Corrida
Funcionários	1	Faz	1	Turno
Funcionários	N	gere	N	Funcionários
Funcionários	N	comunica	N	equipa
Funcionários	1	encomenda	N	Lembranças
Funcionários	1	encomenda	1	Notificação
Funcionários	1	encomenda	N	Fornecedores
Funcionários	N	venda	N	Lembranças
Funcionários	N	venda	N	espectadores
Espectadores	1	tem	N	contacto espectadores
Corrida	1	necessita	N	Classificação
Classificação	N	classifica	1	Pilotos
Corrida	1	ajuizada	N	arbitro
Corrida	1	assegurada	N	comissários
Equipa	1	precisa	N	material de corrida
Equipa	1	possui	N	pilotos
Equipa	1	possui	N	Funcionário de Equipa

Material de corrida	N	oferecido	N	Patrocinadores
---------------------	---	-----------	---	----------------

Tabela 5:Caracterização dos relacionamentos

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

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave-primária
EVENTO	Id	Identificador único do evento	Não	Não	Não	Não	Sim
	Nome	Nome do evento	Não	Não	Não	Não	Não
	Gastos	Gastos do evento	Não	Não	Não	Não	Não
	Data_Inicio	Data de início do evento	Não	Não	Não	Não	Não
	Morada_rua	Rua do evento	Não	Não	Não	Não	Não
	Morada_Localidade	Localidade do evento	Não	Não	Não	Não	Não
	Morada_Porta	Número da porta do evento	Não	Não	Não	Não	Não
	Morada_CodPostal	Código postal do evento	Não	Não	Não	Não	Não
	Contacto_email	E-mail atribuído ao evento	Não	Não	Não	Não	Não
	Contacto_telemovei	Número de telemóvel atribuído ao evento	Não	Não	Não	Não	Não

Tabela 6: Caracterização dos atributos da entidade Evento

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave-primária
FUNCIONÁRIOS	Id	Identificador único do funcionário	Não	Não	Não	Não	Sim
	Nome	Nome do funcionário	Não	Não	Não	Não	Não
	Data_de_Nascimento	Data de nascimento do funcionário	Não	Não	Não	Não	Não
	Tipo	Tipo de funcionário	Não	Não	Não	Não	Não
	Morada_ rua	Rua do funcionário	Não	Não	Não	Não	Não
	Morada_Localidade	Localidade do funcionário	Não	Não	Não	Não	Não
	Morada_Porta	Número da porta do funcionário	Não	Não	Não	Não	Não
	Morada_CodPostal	Código postal do funcionário	Não	Não	Não	Não	Não
	Contacto_email	E-mail atribuído ao funcionário	Não	Não	Não	Não	Não
	Contacto_telemovei	Número de telemóvel atribuído ao funcionário	Não	Não	Não	Não	Não

Tabela 7: Caracterização dos atributos da entidade Funcionários

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave-primária
PROTEÇÃO CIVIL	Id	Identificador único da proteção civil	Não	Não	Não	Não	Sim
	Nome	Nome da proteção civil	Não	Não	Não	Não	Não
	Tipo	Tipo de proteção civil	Não	Não	Não	Não	Não
	Morada_ rua	Rua da proteção civil	Não	Não	Não	Não	Não
	Morada_Localidade	Localidade da proteção civil	Não	Não	Não	Não	Não
	Morada_Porta	Número da porta da proteção civil	Não	Não	Não	Não	Não
	Morada_CodPostal	Código postal da proteção civil	Não	Não	Não	Não	Não
	Contacto_email	E-mail atribuído à proteção civil	Não	Não	Não	Não	Não
	Contacto_telemovei	Número de telemóvel atribuído à proteção civil	Não	Não	Não	Não	Não

Tabela 8: Caracterização dos atributos da entidade Proteção Civil

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave-primária
CORRIDA	Tipo	Tipo de corrida	Não	Não	Não	Não	Sim
	Pista	Pista da corrida	Não	Não	Não	Não	Não

Tabela 9: Caracterização dos atributos da entidade Corrida

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave-primária
VENDA	Id	Identificador único da venda	Não	Não	Não	Não	Sim
	Valor	Valor da venda	Não	Não	Não	Não	Não
	Data	Data da Venda	Não	Não	Não	Não	Não

Tabela 10: Caracterização dos atributos da entidade Venda

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave-primária
ESPECTADORES	Id	Identificador único do espectador	Não	Não	Não	Não	Sim
	Nome	Nome do espectador	Não	Não	Não	Não	Não
	Data_nascimento	Data nascimento do espectador	Não	Não	Não	Não	Não
	Morada_rua	Rua do espectador	Não	Não	Não	Não	Não
	Morada_Localidade	Localidade do espectador	Não	Não	Não	Não	Não
	Morada_Porta	Número da porta do espectador	Não	Não	Não	Não	Não
	Morada_CodPostal	Código postal do espectador	Não	Não	Não	Não	Não

Tabela 11: Caracterização dos atributos da entidade Espectadores

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave-primária
CONTACTO_E SPECTADOR	Contacto	Contacto do espectador	Não	Não	Não	Não	Sim

Tabela 12: Caracterização dos atributos da entidade Contacto_Espectadores

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave-primária
LEMBRANÇAS	Id	Identificador único da lembrança	Não	Não	Não	Não	Sim
	Nome	Nome da lembrança	Não	Não	Não	Não	Não
	Quantidade	Quantidade de lembranças	Não	Não	Não	Não	Não
	Preço	Preço das lembranças	Não	Não	Não	Não	Não

Tabela 13: Caracterização dos atributos da entidade Lembranças

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave-primária
ENCOMENDA	Id	Identificador único da encomenda	Não	Não	Não	Não	Sim
	Estado	Estado da encomenda	Não	Não	Não	Não	Não
	Quantidade	Quantidade de encomendas	Não	Não	Não	Não	Não
	Valor	Valor da encomenda	Não	Não	Não	Não	Não

Tabela 14: Caracterização dos atributos da entidade Encomenda

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave-primária
NOTIFICAÇÃO	Id	Identificador único da Notificação	Não	Não	Não	Não	Sim
	Descrição	Descrição da Notificação	Não	Não	Não	Não	Não
	Data	Data da Notificação	Não	Não	Não	Não	Não

Tabela 15: Caracterização dos atributos da entidade Notificação

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave-primária
FORNECEDORES	Id	Identificador único do fornecedor	Não	Não	Não	Não	Sim
	Nome	Nome do fornecedor	Não	Não	Não	Não	Não
	Tipo_Lembrança	Data nascimento do fornecedor	Não	Não	Não	Não	Não
	Morada_rua	Rua do fornecedor	Não	Não	Não	Não	Não
	Morada_Localidade	Localidade do fornecedor	Não	Não	Não	Não	Não
	Morada_Porta	Número da porta do fornecedor	Não	Não	Não	Não	Não
	Morada_CodPostal	Código postal do fornecedor	Não	Não	Não	Não	Não
	Contacto_telemovel	Número de telemóvel do fornecedor	Não	Não	Não	Não	Não
	Contacto_email	E-mail do fornecedor	Não	Não	Não	Não	Não

Tabela 16: Caracterização dos atributos da entidade Fornecedores

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave-primária
TURNO	Id	Identificador único do Turno	Não	Não	Não	Não	Sim
	Horário_inicio	Horário de início do Funcionário	Não	Não	Não	Não	Não
	Horário_fim	Horário de fim do Funcionário	Não	Não	Não	Não	Não

Tabela 17: Caracterização dos atributos da entidade Turno

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave-primária
EQUIPA	Id	Identificador único da equipa	Não	Não	Não	Não	Sim
	Nome	Nome da equipa	Não	Não	Não	Não	Não
	Morada_ rua	Rua da equipa	Não	Não	Não	Não	Não
	Morada_Localidade	Localidade da equipa	Não	Não	Não	Não	Não
	Morada_Porta	Número da porta da equipa	Não	Não	Não	Não	Não
	Morada_CodPostal	Código postal da equipa	Não	Não	Não	Não	Não
	Contacto_email	E-mail atribuída à equipa	Não	Não	Não	Não	Não
	Contacto_telemovel	Número de telemóvel atribuída à equipa	Não	Não	Não	Não	Não

Tabela 18: Caracterização dos atributos da entidade Equipa

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave-primária
PILOTOS	Número de pista	Número de pista do piloto	Não	Não	Não	Não	Sim
	Nome	Nome do piloto	Não	Não	Não	Não	Não
	Data_de_Nascimento	Data de nascimento do piloto	Sim	Não	Não	Não	Não
	Morada_ rua	Rua do piloto	Não	Não	Não	Não	Não
	Morada_Localidade	Localidade do piloto	Não	Não	Não	Não	Não
	Morada_Porta	Número da porta do piloto	Não	Não	Não	Não	Não
	Morada_CodPostal	Código postal do piloto	Não	Não	Não	Não	Não
	Contacto_email	E-mail atribuído ao piloto	Sim	Não	Não	Não	Não

	Contacto_telemovei	Número de telemóvel atribuído ao piloto	Não	Não	Não	Não	Não
--	--------------------	---	-----	-----	-----	-----	-----

Tabela 19: Caracterização dos atributos da entidade Pilotos

Entidade	Atributo	Descrição	Nulo	Composto	Multivalora- do	Derivado	Chave- primária
FUNCIONÁRIOS DE EQUIPA	Id	Identificador único do funcionário	Não	Não	Não	Não	Sim
	Nome	Nome do funcionário	Não	Não	Não	Não	Não
	Data_de_Nascimento	Data de nascimento do funcionário	Sim	Não	Não	Não	Não
	Tipo	Tipo de funcionário	Não	Não	Não	Não	Não
	Morada_rua	Rua do funcionário	Não	Não	Não	Não	Não
	Morada_Localidade	Localidade do funcionário	Não	Não	Não	Não	Não
	Morada_Porta	Número da porta do funcionário	Não	Não	Não	Não	Não
	Morada_CodPostal	Código postal do funcionário	Não	Não	Não	Não	Não
	Contacto_email	E-mail atribuído ao funcionário	Sim	Não	Não	Não	Não
	Contacto_telemovei	Número de telemóvel atribuído ao funcionário	Não	Não	Não	Não	Não

Tabela 20: Caracterização dos atributos da entidade Funcionários da Equipa

Entidade	Atributo	Descrição	Nulo	Composto	Multivalora- do	Derivado	Chave- primária
MATERIAL DE CORRIDA	Id	Identificador único do material de corrida	Não	Não	Não	Não	Sim
	Nome	Nome do material de corrida	Não	Não	Não	Não	Não
	Quantidade	Quantidade do material de corrida	Não	Não	Não	Não	Não
	Stock_minimo	Stock mínimo do material de corrida	Não	Não	Não	Não	Não

Tabela 21: Caracterização dos atributos da entidade Material de Corrida

Entidade	Atributo	Descrição	Nulo	Composto	Multivalora- do	Derivado	Chave- primária
PATROCINADORES	Id	Identificador único do patrocinador	Não	Não	Não	Não	Sim
	Nome	Nome do patrocinador	Não	Não	Não	Não	Não
	Quantidade	Quantidade de patrocinadores	Não	Não	Não	Não	Não
	Tipo_material	Tipo de material do patrocinador	Não	Não	Não	Não	Não
	Morada_ rua	Rua do patrocinador	Não	Não	Não	Não	Não
	Morada_Localidade	Localidade do patrocinador	Não	Não	Não	Não	Não

	Morada_Porta	Número da porta do patrocinador	Não	Não	Não	Não	Não
	Morada_CodPostal	Código postal do patrocinador	Não	Não	Não	Não	Não
	Contacto_email	E-mail atribuído ao patrocinador	Não	Não	Não	Não	Não
	Contacto_telemovei	Número de telemóvel atribuído ao patrocinador	Não	Não	Não	Não	Não

Tabela 22: Caracterização dos atributos da entidade Patrocinadores

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave-primária
COMISSÁRIOS	Id	Identificador único do comissário	Não	Não	Não	Não	Sim
	Nome	Nome do comissário	Não	Não	Não	Não	Não
	Data_de_Nascimento	Data de nascimento do comissário	Sim	Não	Não	Não	Não
	Tipo	Tipo de comissário	Não	Não	Não	Não	Não
	Morada_rua	Rua do comissário	Não	Não	Não	Não	Não
	Morada_Localidade	Localidade do comissário	Não	Não	Não	Não	Não
	Morada_Porta	Número da porta do comissário	Não	Não	Não	Não	Não
	Morada_CodPostal	Código postal do comissário	Não	Não	Não	Não	Não

	Contacto_email	E-mail atribuído ao comissário	Não	Não	Não	Não	Não
	Contacto_telemovei	Número de telemóvel atribuído ao comissário	Não	Não	Não	Não	Não

Tabela 23: Caracterização dos atributos da entidade Comissários

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave-primária
ÁRBITRO	Id	Identificador único do árbitro	Não	Não	Não	Não	Sim
	Nome	Nome do árbitro	Não	Não	Não	Não	Não
	Data_de_Nascimento	Data de nascimento do árbitro	Sim	Não	Não	Não	Não
	Morada_ rua	Rua do árbitro	Não	Não	Não	Não	Não
	Morada_Localidade	Localidade do árbitro	Não	Não	Não	Não	Não
	Morada_Porta	Número da porta do árbitro	Não	Não	Não	Não	Não
	Morada_CodPostal	Código postal do árbitro	Não	Não	Não	Não	Não
	Contacto_email	E-mail atribuído ao árbitro	Não	Não	Não	Não	Não
	Contacto_telemovei	Número de telemóvel atribuído ao árbitro	Não	Não	Não	Não	Não

Tabela 24: Caracterização dos atributos da entidade Árbitro

Entidade	Atributo	Descrição	Nulo	Composto	Multivalorado	Derivado	Chave-primária
CLASSIFICACAO	Posição	Lugar do piloto no fim da prova	Não	Não	Não	Não	Sim
	Pontos	Pontos que cada posição recebe no fim da prova	Não	Não	Não	Não	Não

Tabela 25: Caracterização dos atributos da entidade Classificação

3.5. Apresentação e explicação do diagrama ER produzido

No fim de identificarmos e caracterizarmos detalhadamente cada entidade, atributo e relacionamento nos tópicos anteriores, avançamos agora para a apresentação do diagrama ER completo, que serve de suporte ao modelo da base de dados em análise.

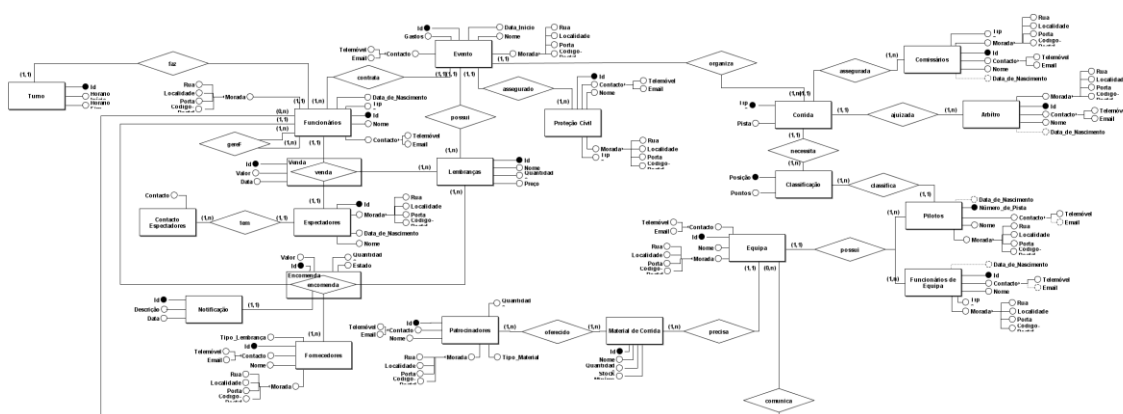


Figura 3: Ilustração do diagrama ER

Neste diagrama, é possível identificar de forma clara todas as entidades da base de dados, bem como os relacionamentos responsáveis por interligar as mesmas. Deste modo, o diagrama é interpretado da seguinte forma: o Evento organiza corridas, nas quais participam as equipes. As equipes são constituídas por pilotos e funcionários da equipe, além disso, a equipe recebe

Material de Corrida fornecido pelo patrocinador. A corrida é assegurada pelos comissários e ajuizada pelos árbitros. O Evento é assegurado pela Proteção Cível. Os funcionários do evento trabalham por turnos e podem realizar três tarefas, venda de lembranças, encomenda de lembranças e comunicação com as equipas. As lembranças vendidas aos espectadores, quando encomendadas têm um fornecedor. A cada atualização da encomenda é imitada uma notificação.

É possível identificar, em cada entidade e em alguns relacionamentos, os atributos que os definem. Em todas as entidades destacam-se os atributos que constituem uma chave primária da mesma, visto serem representados por um círculo totalmente preenchido. Para além disso é possível observar a presença de atributos opcionais.

Dado esta fase por terminado, é possível passar para a construção do modelo lógico.

4. Modelação Lógica

4.1. Construção e validação do modelo de dados lógico

Após a conclusão e revisão da modelação concetual, avançou-se para a elaboração do Modelo de Dados Lógico. Este modelo detalha os elementos de dados e oferece uma visão clara e estruturada das entidades, atributos, chaves e relacionamentos, proporcionando uma base sólida para a compreensão do sistema.

Para a criação deste modelo a equipa *IJR* optou por utilizar a ferramenta MySQL WorkBench, pois ele permite-nos criar representações precisas dos dados e das suas inter-relações, o que facilita bastante o desenvolvimento de sistemas de informação robustos e eficazes.

4.2. Normalização de Dados

A normalização de dados é um processo importantíssimo na execução de um sistema de gestão de base de dados. O objetivo principal da normalização é evitar a redundância de dados, ou seja, minimizar a duplicação de informação, minimizar ambiguidade e assegurar consistência.

Esta normalização é feita através de um conjunto de regras, aplicadas em etapas sucessivas, chamadas de formas normais, a cada etapa é abordada diferentes tipos de problemas de difusão e dependências funcionais entre as tabelas.

A Primeira Forma Normal elimina grupos repetitivos, ou seja, garante que todas as colunas de uma tabela contêm apenas valores atômicos e proíbe grupos repetitivos numa linha, ou seja, não podem existir múltiplos valores numa única célula.

A Segunda Forma Normal remove as dependências parciais, o que significa que todos os atributos não chave devem depender completamente da chave primária e não apenas de parte dela (se for uma chave composta)

Na Terceira Forma Normal as dependências transitivas são eliminadas, ou seja, não existe atributos que dependam de outros que não seja chave primaria da tabela em que se encontram definidos.

Ao verificar estas três formas normais, confirmar-se a validade do modelo lógico em produção. Complementarmente, existem mais fórmulas como a Forma Normal de Boyce-Codd (BCNF) ou Quarta e Quinta Formas Normais que impõem uma especificidade e detalhe superior na minimização de redundâncias e irregularidades, no entanto, descartámo-las dado a complexidade excessiva no âmbito do nosso desenvolvimento.

4.3. Apresentação e explicação do modelo lógico produzido

Para a construção do Modelo Lógico de Dados, os profissionais da *IJR* iniciaram a produção de 22 tabelas:

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Evento	idEvento	INT	-
	Gastos	INT	-
	Nome	VARCHAR(45)	-
	Data_Inicio	DATE	-
	morada_ rua	VARCHAR(45)	-
	morada_localidade	VARCHAR(45)	-
	morada_porta	INT	-
	morada_codigopostal	VARCHAR(45)	-
	contacto_telemovei	INT	-
	contacto_email	VARCHAR(45)	-

Tabela 26: Caracterização da tabela lógica do Evento

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Funcionarios	idFuncionarios	INT	-
	Evento _idEvento	INT	idEvento(Evento)
	Nome	VARCHAR(45)	-
	data_nasc	DATE	-
	morada_ rua	VARCHAR(45)	-
	morada_localidade	VARCHAR(45)	-
	morada_porta	INT	-
	morada_codigopostal	VARCHAR(45)	-
	contacto_telemovei	INT	-
	contacto_email	VARCHAR(45)	-

Tabela 27: Caracterização da tabela lógica do Funcionarios

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
GereF	Funcionarios_idFuncionarioGere	INT	idFuncionarios(Funcionarios)
	Funcionarios_idFuncionarios	INT	idFuncionarios(Funcionarios)

Tabela 28: Caracterização da tabela lógica do GereF

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Venda	idVenda	INT	-
	Data	DATETIME	-
	Valor	DECIMAL(8,2)	-
	funcionario	INT	idFuncionarios(Funcionarios)
	lembrancas	INT	idLembrancas(Lembrancas)
	espetadores	INT	idEspectadores(Espectadores)

Tabela 29: Caracterização da tabela lógica do Venda

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Lembrancas	idLembrancas	INT	-
	Evento_idEvento	INT	idEvento(Evento)
	Nome	VARCHAR(45)	-
	Quantidade	INT	-
	Preço	DECIMAL(5,2)	-

Tabela 30: Caracterização da tabela lógica do Lembrancas

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Espectadores	idEspectadores	INT	-
	Nome	VARCHAR(45)	-
	data_nasc	DATE	-
	morada_ rua	VARCHAR(45)	-
	morada_localidade	VARCHAR(45)	-
	morada_porta	INT	-

	morada_codigopostal	VARCHAR(45)	-
	contacto_telemovel	INT	-
	contacto_email	VARCHAR(45)	-

Tabela 31: Caracterização da tabela lógica do Espectadores

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Encomenda	idEncomenda	INT	-
	Valor	DECIMAL(5,2)	-
	Quantidade	DATE	-
	Estado	TINYINT	-
	idfuncionario	INT	"idFuncionario"(Funcionarios)
	idfornecedor	INT	"idFornecedor"(Fornecedor)
	idlembanca	INT	"idLembranca"(Lembrancas)

Tabela 32: Caracterização da tabela lógica do Encomenda

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Fornecedor	idFornecedor	INT	-
	Nome	VARCHAR(45)	-
	tipo_lembanca	VARCHAR(45)	-
	morada_rua	VARCHAR(45)	-
	morada_localidade	VARCHAR(45)	-
	morada_porta	INT	-
	morada_codigopostal	VARCHAR(45)	-
	contacto_telemovel	INT	-
	contacto_email	VARCHAR(45)	-

Tabela 33: Caracterização da tabela lógica do Fornecedor

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Comunica	Funcionarios_idFuncionarios	INT	idFuncionarios(Funcionarios)
	Equipas_idEquipas	INT	idEquipas(Equipas)

Tabela 34: Caracterização da tabela lógica do Comunica

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Protecao Civil	idProtecaoCivil	INT	-
	Nome	VARCHAR(45)	-
	tipo	ENUM('Bombeiros', 'GNR')	-
	morada_rua	VARCHAR(45)	-
	morada_localidade	VARCHAR(45)	-
	morada_porta	INT	-
	morada_codigopostal	VARCHAR(45)	-
	contacto_telemovel	INT	-
	contacto_email	VARCHAR(45)	-
	Evento_idEvento	INT	idEvento(Evento)

Tabela 35: Caracterização da tabela lógica do Protecao Civil

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Equipas	idEquipas	INT	-
	Nome	VARCHAR(45)	-
	morada_rua	VARCHAR(45)	-
	morada_localidade	VARCHAR(45)	-
	morada_porta	INT	-
	morada_codigopostal	VARCHAR(45)	-
	contacto_telemovel	INT	-
	contacto_email	VARCHAR(45)	-
	Pontos	INT	-

Tabela 36: Caracterização da tabela lógica do Equipas

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Pilotos	numero_Piloto	INT	-
	Nome	VARCHAR(45)	-
	data_nasc	DATE	-
	morada_rua	VARCHAR(45)	-
	morada_localidade	VARCHAR(45)	-
	morada_porta	INT	-
	morada_codigopostal	VARCHAR(45)	-

	contacto_telemovel	INT	-
	contacto_email	VARCHAR(45)	-
	Equipas_idEquipas	INT	IdEquipas(Equipas)

Tabela 37: Caracterização da tabela lógica do Pilotos

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Funcionarios da Equipa	idFuncionariosDaEquipa	INT	-
	Nome	VARCHAR(45)	-
		ENUM('Chefe', 'Mecânico', 'Engenheiro', 'Analista', 'PitCrew')	
	data_nasc	DATE	-
	morada_ rua	VARCHAR(45)	-
	morada_localidade	VARCHAR(45)	-
	morada_porta	INT	-
	morada_codigopostal	VARCHAR(45)	-
	contacto_telemovel	INT	-
	contacto_email	VARCHAR(45)	-
	Equipas_idEquipas	INT	IdEquipas(Equipas)

Tabela 38: Caracterização da tabela lógica do Funcionarios da Equipa

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Corrida	tipo	ENUM('Clássica', 'DragRace', '24', 'Rally')	-
	Pista	VARCHAR(45)	-
	idEvento	INT	idEvento(Evento)

Tabela 39: Caracterização da tabela lógica do Corrida

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Arbitro	idArbitro	INT	-
	Nome	VARCHAR(45)	-

	data_nasc	DATE	-
	morada_rua	VARCHAR(45)	-
	morada_localidade	VARCHAR(45)	-
	morada_porta	INT	-
	morada_codigopostal	VARCHAR(45)	-
	contacto_telemovei	INT	-
	contacto_email	VARCHAR(45)	-
	Corrida_tipo	ENUM('Clássica', 'DragRace', '24', 'Rally')	tipo(Corrida)
	idEvento	INT	idEvento(Evento)

Tabela 40: Caracterização da tabela lógica do Arbitro

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Comissarios	idComissarios	INT	-
	Nome	VARCHAR(45)	-
	data_nasc	DATE	-
	tipo	ENUM('Pista', 'Técnico', 'Médico', 'Controlo de Corrida')	-
	morada_rua	VARCHAR(45)	-
	morada_localidade	VARCHAR(45)	-
	morada_porta	INT	-
	morada_codigopostal	VARCHAR(45)	-
	contacto_telemovei	INT	-
	contacto_email	VARCHAR(45)	-
	Corrida_tipo	ENUM('Clássica', 'DragRace', '24', 'Rally')	"tipo"(Corrida)
	idEvento	INT	"idEvento"(Corrida)

Tabela 41: Caracterização da tabela lógica do Comissarios

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Material de Corrida	idMaterialDeCorrida	INT	-
	Nome	VARCHAR(45)	-
	Quantidade	INT	-
	Equipas_idEquipas	INT	"idEquipas"(Equipas)

Tabela 42: Caracterização da tabela lógica do Material de Corrida

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Patrocinadores	idPatrocinadores	INT	-
	Nome	VARCHAR(45)	-
	TipoMaterial	VARCHAR(45)	-
	morada_ rua	VARCHAR(45)	-
	morada_localidade	VARCHAR(45)	-
	morada_porta	INT	-
	morada_codigopostal	VARCHAR(45)	-
	contacto_telemovei	INT	-
	contacto_email	VARCHAR(45)	-

Tabela 43: Caracterização da tabela lógica do Patrocinadores

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
Classificacao	Posicao	INT	-
	numero_Piloto	INT	numero_Piloto(Pilotos)
	idEvento	INT	idEvento(Corrida)
	Pontos	INT	-
	Corrida_tipo	ENUM('Clássica', 'DragRace', '24', 'Rally')	tipo(Corrida)

Tabela 44: Caracterização da tabela lógica do participa

Tabela	Nome da Coluna	Tipo de dado	Referência (tabela)
oferecido	idMaterialDeCorrida	INT	“idMaterialDeCorrida”(Material de Corrida)
	idPatrocinadores	INT	“idPatrocinadores”(Patrocinadores)

Tabela 45: Caracterização da tabela lógica do oferecido

Tabela	Nome da	Tipo de dado	Referência (tabela)
--------	---------	--------------	---------------------

	Coluna		
Notificacao	idNotificacao	INT	-
	Descricao	VARCHAR(250)	-
	Data_notificacao	TIMESTAMP	-
	idEncomenda	INT	"idEncomenda"(Encomenda)

Tabela 46: Caracterização da tabela lógica da Notificacao

Ao relacionar as 22 tabelas, criamos um modelo lógico.

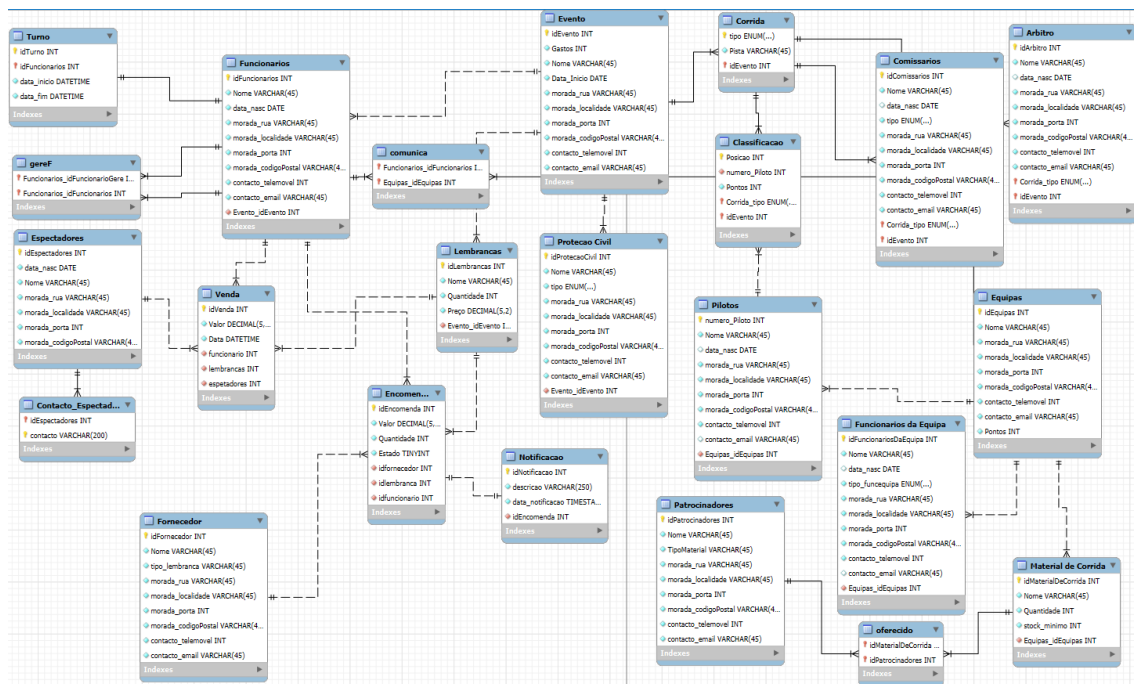


Figura 4: Ilustração do Modelo Lógico

4.4. Validação do modelo com interrogações do utilizador

No contexto do projeto, é essencial assegurar que o modelo lógico desenvolvido atenda aos requisitos de manipulação definidos. Uma das maneiras de validar o modelo é por meio da execução de consultas, ou queries, representadas em Álgebra Relacional. Essa abordagem possibilita verificar se as operações de manipulação de dados são capazes de fornecer as informações esperadas.

5. Implementação Física

5.1. Apresentação do processo de criação da base de dados

Para a criação do sistema de gestão de base de dados, foi implementado, no mySQL, cada tabela anteriormente mencionada na apresentação do modelo lógico.

Este processo inicia-se com duas instruções SQL essenciais: a primeira é responsável por criar a base de dados que vai ser utilizada, apelidando-a de *SuperKartsCup*, a segunda instrução indica ao sistema em qual a base de dados deve operar.

```
-- DROP DATABASE `SuperKartsCup` ;  
CREATE DATABASE `SuperKartsCup` ;  
  
USE `SuperKartsCup` ;
```

Figura 5: Instruções para a criação e utilização da base de dados

A primeira tabela a ser implementada é a **Evento**. Esta possui o identificador único “idEvento” definido pelos parâmetros “INT”, “NOT NULL”, “AUTO_INCREMENT” e “PRIMARY KEY”. As restantes colunas são “Gastos”, definido por “INT” e “NOT NULL”, “Nome”, definido por “VARCHAR(45)” e “NOT NULL”, “Data_Inicio” definida por “DATE” e “NOT NULL”, “morada_ rua” definida por “VARCHAR(45)” e “NOT NULL”, “morada_localidade” definida por “VARCHAR(45)” e “NOT NULL”, “morada_porta” definida por “INT” e “NOT NULL”, “morada_codigoPostal” definida por “VARCHAR(45)” e “NOT NULL”, “contacto_telemovei” definido por “INT” e “NOT NULL”, “contacto_email” definido por “VARCHAR(45)” e “NOT NULL”.

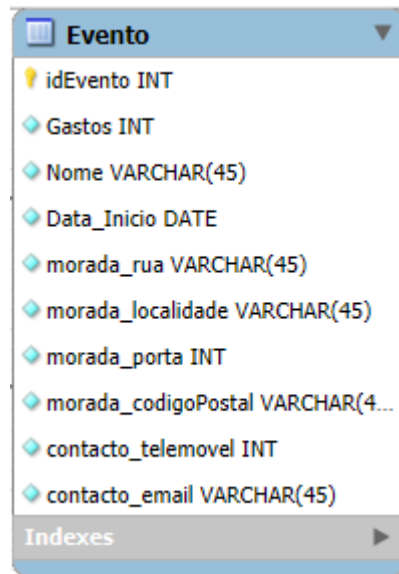


Figura 6: Tabela lógica do Evento

```
-- DROP TABLE `SuperKartsCup`.`Evento`;
CREATE TABLE `SuperKartsCup`.`Evento` (
  `idEvento` INT NOT NULL AUTO_INCREMENT,
  `Gastos` INT NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `Data_Inicio` DATE NOT NULL,
  `morada_rua` VARCHAR(45) NOT NULL,
  `morada_localidade` VARCHAR(45) NOT NULL,
  `morada_porta` INT NOT NULL,
  `morada_codigoPostal` VARCHAR(45) NOT NULL,
  `contacto_telemovel` INT NOT NULL,
  `contacto_email` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idEvento`)
);
```

Figura 7: Implementação do Evento

A próxima tabela a ser implementada é tabela **Funcionarios** possui como identificador único a coluna "idFuncionarios", definida com "INT", "NOT NULL", "AUTO_INCREMENT" e "PRIMARY KEY". Inclui ainda as colunas "Nome" ("VARCHAR(45)", "NOT NULL") e "data_nasc" ("DATE", "NOT NULL"), além de detalhes de morada, como "morada_rua", "morada_localidade" ("VARCHAR(45)", "NOT NULL"), "morada_porta" ("INT", "NOT NULL") e "morada_codigoPostal" ("VARCHAR(45)", "NOT NULL"). Os contactos são registados nas colunas "contacto_telemovel" ("INT", "NOT NULL") e "contacto_email" ("VARCHAR(45)", "NOT NULL"). A coluna "Evento_idEvento" ("INT", "NOT NULL") estabelece uma chave estrangeira que referência a tabela Evento, garantindo que cada funcionário esteja associado a um evento específico.

Funcionarios	
idFuncionarios	INT
Nome	VARCHAR(45)
data_nasc	DATE
morada_rua	VARCHAR(45)
morada_localidade	VARCHAR(45)
morada_porta	INT
morada_codigoPostal	VARCHAR(45)
contacto_telemovel	INT
contacto_email	VARCHAR(45)
Evento_idEvento	INT

Figura 8: Tabela lógica de Funcionarios

```
-- DROP TABLE `SuperKartsCup`.`Funcionarios`;
CREATE TABLE `SuperKartsCup`.`Funcionarios` (
  `idFuncionarios` INT NOT NULL AUTO_INCREMENT,
  `Nome` VARCHAR(45) NOT NULL,
  `data_nasc` DATE NOT NULL,
  `morada_rua` VARCHAR(45) NOT NULL,
  `morada_localidade` VARCHAR(45) NOT NULL,
  `morada_porta` INT NOT NULL,
  `morada_codigoPostal` VARCHAR(45) NOT NULL,
  `contacto_telemovel` INT NOT NULL,
  `contacto_email` VARCHAR(45) NOT NULL,
  `Evento_idEvento` INT NOT NULL,
  PRIMARY KEY (`idFuncionarios`),
  FOREIGN KEY (`Evento_idEvento`)
    REFERENCES `SuperKartsCup`.`Evento` (`idEvento`)
);
```

Figura 9: Implementação de Funcionarios

A tabela **Espectadores** possui como identificador único a coluna "idEspectadores", definida com "INT", "NOT NULL", "AUTO_INCREMENT" e "PRIMARY KEY". Inclui ainda a coluna "data_nasc", definida como "DATE" e "NOT NULL", para registar a data de nascimento de cada espectador, e a coluna "Nome", definida como "VARCHAR(45)" e "NOT NULL", para armazenar o nome. Os detalhes de morada são registados nas colunas "morada_rua" e "morada_localidade", ambas definidas como "VARCHAR(45)" e "NOT NULL", "morada_porta", definida como "INT" e "NOT NULL", e "morada_codigoPostal", definida como "VARCHAR(45)" e "NOT NULL".

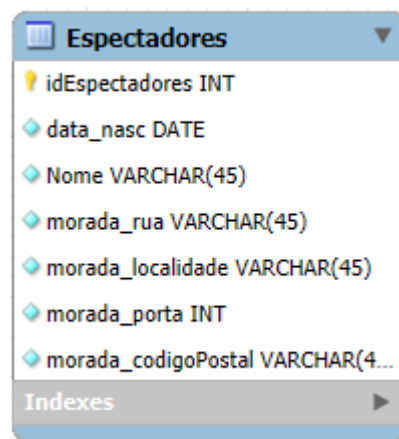


Figura 10: Tabela lógica de Espectadores

```
-- DROP TABLE `SuperKartsCup`.`Espectadores`;
CREATE TABLE `SuperKartsCup`.`Espectadores` (
  `idEspectadores` INT NOT NULL AUTO_INCREMENT,
  `data_nasc` DATE NOT NULL,
  `Nome` VARCHAR(45) NOT NULL,
  `morada_rua` VARCHAR(45) NOT NULL,
  `morada_localidade` VARCHAR(45) NOT NULL,
  `morada_porta` INT NOT NULL,
  `morada_codigoPostal` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idEspectadores`)
);
```

Figura 11: Implementação de Espectadores

A tabela **Lembrancas** possui como identificador único a coluna "idLembrancas", definida com "INT", "NOT NULL", "AUTO_INCREMENT" e "PRIMARY KEY". Inclui a coluna "Nome", definida como "VARCHAR(45)" e "NOT NULL", para armazenar o nome da lembrança, e a coluna "Quantidade", definida como "INT" e "NOT NULL", que indica o número disponível de cada item. O preço de cada lembrança é registado na coluna "Preço", definida como "DECIMAL(5,2)" e "NOT NULL", garantindo a precisão necessária para valores monetários. A coluna "Evento_idEvento", definida como "INT" e "NOT NULL", estabelece uma chave estrangeira que referencia a coluna "idEvento" da tabela Evento, ligando cada lembrança a um evento específico.

Lembrancas	
idLembrancas	INT
Nome	VARCHAR(45)
Quantidade	INT
Preço	DECIMAL(5,2)
Evento_idEvento	INT

Indexes

Figura 12: Tabela lógica de Lembrancas

```
-- DROP TABLE `SuperKartsCup`.`Lembrancas`;
CREATE TABLE `SuperKartsCup`.`Lembrancas` (
  `idLembrancas` INT NOT NULL AUTO_INCREMENT,
  `Nome` VARCHAR(45) NOT NULL,
  `Quantidade` INT NOT NULL,
  `Preço` DECIMAL(5,2) NOT NULL,
  `Evento_idEvento` INT NOT NULL,
  PRIMARY KEY (`idLembrancas`),
  FOREIGN KEY (`Evento_idEvento`)
    REFERENCES `SuperKartsCup`.`Evento` (`idEvento`)
);
```

Figura 13: Implementação de Lembrancas

A tabela **Venda** possui como identificador único a coluna "idVenda", definida com "INT", "NOT NULL", "AUTO_INCREMENT" e "PRIMARY KEY". A coluna "Valor", definida como "DECIMAL(5,2)" e "NOT NULL", registra o valor da venda, enquanto a coluna "Data_venda", definida como "DATETIME" e "NOT NULL", armazena a data e hora em que a venda foi realizada. A tabela também inclui três colunas que estabelecem relações com outras tabelas: "funcionario", definida como "INT" e "NOT NULL", que faz referência à coluna "idFuncionarios" da tabela Funcionarios; "lembrancas", definida como "INT" e "NOT NULL", que referencia a coluna "idLembrancas" da tabela Lembrancas; e "espectadores", também definida como "INT" e "NOT NULL", que faz referência à coluna "idEspectadores" da tabela Espectadores. As chaves estrangeiras garantem a integridade referencial entre a tabela Venda e as tabelas associadas, assegurando que cada venda esteja corretamente ligada a um funcionário, um espectador e uma lembrança.

Venda	
idVenda	INT
Valor	DECIMAL(5,...
Data	DATETIME
funcionario	INT
lembrancas	INT
espetadores	INT
Indexes	

Figura 14: Tabela lógica da Venda

```
-- DROP TABLE `SuperKartsCup`.`Venda`;
CREATE TABLE `SuperKartsCup`.`Venda` (
  `idVenda` INT NOT NULL AUTO_INCREMENT,
  `Valor` DECIMAL(5,2) NOT NULL,
  `Data_venda` DATETIME NOT NULL,
  `funcionario` INT NOT NULL,
  `lembrancas` INT NOT NULL,
  `espectadores` INT NOT NULL,
  PRIMARY KEY (`idVenda`),
  FOREIGN KEY (`funcionario`)
    REFERENCES `SuperKartsCup`.`Funcionarios` (`idFuncionarios`),
  FOREIGN KEY (`espectadores`)
    REFERENCES `SuperKartsCup`.`Espectadores` (`idEspectadores`),
  FOREIGN KEY (`lembrancas`)
    REFERENCES `SuperKartsCup`.`Lembrancas` (`idLembrancas`)
);
```

Figura 15: Implementação da Venda

A tabela **Fornecedor** possui como identificador único a coluna "idFornecedor", definida com "INT", "NOT NULL", "AUTO_INCREMENT" e "PRIMARY KEY". A coluna "Nome", definida como "VARCHAR(45)" e "NOT NULL", armazena o nome do fornecedor, enquanto a coluna "tipo_lembraça", também definida como "VARCHAR(45)" e "NOT NULL", especifica o tipo de lembrança fornecido. As informações sobre a morada do fornecedor são registadas nas colunas "morada_ rua", "morada_localidade" (ambas definidas como "VARCHAR(45)" e "NOT NULL"), "morada_porta", definida como "INT" e "NOT NULL", e "morada_codigoPostal", definida como "VARCHAR(45)" e "NOT NULL". Além disso, os meios de contacto são armazenados nas colunas "contacto_telemovei", definida como "INT" e "NOT NULL", e "contacto_email", definida como "VARCHAR(45)" e "NOT NULL".

Fornecedor	
idFornecedor	INT
Nome	VARCHAR(45)
tipo_lembanca	VARCHAR(45)
morada_rua	VARCHAR(45)
morada_localidade	VARCHAR(45)
morada_porta	INT
morada_codigoPostal	VARCHAR(45)
contacto_telemovel	INT
contacto_email	VARCHAR(45)

Figura 16: Tabela lógica do Fornecedor

```
-- DROP TABLE `SuperKartsCup`.`Fornecedor`;
CREATE TABLE `SuperKartsCup`.`Fornecedor` (
  `idFornecedor` INT NOT NULL AUTO_INCREMENT,
  `Nome` VARCHAR(45) NOT NULL,
  `tipo_lembanca` VARCHAR(45) NOT NULL,
  `morada_rua` VARCHAR(45) NOT NULL,
  `morada_localidade` VARCHAR(45) NOT NULL,
  `morada_porta` INT NOT NULL,
  `morada_codigoPostal` VARCHAR(45) NOT NULL,
  `contacto_telemovel` INT NOT NULL,
  `contacto_email` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idFornecedor`)
);
```

Figura 17: Implementação do Fornecedor

A tabela **Encomenda** possui como identificador único a coluna "idEncomenda", definida com "INT", "NOT NULL", "AUTO_INCREMENT" e "PRIMARY KEY". A coluna "Valor", definida como "DECIMAL(5,2)" e "NOT NULL", regista o valor total da encomenda, enquanto a coluna "Quantidade", definida como "INT" e "NOT NULL", especifica a quantidade de itens na encomenda. A coluna "Estado", definida como "TINYINT" e "NOT NULL", é utilizada para indicar o estado da encomenda (por exemplo, 1 para pendente, 0 para concluída). As colunas "idfuncionario", "idfornecedor" e "idlembanca" são definidas como "INT" e "NOT NULL", e representam as relações com a tabela Funcionarios, Fornecedor e Lembrancas, respetivamente. As chaves estrangeiras garantem que cada encomenda esteja associada a um funcionário responsável, um fornecedor e uma lembrança específica, mantendo a integridade referencial entre as tabelas.

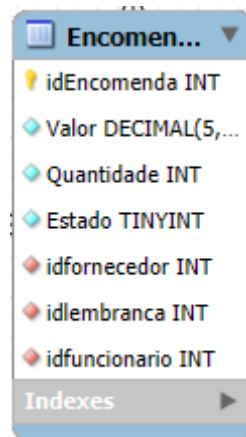


Figura 18: Tabela lógica da Encomenda

```
-- DROP TABLE `SuperKartsCup`.`Encomenda`;
CREATE TABLE `SuperKartsCup`.`Encomenda` (
  `idEncomenda` INT NOT NULL AUTO_INCREMENT,
  `Valor` DECIMAL(5,2) NOT NULL,
  `Quantidade` INT NOT NULL,
  `Estado` TINYINT NOT NULL,
  `idfuncionario` INT NOT NULL,
  `idfornecedor` INT NOT NULL,
  `idlembanca` INT NOT NULL,
  PRIMARY KEY (`idEncomenda`),
  FOREIGN KEY (`idfuncionario`)
    REFERENCES `SuperKartsCup`.`Funcionarios` (`idFuncionarios`),
  FOREIGN KEY (`idfornecedor`)
    REFERENCES `SuperKartsCup`.`Fornecedor` (`idFornecedor`),
  FOREIGN KEY (`idlembanca`)
    REFERENCES `SuperKartsCup`.`Lembrancas` (`idLembrancas`)
);
```

Figura 19: Implementação da Encomenda

A tabela **Corrida** possui como chave primária composta as colunas "tipo" e "idEvento". A coluna "tipo" é definida como ENUM, com os valores possíveis 'Clássica', 'DragRace', '24' e 'Rally', representando os diferentes tipos de corridas. A coluna "Pista", definida como "VARCHAR(45)" e "NOT NULL", armazena o nome ou a localização da pista onde a corrida ocorre. A coluna "idEvento", definida como "INT" e "NOT NULL", estabelece uma relação com a tabela Evento, sendo referenciada por uma chave estrangeira que garante a integridade referencial. Essa chave estrangeira assegura que cada corrida esteja associada a um evento específico.

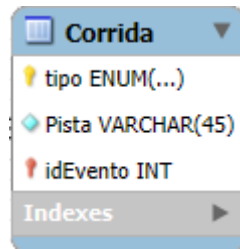


Figura 20: Tabela lógica da Corrida

```
-- DROP TABLE `SuperKartsCup`.`Corrida`;
CREATE TABLE `SuperKartsCup`.`Corrida` (
  `tipo` ENUM('Clássica', 'DragRace', '24', 'Rally') NOT NULL,
  `Pista` VARCHAR(45) NOT NULL,
  `idEvento` INT NOT NULL,
  PRIMARY KEY (`tipo`, `idEvento`),
  FOREIGN KEY (`idEvento`)
    REFERENCES `SuperKartsCup`.`Evento` (`idEvento`)
);
```

Figura 21: Implementação da Corrida

A tabela **Equipas** possui como identificador único a coluna "idEquipas", definida com "INT", "NOT NULL", "AUTO_INCREMENT" e "PRIMARY KEY". A coluna "Nome", definida como "VARCHAR(45)" e "NOT NULL", armazena o nome da equipa. As informações relativas à morada são registadas nas colunas "morada_rua" e "morada_localidade", ambas definidas como "VARCHAR(45)" e "NOT NULL", "morada_porta", definida como "INT" e "NOT NULL", e "morada_codigoPostal", definida como "VARCHAR(45)" e "NOT NULL". Os meios de contacto são registados nas colunas "contacto_telemovei", definida como "INT" e "NOT NULL", e "contacto_email", definida como "VARCHAR(45)" e "NOT NULL". A coluna "Pontos", definida como "INT" com valor padrão 0, permite registar os pontos acumulados pela equipa, sendo atribuída inicialmente como 0.

Equipas	
idEquipas	INT
Nome	VARCHAR(45)
morada_rua	VARCHAR(45)
morada_localidade	VARCHAR(45)
morada_porta	INT
morada_codigoPostal	VARCHAR(45)
contacto_telemovel	INT
contacto_email	VARCHAR(45)
Pontos	INT

Figura 22: Tabela lógica de Equipas

```
-- DROP TABLE `SuperKartsCup`.`Equipas`;
CREATE TABLE `SuperKartsCup`.`Equipas` (
  `idEquipas` INT NOT NULL AUTO_INCREMENT,
  `Nome` VARCHAR(45) NOT NULL,
  `morada_rua` VARCHAR(45) NOT NULL,
  `morada_localidade` VARCHAR(45) NOT NULL,
  `morada_porta` INT NOT NULL,
  `morada_codigoPostal` VARCHAR(45) NOT NULL,
  `contacto_telemovel` INT NOT NULL,
  `contacto_email` VARCHAR(45) NOT NULL,
  `Pontos` INT DEFAULT 0,
  PRIMARY KEY (`idEquipas`)
);
```

Figura 23: Implementação de Equipas

A tabela **Pilotos** possui como identificador único a coluna "numero_Piloto", definida com "INT" e "NOT NULL", que representa o número do piloto. A coluna "Nome", definida como "VARCHAR(45)" e "NOT NULL", armazena o nome do piloto, enquanto a coluna "data_nasc", definida como "DATE" e permitida como "NULL", armazena a data de nascimento do piloto, caso esteja disponível. As informações sobre a morada do piloto são registadas nas colunas "morada_rua" e "morada_localidade", ambas definidas como "VARCHAR(45)" e "NOT NULL", "morada_porta", definida como "INT" e "NOT NULL", e "morada_codigoPostal", definida como "VARCHAR(45)" e "NOT NULL". Os contactos do piloto são registados nas colunas "contacto_telemovel", definida como "INT" e "NOT NULL", e "contacto_email", definida como "VARCHAR(45)" e permitida como "NULL". A coluna "Equipas_idEquipas", definida como "INT"

e "NOT NULL", estabelece uma chave estrangeira que referencia a coluna "idEquipas" da tabela Equipas, ligando cada piloto a uma equipa específica.

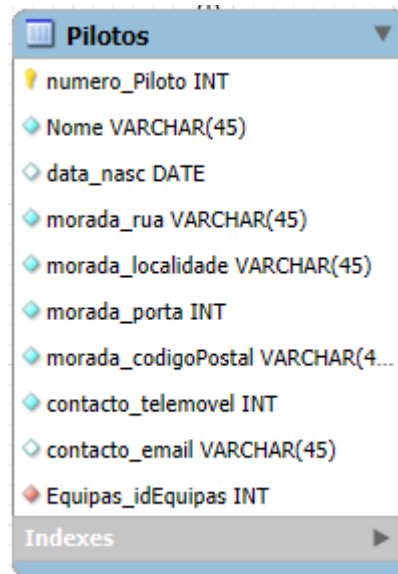


Figura 24: Tabela lógica de Pilotos

```
-- DROP TABLE `SuperKartsCup`.`Pilotos`;  
CREATE TABLE `SuperKartsCup`.`Pilotos` (  
  `numero_Piloto` INT NOT NULL,  
  `Nome` VARCHAR(45) NOT NULL,  
  `data_nasc` DATE NULL,  
  `morada_rua` VARCHAR(45) NOT NULL,  
  `morada_localidade` VARCHAR(45) NOT NULL,  
  `morada_porta` INT NOT NULL,  
  `morada_codigoPostal` VARCHAR(45) NOT NULL,  
  `contacto_telemovel` INT NOT NULL,  
  `contacto_email` VARCHAR(45) NULL,  
  `Equipas_idEquipas` INT NOT NULL,  
  PRIMARY KEY (`numero_Piloto`),  
  FOREIGN KEY (`Equipas_idEquipas`)  
    REFERENCES `SuperKartsCup`.`Equipas` (`idEquipas`)  
)
```

Figura 25: Implementação de Pilotos

A tabela **Classificacao** possui como chave primária composta as colunas "Corrida_tipo", "Posicao" e "idEvento". A coluna "Corrida_tipo", definida como ENUM com os valores possíveis 'Clássica', 'DragRace', '24' e 'Rally', indica o tipo de corrida. A coluna "Posicao", definida como "INT" e "NOT NULL", regista a posição final do piloto na corrida. A coluna "numero_Piloto", definida como "INT" e "NOT NULL", faz referência ao número do piloto e estabelece uma

ligação com a tabela Pilotos. A coluna "idEvento", definida como "INT" e "NOT NULL", identifica o evento da corrida e estabelece uma chave estrangeira com a tabela Corrida através das colunas "Corrida_tipo" e "idEvento". A coluna "Pontos", definida como "INT" e "NOT NULL", armazena os pontos atribuídos ao piloto conforme a sua posição na corrida.

As chaves estrangeiras asseguram a integridade referencial, garantindo que cada classificação esteja associada ao tipo de corrida, ao evento e ao piloto correspondentes.

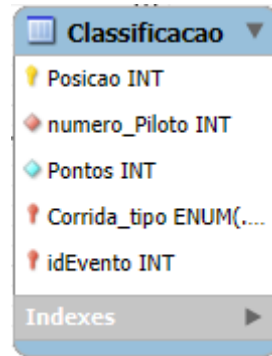


Figura 26: Tabela lógica da Classificacao

```
-- DROP TABLE `SuperKartsCup`.`Classificacao`;
CREATE TABLE `SuperKartsCup`.`Classificacao` (
  `Corrida_tipo` ENUM('Clássica', 'DragRace', '24', 'Rally') NOT NULL,
  `Posicao` INT NOT NULL,
  `numero_Piloto` INT NOT NULL,
  `idEvento` INT NOT NULL,
  `Pontos` INT NOT NULL,
  PRIMARY KEY (`Corrida_tipo`, `Posicao`, `idEvento`),
  FOREIGN KEY (`Corrida_tipo`, `idEvento`)
    REFERENCES `SuperKartsCup`.`Corrida` (`tipo`, `idEvento`),
  FOREIGN KEY (`numero_Piloto`)
    REFERENCES `SuperKartsCup`.`Pilotos` (`numero_Piloto`)
);
```

Figura 27: Implementação da Classificacao

A tabela **Funcionarios da Equipe** possui como identificador único a coluna "idFuncionariosDaEquipa", definida com "INT", "NOT NULL", "AUTO_INCREMENT" e "PRIMARY KEY". A coluna "Nome", definida como "VARCHAR(45)" e "NOT NULL", armazena o nome do funcionário da equipa. A coluna "data_nasc", definida como "DATE" e permitida como "NULL", regista a data de nascimento do funcionário, caso disponível.

A coluna "tipo_funcequipa", definida como ENUM com os valores possíveis 'Chefe', 'Mecânico', 'Engenheiro', 'Analista' e 'PitCrew', especifica o cargo ou função do funcionário dentro da equipa. As informações sobre a morada do funcionário são registadas nas colunas

"morada_rua" e "morada_localidade", ambas definidas como "VARCHAR(45)" e "NOT NULL", "morada_porta", definida como "INT" e "NOT NULL", e "morada_codigoPostal", definida como "VARCHAR(45)" e "NOT NULL". O contacto do funcionário é registado nas colunas "contacto_telemovel", definida como "INT" e "NOT NULL", e "contacto_email", definida como "VARCHAR(45)" e permitida como "NULL". A coluna "Equipas_idEquipas", definida como "INT" e "NOT NULL", estabelece uma chave estrangeira que faz referência à tabela Equipas e à coluna "idEquipas", garantindo que cada funcionário da equipa esteja associado a uma equipa específica.

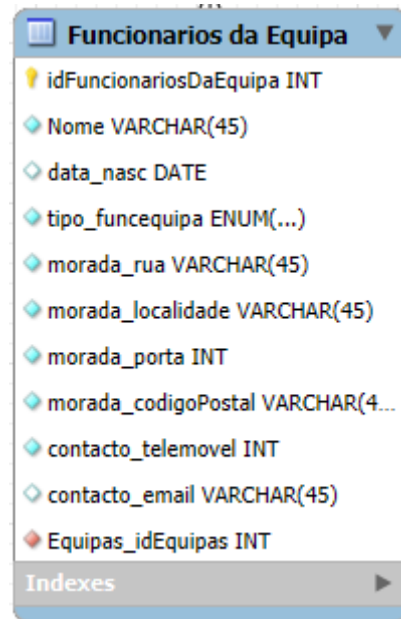


Figura 28: Tabela lógica de Funcionarios da Equipa

```
-- DROP TABLE `SuperKartsCup`.`Funcionarios da Equipa`;
CREATE TABLE `SuperKartsCup`.`Funcionarios da Equipa` (
  `idFuncionariosDaEquipas` INT NOT NULL AUTO_INCREMENT,
  `Nome` VARCHAR(45) NOT NULL,
  `data_nasc` DATE NULL,
  `tipo_funcequipa` ENUM('Chefe', 'Mecânico', 'Engenheiro', 'Analista', 'PitCrew') NOT NULL,
  `morada_rua` VARCHAR(45) NOT NULL,
  `morada_localidade` VARCHAR(45) NOT NULL,
  `morada_porta` INT NOT NULL,
  `morada_codigoPostal` VARCHAR(45) NOT NULL,
  `contacto_telemovel` INT NOT NULL,
  `contacto_email` VARCHAR(45) NULL,
  `Equipas_idEquipas` INT NOT NULL,
  PRIMARY KEY (`idFuncionariosDaEquipas`),
  FOREIGN KEY (`Equipas_idEquipas`)
    REFERENCES `SuperKartsCup`.`Equipas` (`idEquipas`)
);
```

Figura 29: Implementação de Funcionarios da Equipa

A tabela **Protecao Civil** possui como identificador único a coluna "idProtecaoCivil", definida com "INT", "NOT NULL", "AUTO_INCREMENT" e "PRIMARY KEY". A coluna "Nome", definida como "VARCHAR(45)" e "NOT NULL", armazena o nome da instituição ou entidade de proteção civil. A coluna "tipo", definida como ENUM com os valores possíveis 'Bombeiros' e 'GNR', especifica o tipo de unidade de proteção civil associada. As informações relativas à morada são registadas nas colunas "morada_rua" e "morada_localidade", ambas definidas como "VARCHAR(45)" e "NOT NULL", "morada_porta", definida como "INT" e "NOT NULL", e "morada_codigoPostal", definida como "VARCHAR(45)" e "NOT NULL". Os contactos são registados nas colunas "contacto_telemovel", definida como "INT" e "NOT NULL", e "contacto_email", definida como "VARCHAR(45)" e "NOT NULL". A coluna "Evento_idEvento", definida como "INT" e "NOT NULL", estabelece uma chave estrangeira que faz referência à tabela Evento e à coluna "idEvento", garantindo que cada unidade de proteção civil esteja associada a um evento específico.

idProtecaoCivil	INT
Nome	VARCHAR(45)
tipo	ENUM(...)
morada_rua	VARCHAR(45)
morada_localidade	VARCHAR(45)
morada_porta	INT
morada_codigoPostal	VARCHAR(45)
contacto_telemovel	INT
contacto_email	VARCHAR(45)
Evento_idEvento	INT

Indexes

Figura 30: Tabela lógica da Protecao Civil

```

-- DROP TABLE `SuperKartsCup`.`Protecao Civil`;
CREATE TABLE `SuperKartsCup`.`Protecao Civil` (
  `idProtecaoCivil` INT NOT NULL AUTO_INCREMENT,
  `Nome` VARCHAR(45) NOT NULL,
  `tipo` ENUM('Bombeiros', 'GNR') NOT NULL,
  `morada_rua` VARCHAR(45) NOT NULL,
  `morada_localidade` VARCHAR(45) NOT NULL,
  `morada_porta` INT NOT NULL,
  `morada_codigoPostal` VARCHAR(45) NOT NULL,
  `contacto_telemovel` INT NOT NULL,
  `contacto_email` VARCHAR(45) NOT NULL,
  `Evento_idEvento` INT NOT NULL,
  PRIMARY KEY (`idProtecaoCivil`),
  FOREIGN KEY (`Evento_idEvento`)
    REFERENCES `SuperKartsCup`.`Evento` (`idEvento`)
);

```

Figura 31: Implementação da Protecao Civil

A tabela **Arbitro** possui como chave primária composta as colunas "idArbitro", "Corrida_tipo" e "idEvento". A coluna "idArbitro", definida como "INT", "NOT NULL" e "AUTO_INCREMENT", é o identificador único para cada árbitro. A coluna "Nome", definida como "VARCHAR(45)" e "NOT NULL", armazena o nome do árbitro, enquanto a coluna "data_nasc", definida como "DATE" e permitida como "NULL", armazena a data de nascimento do árbitro, se disponível. As informações de morada são registadas nas colunas "morada_rua" e "morada_localidade", ambas definidas como "VARCHAR(45)" e "NOT NULL", "morada_porta", definida como "INT" e "NOT NULL", e "morada_codigoPostal", definida como "VARCHAR(45)" e "NOT NULL". Os contactos são registados nas colunas "contacto_telemovel", definida como "INT" e "NOT NULL", e "contacto_email", definida como "VARCHAR(45)" e "NOT NULL". A coluna "Corrida_tipo", definida como ENUM com os valores possíveis 'Clássica', 'DragRace', '24' e 'Rally', especifica o tipo de corrida em que o árbitro irá atuar. A coluna "idEvento", definida como "INT" e "NOT NULL", faz referência ao evento ao qual o árbitro está associado. As chaves estrangeiras garantem que cada árbitro esteja vinculado a um tipo de corrida e a um evento específico, através da tabela Corrida.

Arbitro	
idArbitro	INT
Nome	VARCHAR(45)
data_nasc	DATE
morada_rua	VARCHAR(45)
morada_localidade	VARCHAR(45)
morada_porta	INT
morada_codigoPostal	VARCHAR(45)
contacto_telemovel	INT
contacto_email	VARCHAR(45)
Corrida_tipo	ENUM(...)
idEvento	INT
Indexes	

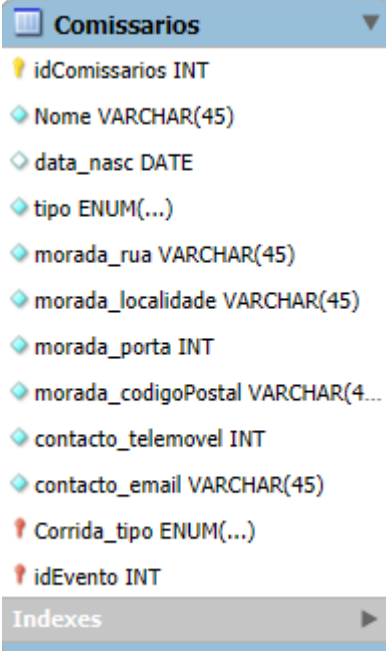
Figura 32: Tabela lógica do Arbitro

```
-- DROP TABLE `SuperKartsCup`.`Arbitro`;
CREATE TABLE `SuperKartsCup`.`Arbitro` (
  `idArbitro` INT NOT NULL AUTO_INCREMENT,
  `Nome` VARCHAR(45) NOT NULL,
  `data_nasc` DATE NULL,
  `morada_rua` VARCHAR(45) NOT NULL,
  `morada_localidade` VARCHAR(45) NOT NULL,
  `morada_porta` INT NOT NULL,
  `morada_codigoPostal` VARCHAR(45) NOT NULL,
  `contacto_telemovel` INT NOT NULL,
  `contacto_email` VARCHAR(45) NOT NULL,
  `Corrida_tipo` ENUM('Clássica', 'DragRace', '24', 'Rally') NOT NULL,
  `idEvento` INT NOT NULL,
  PRIMARY KEY (`idArbitro`, `Corrida_tipo`, `idEvento`),
  FOREIGN KEY (`Corrida_tipo`, `idEvento`)
    REFERENCES `SuperKartsCup`.`Corrida` (`tipo`, `idEvento`)
);
```

Figura 33: Implementação do Arbitro

A tabela **Comissarios** possui como chave primária composta as colunas "idComissarios", "Corrida_tipo" e "idEvento". A coluna "idComissarios", definida como "INT", "NOT NULL" e "AUTO_INCREMENT", é o identificador único para cada comissário. A coluna "Nome", definida como "VARCHAR(45)" e "NOT NULL", armazena o nome do comissário. A coluna "data_nasc", definida como "DATE" e permitida como "NULL", armazena a data de nascimento do

comissário, caso disponível. A coluna "tipo", definida como ENUM com os valores possíveis 'Pista', 'Técnico', 'Médico' e 'Controlo de Corrida', especifica o tipo de comissário, indicando a função que ele desempenha durante o evento. As informações sobre a morada são registadas nas colunas "morada_rua" e "morada_localidade", ambas definidas como "VARCHAR(45)" e "NOT NULL", "morada_porta", definida como "INT" e "NOT NULL", e "morada_codigoPostal", definida como "VARCHAR(45)" e "NOT NULL". Os contactos são registados nas colunas "contacto_telemovel", definida como "INT" e "NOT NULL", e "contacto_email", definida como "VARCHAR(45)" e "NOT NULL". A coluna "Corrida_tipo", definida como ENUM com os valores possíveis 'Clássica', 'DragRace', '24' e 'Rally', especifica o tipo de corrida em que o comissário está envolvido. A coluna "idEvento", definida como "INT" e "NOT NULL", faz referência ao evento ao qual o comissário está associado. As chaves estrangeiras garantem que cada comissário esteja vinculado a um tipo de corrida e a um evento específico, através da tabela Corrida.



Comissarios	
idComissarios	INT
Nome	VARCHAR(45)
data_nasc	DATE
tipo	ENUM(...)
morada_rua	VARCHAR(45)
morada_localidade	VARCHAR(45)
morada_porta	INT
morada_codigoPostal	VARCHAR(45)
contacto_telemovel	INT
contacto_email	VARCHAR(45)
Corrida_tipo	ENUM(...)
idEvento	INT
Indexes	

Figura 34: Tabela lógica de Comissarios

```
-- DROP TABLE `SuperKartsCup`.`Comissarios`;
CREATE TABLE `SuperKartsCup`.`Comissarios` (
  `idComissarios` INT NOT NULL AUTO_INCREMENT,
  `Nome` VARCHAR(45) NOT NULL,
  `data_nasc` DATE NULL,
  `tipo` ENUM('Pista', 'Técnico', 'Médico', 'Controlo de Corrida') NOT NULL,
  `morada_ rua` VARCHAR(45) NOT NULL,
  `morada_localidade` VARCHAR(45) NOT NULL,
  `morada_porta` INT NOT NULL,
  `morada_codigoPostal` VARCHAR(45) NOT NULL,
  `contacto_telemovei` INT NOT NULL,
  `contacto_email` VARCHAR(45) NOT NULL,
  `Corrida_tipo` ENUM('Clássica', 'DragRace', '24', 'Rally') NOT NULL,
  `idEvento` INT NOT NULL,
  PRIMARY KEY (`idComissarios`,`Corrida_tipo`,`idEvento`),
  FOREIGN KEY (`Corrida_tipo`,`idEvento`)
    REFERENCES `SuperKartsCup`.`Corrida` (`tipo`,`idEvento`)
);
```

Figura 35: Implementação de Comissarios

A tabela **Material de Corrida** possui como identificador único a coluna "idMaterialDeCorrida", definida como "INT", "NOT NULL", "AUTO_INCREMENT" e "PRIMARY KEY". A coluna "Nome", definida como "VARCHAR(45)" e "NOT NULL", armazena o nome do material de corrida. A coluna "Quantidade", definida como "INT" e "NOT NULL", regista a quantidade disponível desse material. A coluna "stock_minimo", definida como "INT" e "NOT NULL", especifica a quantidade mínima de material de corrida que deve ser mantida em stock. Esta informação é importante para a gestão do material, garantindo que não haja falta de itens essenciais para a equipa. A coluna "Equipas_idEquipas", definida como "INT" e "NOT NULL", estabelece uma chave estrangeira que faz referência à tabela Equipas e à coluna "idEquipas", associando cada item de material a uma equipa específica.

Material de Corrida	
idMaterialDeCorrida	INT
Nome	VARCHAR(45)
Quantidade	INT
stock_minimo	INT
Equipas_idEquipas	INT
Indexes	

Figura 36: Tabela lógica de Material de Corrida

```
-- DROP TABLE `SuperKartsCup`.`Material de Corrida`;
CREATE TABLE `SuperKartsCup`.`Material de Corrida` (
  `idMaterialDeCorrida` INT NOT NULL AUTO_INCREMENT,
  `Nome` VARCHAR(45) NOT NULL,
  `Quantidade` INT NOT NULL,
  `stock_minimo` INT NOT NULL,
  `Equipas_idEquipas` INT NOT NULL,
  PRIMARY KEY (`idMaterialDeCorrida`),
  FOREIGN KEY (`Equipas_idEquipas`)
    REFERENCES `SuperKartsCup`.`Equipas` (`idEquipas`)
);
```

Figura 37: Implementação de Material de Corrida

A tabela **Patrocinadores** possui como identificador único a coluna "idPatrocinadores", definida como "INT", "NOT NULL", "AUTO_INCREMENT" e "PRIMARY KEY". A coluna "Nome", definida como "VARCHAR(45)" e "NOT NULL", armazena o nome do patrocinador. A coluna "TipoMaterial", também definida como "VARCHAR(45)" e "NOT NULL", especifica o tipo de material fornecido pelo patrocinador, como por exemplo equipamentos ou produtos relacionados com o evento. As informações de morada são registadas nas colunas "morada_rua" e "morada_localidade", ambas definidas como "VARCHAR(45)" e "NOT NULL", "morada_porta", definida como "INT" e "NOT NULL", e "morada_codigoPostal", definida como "VARCHAR(45)" e "NOT NULL". Os contactos são registados nas colunas "contacto_telemovei", definida como "INT" e "NOT NULL", e "contacto_email", definida como "VARCHAR(45)" e "NOT NULL".

Patrocinadores	
idPatrocinadores	INT
Nome	VARCHAR(45)
TipoMaterial	VARCHAR(45)
morada_rua	VARCHAR(45)
morada_localidade	VARCHAR(45)
morada_porta	INT
morada_codigoPostal	VARCHAR(45)
contacto_telemovei	INT
contacto_email	VARCHAR(45)
Indexes	

Figura 38: Tabela lógica de Patrocinadores

```
-- DROP TABLE `SuperKartsCup`.`Patrocinadores`;
CREATE TABLE `SuperKartsCup`.`Patrocinadores` (
  `idPatrocinadores` INT NOT NULL AUTO_INCREMENT,
  `Nome` VARCHAR(45) NOT NULL,
  `TipoMaterial` VARCHAR(45) NOT NULL,
  `morada_rua` VARCHAR(45) NOT NULL,
  `morada_localidade` VARCHAR(45) NOT NULL,
  `morada_porta` INT NOT NULL,
  `morada_codigoPostal` VARCHAR(45) NOT NULL,
  `contacto_telemovel` INT NOT NULL,
  `contacto_email` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idPatrocinadores`)
);
```

Figura 39: Implementação de Patrocinadores

A tabela **comunica** possui como chave primária composta as colunas "Funcionarios_idFuncionarios" e "Equipas_idEquipas", que estabelecem uma relação entre os funcionários e as equipas. A coluna "Funcionarios_idFuncionarios", definida como "INT" e "NOT NULL", faz referência ao identificador único de um funcionário da tabela Funcionarios. A coluna "Equipas_idEquipas", também definida como "INT" e "NOT NULL", faz referência ao identificador único de uma equipa da tabela Equipas. As chaves estrangeiras garantem a integridade referencial entre as tabelas, assegurando que cada funcionário esteja associado a uma equipa válida no sistema. A coluna "Funcionarios_idFuncionarios" referencia a coluna "idFuncionarios" na tabela Funcionarios, enquanto a coluna "Equipas_idEquipas" referencia a coluna "idEquipas" na tabela Equipas.

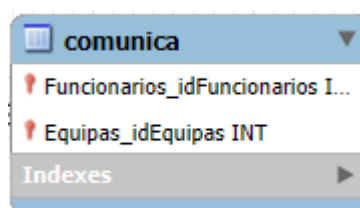


Figura 40: Tabela lógica da Comunica

```
-- DROP TABLE `SuperKartsCup`.`comunica`;
CREATE TABLE `SuperKartsCup`.`comunica` (
  `Funcionarios_idFuncionarios` INT NOT NULL,
  `Equipas_idEquipas` INT NOT NULL,
  PRIMARY KEY (`Funcionarios_idFuncionarios`, `Equipas_idEquipas`),
  FOREIGN KEY (`Funcionarios_idFuncionarios`)
    REFERENCES `SuperKartsCup`.`Funcionarios` (`idFuncionarios`),
  FOREIGN KEY (`Equipas_idEquipas`)
    REFERENCES `SuperKartsCup`.`Equipas` (`idEquipas`)
);
```

Figura 41: Implementação de Comunica

A tabela **gereF** possui como chave primária composta as colunas "Funcionarios_idFuncionarioGere" e "Funcionarios_idFuncionarios", que representam a relação de gestão ou supervisão entre funcionários dentro da empresa. A coluna "Funcionarios_idFuncionarioGere", definida como "INT" e "NOT NULL", faz referência ao identificador único de um funcionário que gerencia ou supervisiona outro. Já a coluna "Funcionarios_idFuncionarios", também definida como "INT" e "NOT NULL", faz referência ao identificador único de um funcionário que está sob a gestão ou supervisão de outro. As chaves estrangeiras garantem que ambas as colunas "Funcionarios_idFuncionarioGere" e "Funcionarios_idFuncionarios" façam referência à tabela Funcionarios, através da coluna "idFuncionarios", assegurando que as relações de gestão sejam entre funcionários válidos.

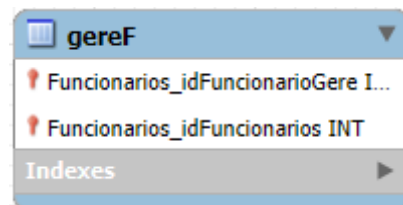


Figura 42: Tabela lógica de gereF

```
-- DROP TABLE `SuperKartsCup`.`gereF`;
CREATE TABLE `SuperKartsCup`.`gereF` (
  `Funcionarios_idFuncionarioGere` INT NOT NULL,
  `Funcionarios_idFuncionarios` INT NOT NULL,
  PRIMARY KEY (`Funcionarios_idFuncionarioGere`, `Funcionarios_idFuncionarios`),
  FOREIGN KEY (`Funcionarios_idFuncionarioGere`)
    REFERENCES `SuperKartsCup`.`Funcionarios` (`idFuncionarios`),
  FOREIGN KEY (`Funcionarios_idFuncionarios`)
    REFERENCES `SuperKartsCup`.`Funcionarios` (`idFuncionarios`)
);
```

Figura 43: Implementação de gereF

A tabela **Turno** possui como chave primária composta as colunas "idTurno" e "idFuncionarios". A coluna "idTurno", definida como "INT", "NOT NULL" e "AUTO_INCREMENT", é o identificador único para cada turno. A coluna "idFuncionarios", definida como "INT" e "NOT NULL", faz referência ao identificador único de um funcionário da tabela Funcionarios, associando o turno a um funcionário específico. As colunas "data_inicio" e "data_fim", ambas definidas como "DATETIME" e "NOT NULL", registram o horário de início e o horário de fim do turno do funcionário. Isso permite controlar a duração do turno e gerenciar os horários de trabalho dos funcionários. A chave estrangeira na coluna "idFuncionarios" garante que cada turno esteja associado a um funcionário válido da tabela Funcionarios.

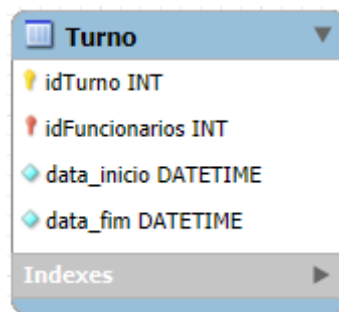


Figura 44: Tabela lógica do Turno

```
-- DROP TABLE `SuperKartsCup`.`Turno`;
CREATE TABLE `SuperKartsCup`.`Turno` (
  `idTurno` INT NOT NULL AUTO_INCREMENT,
  `idFuncionarios` INT NOT NULL,
  `data_inicio` DATETIME NOT NULL,
  `data_fim` DATETIME NOT NULL,
  PRIMARY KEY (`idTurno`, `idFuncionarios`),
  FOREIGN KEY (`idFuncionarios`)
    REFERENCES `SuperKartsCup`.`Funcionarios` (`idFuncionarios`)
);
```

Figura 45: Implementação do Turno

A tabela **Notificacao** possui como chave primária composta as colunas "idNotificacao" e "idEncomenda". A coluna "idNotificacao", definida como "INT", "NOT NULL" e "AUTO_INCREMENT", é o identificador único para cada notificação. A coluna "idEncomenda", definida como "INT" e "NOT NULL", faz referência ao identificador único de uma encomenda na tabela Encomenda, associando cada notificação a uma encomenda específica. A coluna "descricao", definida como "VARCHAR(250)" e "NOT NULL", armazena o conteúdo da notificação, permitindo uma descrição detalhada do evento ou situação que gerou a notificação. A coluna "data_notificacao", definida como "TIMESTAMP" com valor padrão "CURRENT_TIMESTAMP", registra automaticamente a data e hora da criação da notificação,

permitindo rastrear quando a notificação foi gerada. A chave estrangeira na coluna "idEncomenda" garante que a notificação esteja associada a uma encomenda válida da tabela Encomenda.

Figura 46: Tabela lógica de Notificacao

```
-- DROP TABLE `SuperKartsCup`.`Notificacao`;
CREATE TABLE `SuperKartsCup`.`Notificacao` (
  `idNotificacao` INT NOT NULL AUTO_INCREMENT,
  `descricao` VARCHAR(250) NOT NULL,
  `data_notificacao` TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  `idEncomenda` INT NOT NULL,
  PRIMARY KEY (`idNotificacao`, `idEncomenda`),
  FOREIGN KEY (`idEncomenda`)
    REFERENCES `SuperKartsCup`.`Encomenda` (`idEncomenda`)
);
```

Figura 47: Implementação de Notificacao

A tabela **Contacto_Espectador** possui como chave primária composta as colunas "idEspectadores" e "contacto". A coluna "idEspectadores", definida como "INT" e "NOT NULL", faz referência ao identificador único de um espectador na tabela Espectadores, associando cada contacto a um espectador específico. A coluna "contacto", definida como "VARCHAR(200)" e "NOT NULL", armazena as informações de contacto do espectador, que podem incluir números de telefone, e-mails, ou outros meios de comunicação. A chave estrangeira na coluna "idEspectadores" garante que o contacto esteja vinculado a um espectador válido da tabela Espectadores.

Figura 48: Tabela lógica de Contacto_Espectador


```
-- DROP TABLE `SuperKartsCup`.`Contacto_Espectador`;
CREATE TABLE `SuperKartsCup`.`Contacto_Espectador` (
  `idEspectadores` INT NOT NULL,
  `contacto` VARCHAR(200) NOT NULL,
  PRIMARY KEY (`idEspectadores`,`contacto`),
  FOREIGN KEY (`idEspectadores`)
    REFERENCES `SuperKartsCup`.`Espectadores` (`idEspectadores`)
);
```

Figura 49: Implementação de Contacto_Espectador

A última tabela a ser criada é a tabela **oferecido**, que possui como chave primária composta as colunas "idmaterialdecorrida" e "idPatrocinadores". A coluna "idmaterialdecorrida", definida como "INT" e "NOT NULL", faz referência ao identificador único de um material de corrida na tabela Material de Corrida, associando o material de corrida a um patrocinador. A coluna "idPatrocinadores", definida como "INT" e "NOT NULL", faz referência ao identificador único de um patrocinador na tabela Patrocinadores, indicando que o patrocinador oferece o material de corrida. As chaves estrangeiras garantem que tanto o material de corrida quanto o patrocinador sejam válidos e estejam corretamente associados a registros nas tabelas Material de Corrida e Patrocinadores, respetivamente.

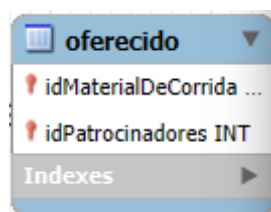


Figura 50: Tabela lógica de Oferecido

```
-- DROP TABLE `SuperKartsCup`.`oferecido`;
CREATE TABLE `SuperKartsCup`.`oferecido` (
  `idmaterialdecorrida` INT NOT NULL,
  `idPatrocinadores` INT NOT NULL,
  PRIMARY KEY (`idmaterialdecorrida`,`idPatrocinadores`),
  FOREIGN KEY (`idmaterialdecorrida`)
    REFERENCES `SuperKartsCup`.`Material de Corrida` (`idMaterialDeCorrida`),
  FOREIGN KEY (`idPatrocinadores`)
    REFERENCES `SuperKartsCup`.`Patrocinadores` (`idPatrocinadores`)
);
```

Figura 51: Implementação de Oferecido

5.2. Cálculo do espaço da base de dados

Para perceber o espaço que a base de dados do SuperKarts Cup pudesse vir a ocupar, calculamos o tamanho de cada tabela, inicialmente com um registo apenas em cada e, numa fase final, estimou-se o espaço necessário para armazenar toda a informação presente na base de dados.

De forma que fosse mais fácil realizar os respetivos cálculos, criamos uma tabela com os tipos de dados contidos na base de dados e os seus respetivos tamanhos em *bytes*.

Tipos de Dados	Tamanho (bytes)
INT	4
VARCHAR(M)	M + 1
DECIMAL(M,2)	[M/9] * 4
TINYINT	1
DATE	3
DATETIME(M)	5 + M
TIMESTAMP	4
ENUM('valor1', 'valor2', ...)	1 ou 2

Tabela 47: Tipos de dados e seus respetivos tamanhos em bytes

De seguida, criamos tabelas de forma a calcular o espaço ocupado por cada tabela lógica, se cada uma tivesse apenas um registo.

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Arbitros	idArbitros	INT	4
	Nome	VARCHAR(45)	46
	data_nasc	DATE	3
	tipo	ENUM('Pista', 'Técnico', 'Médico', 'Controlo de Corrida')	1
	morada_ rua	VARCHAR(45)	46
	morada_localidade	VARCHAR(45)	46
	morada_porta	INT	4
	morada_codigoPostal	VARCHAR(45)	46
	contacto_telemovel	INT	4
	contacto_email	VARCHAR(45)	46
	corrida_tipo	ENUM('Clássica', 'DragRace', '24', 'Rally')	1
	idEvento	INT	4
Total	-	-	251

Tabela 48: Espaço ocupado por Arbitros com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Comissarios	idComissarios	INT	4
	Nome	VARCHAR(45)	46
	data_nasc	DATE	3
	morada_ rua	VARCHAR(45)	46
	morada_localidade	VARCHAR(45)	46
	morada_porta	INT	4
	morada_codigoPostal	VARCHAR(45)	46
	contacto_telemovei	INT	4
	contacto_email	VARCHAR(45)	46
	corrida_tipo	ENUM('Clássica', 'DragRace', '24', 'Rally')	1
	idEvento	INT	4
Total	-	-	250

Tabela 49: Espaço ocupado por Comissarios com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Comunica	Funcionarios_idFuncionarios	INT	4
	Equipas_idEquipas	INT	4
Total	-	-	8

Tabela 50: Espaço ocupado por Comunica com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Contacto_Espectador	idEspectadoires	INT	4
	contacto	VARCHAR(200)	201
Total	-	-	205

Tabela 51: Espaço ocupado por Contacto_Espectador com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Corrida	tipo	ENUM('Clássica', 'DragRace', '24', 'Rally')	1
	Pista	VARCHAR(45)	46

	Evento_idEvento	INT	4
Total	-	-	51

Tabela 52: Espaço ocupado por Corrida com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Encomenda	idEncomenda	INT	4
	Valor	DECIMAL(5,2)	4
	Quantidade	INT	4
	Estado	TINYINT	1
	idFornecedor	INT	4
	idLembranca	INT	4
	idFuncionario	INT	4
Total	-	-	25

Tabela 53: Espaço ocupado por Encomenda com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Equipa	idEquipa	INT	4
	Nome	VARCHAR(45)	46
	morada_ rua	VARCHAR(45)	46
	morada_localidade	VARCHAR(45)	46
	morada_porta	INT	4
	morada_codigoPostal	VARCHAR(45)	46
	contacto_telemovei	INT	4
	contacto_email	VARCHAR(45)	46
	Pontos	INT	4
Total	-	-	246

Tabela 54: Espaço ocupado por Equipa com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Espectadores	idEspectadores	INT	4
	Nome	VARCHAR(45)	46
	data_nasc	DATE	3
	morada_ rua	VARCHAR(45)	46
	morada_localidade	VARCHAR(45)	46
	morada_porta	INT	4
	morada_codigoPostal	VARCHAR(45)	46
	contacto_telemovei	INT	4

Total	-	-	199
--------------	---	---	-----

Tabela 55: Espaço ocupado por Espectadores com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Evento	idEvento	INT	4
	Nome	VARCHAR(45)	46
	Gastos	INT	4
	data_inicio	DATE	3
	morada_ rua	VARCHAR(45)	46
	morada_localidade	VARCHAR(45)	46
	morada_porta	INT	4
	morada_codigoPostal	VARCHAR(45)	46
	contacto_telemovei	INT	4
	contacto_email	VARCHAR(45)	46
Total	-	-	249

Tabela 56: Espaço ocupado por Evento com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Fornecedor	idFornecedor	INT	4
	Nome	VARCHAR(45)	46
	Tipo_lembrança	VARCHAR(45)	46
	morada_ rua	VARCHAR(45)	46
	morada_localidade	VARCHAR(45)	46
	morada_porta	INT	4
	morada_codigoPostal	VARCHAR(45)	46
	contacto_telemovei	INT	4
	contacto_email	VARCHAR(45)	46
Total	-	-	288

Tabela 57: Espaço ocupado por Fornecedor com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
FuncionariosdaEquipa	idFuncionariosdaEquipa	INT	4
	Nome	VARCHAR(45)	46
	data_nasc	DATE	3
	tipo_funcionaria	ENUM('Chefe', 'Mecânico', 'Engenheiro', 'Analista', 'PitCrew')	1
	morada_ rua	VARCHAR(45)	46

	morada_localidade	VARCHAR(45)	46
	morada_porta	INT	4
	morada_codigoPostal	VARCHAR(45)	46
	contacto_email	VARCHAR(45)	46
	Equipas_idEquipas	INT	4
Total	-	-	246

Tabela 58: Espaço ocupado por Funcionarios da Equipa com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Funcionarios	idFuncionarios	INT	4
	Nome	VARCHAR(45)	46
	data_nasc	DATE	3
	morada_rua	VARCHAR(45)	46
	morada_localidade	VARCHAR(45)	46
	morada_porta	INT	4
	morada_codigoPostal	VARCHAR(45)	46
	contacto_email	VARCHAR(45)	46
	contacto_telemovei	INT	4
	Evento_idEvento	INT	4
Total	-	-	249

Tabela 59: Espaço ocupado por Funcionarios com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
gereF	Funcionarios_idFuncionarioGere	INT	4
	Funcionarios_idFuncionarios	INT	4
Total	-	-	8

Tabela 60: Espaço ocupado por gereF com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Lembrancas	idLembrancas	INT	4
	Nome	VARCHAR(45)	46
	Quantidade	INT	4
	Preço	DECIMAL(5,2)	4
	Evento_idEvento	INT	4
Total	-	-	62

Tabela 61: Espaço ocupado por Lembrancas com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Material de Corrida	idMaterialDeCorrida	INT	4
	Nome	VARCHAR(45)	46
	Quantidade	INT	4
	Stock_minimo	INT	4
	Equipas_idEquipas	INT	4
Total	-	-	62

Tabela 62: Espaço ocupado por Material de Corrida com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Notificação	idNotificação	INT	4
	Descrição	VARCHAR(250)	251
	Data_notificação	TIMESTAMP	4
	idEncomenda	INT	4
Total	-	-	263

Tabela 63: Espaço ocupado por Notificacao com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
oferecido	idMaterialDeCorrida	INT	4
	idPatrocinadores	INT	4
Total	-	-	8

Tabela 64: Espaço ocupado por Oferecido com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Patrocinadores	idPatrocinadoress	INT	4
	Nome	VARCHAR(45)	46
	TipoMaterial	VARCHAR(45)	46
	morada_ rua	VARCHAR(45)	46
	morada_localidade	VARCHAR(45)	46
	morada_porta	INT	4
	morada_codigoPostal	VARCHAR(45)	46
	contacto_telemovei	INT	4
	contacto_email	VARCHAR(45)	46

Total	-	-	288
--------------	---	---	-----

Tabela 65: Espaço ocupado por Patrocinadores com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Pilotos	numero_piloto	INT	4
	Nome	VARCHAR(45)	46
	data_nasc	DATE	3
	morada_ rua	VARCHAR(45)	46
	morada_localidade	VARCHAR(45)	46
	morada_porta	INT	4
	morada_codigoPostal	VARCHAR(45)	46
	contacto_email	VARCHAR(45)	46
	Equipas_idEquipas	INT	4
Total	-	-	245

Tabela 66: Espaço ocupado por Pilotos com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Proteção Civil	idProtecaoCivil	INT	4
	Nome	VARCHAR(45)	46
	Tipo	ENUM('Bombeiros', 'GNR')	1
	morada_ rua	VARCHAR(45)	46
	morada_localidade	VARCHAR(45)	46
	morada_porta	INT	4
	morada_codigoPostal	VARCHAR(45)	46
	contacto_telemovel	INT	4
	contacto_email	VARCHAR(45)	46
	Evento_idEvento	INT	4
Total	-	-	247

Tabela 67: Espaço ocupado por Protecao Civil com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Turno	idTurno	INT	4
	Data_Inicio	DATETIME	5
	Data_Fim	DATETIME	5
	idFuncionario	INT	4

Total	-	-	18
--------------	---	---	----

Tabela 68: Espaço ocupado por Turnos com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Venda	idVenda	INT	4
	Valor	DECIMAL(5,2)	4
	Data	DATETIME	5
	funcionario	INT	4
	lembrancas	INT	4
	espectadores	INT	4
Total	-	-	25

Tabela 69: Espaço ocupado por Venda com um registo

Tabela	Nome da Coluna	Tipo de Dado	Tamanho(bytes)
Classificação	Corrida_tipo	ENUM('Clássica', 'DragRace', '24', 'Rally')	1
	Posicao	INT	4
	numero_Piloto	INT	4
	idFuncionario	INT	4
	idEvento	INT	4
Total	-	-	17

Tabela 70: Espaço ocupado por Classificacao com um registo

Assim, o espaço total ocupado é igual a 3366 bytes.

Depois de realizarmos o povoamento inicial, a base de dados ocupa um total de 58141 bytes.

Tabela	Quantidade de Registos	Tamanho da Tabela (bytes)
Arbitros	20	5020
Comissarios	20	5000
Comunica	10	80
Contacto_Espectador	1	205
Corrida	4	204
Encomenda	1	25
Equipas	10	2460
Espectadores	1	199

Evento	1	249
Fornecedor	1	144
Funcionarios da Equipa	100	24600
Funcionarios	20	4980
gereF	17	136
Lembrancas	1	62
Material de Corrida	20	1240
Notificação	1	263
oferecido	10	80
Patrocinadores	2	576
Pilotos	37	9065
Proteção Civil	10	2470
Turno	21	378
Classificacao	40	680
Venda	1	25
Total	-	58141

Tabela 71: Espaço ocupado pelas tabelas após o povoamento inicial

Tendo em conta que o número de espectadores cresce cerca de 12% anualmente e cerca de 25% dos espectadores realizam compras durante o evento, sendo que a média de espectadores para a SuperKarts Cup deste ano ronda os 5000.

Visto que 25% dos espectadores realizam compras durante o evento, seriam feitas cerca de 1250 vendas o que levaria o espaço ocupado pela base de dados para os 3049579 bytes.

Em suma, a taxa de crescimento anual da base de dados seria de aproximadamente **104%**, pois como podemos ver na tabela:

Tabela	Quantidade de Registos	Tamanho da Tabela (bytes)
Arbitros	40	10040
Comissarios	40	10000
Comunica	20	160
Contacto_Espectador	18642	3821610
Corrida	8	408
Encomenda	210	5250
Equipas	20	4920
Espectadores	10600	2109400
Evento	2	498

Fornecedor	2	288
Funcionarios da Equipa	200	49200
Funcionarios gereF	40	9960
	34	272
Lembrancas	2	124
Material de Corrida	40	2480
Notificação oferecido	420	110460
	20	160
Patrocinadores	4	1152
Pilotos	77	18865
Proteção Civil	20	4940
Turno	50	900
Classificacao	80	1360
Venda	2650	66250
Total	-	6228697

Tabela 72: Estimativa do tamanho da base de dados após um ano

5.3. Realização do povoamento da base de dados

Para a povoamento da base de dados, optamos por utilizar um ficheiro SQL, onde, com a utilização de instruções “INSERT”, inserimos uma entrada modelo, para que depois, com a ajuda da inteligência artificial, podemos criar enumeras entradas com dados fictícios.

Começamos introdução de dados na tabela **Evento** e terminamos na tabela **gereF**.

```
-- DELETE FROM `SuperKartsCup`.`Evento`;
INSERT INTO `SuperKartsCup`.`Evento`
(idEvento, Nome, Gastos, data_inicio, morada_rua, morada_localidade, morada_porta, morada_codigoPostal, contacto_telemovel, contacto_email)
VALUES ('1', 'SuperKarts Cup 21', '30000', '2024-09-22', 'Sítio do Escampadinho', 'Portimão', '1', '8500-148', '911111111', 'contact@superkartscup.com');
-- SELECT * FROM `SuperKartsCup`.`Evento`;
```

Figura 52: Povoamento da tabela Evento

```
-- DELETE FROM `SuperKartsCup`.`Protecao Civil`;
INSERT INTO `SuperKartsCup`.`Protecao Civil`
(idProtecaocivil, Nome, tipo, morada_rua, morada_localidade, morada_porta, morada_codigoPostal, contacto_telemovel, contacto_email, Evento_idEvento)
VALUES
('1', 'João Matos', 'Bombeiros', 'Rua da Santa', 'Vizela', '6', '4620-525', '910067097', 'samuraidatemu@gmail.com', '1'),
('2', 'Maria Silva', 'Bombeiros', 'Avenida Central', 'Braga', '45', '4710-370', '920123456', 'maria.silva@gmail.com', '1'),
('3', 'Pedro Costa', 'GNR', 'Rua Nova', 'Porto', '10', '4000-150', '931987654', 'pedro.costa@gmail.com', '1'),
('4', 'Ana Rocha', 'Bombeiros', 'Travessa do Sol', 'Lisboa', '23', '1200-452', '917234567', 'ana.rocha@gmail.com', '1'),
('5', 'Tiago Martins', 'GNR', 'Rua do Campo', 'Coimbra', '78', '3000-200', '913456789', 'tiago.martins@gmail.com', '1'),
('6', 'Clara Mendes', 'Bombeiros', 'Praça das Flores', 'Faro', '12', '8000-072', '925678123', 'clara.mendes@gmail.com', '1'),
('7', 'Rui Pinto', 'GNR', 'Rua da Esperança', 'Guimarães', '34', '4800-192', '912345678', 'rui.pinto@gmail.com', '1'),
('8', 'Carla Monteiro', 'Bombeiros', 'Rua da Paz', 'Aveiro', '56', '3800-222', '915678234', 'carla.monteiro@gmail.com', '1'),
('9', 'Miguel Lopes', 'GNR', 'Rua do Mar', 'Setúbal', '89', '2900-316', '918123456', 'miguel.lopes@gmail.com', '1'),
('10', 'Joana Santos', 'GNR', 'Rua do Norte', 'Viana do Castelo', '66', '4900-231', '916543210', 'joana.santos@gmail.com', '1');
-- SELECT * FROM `SuperKartsCup`.`Protecao Civil`;
```

Figura 53: Povoamento da tabela Protecao Civil

```
-- DELETE FROM `SuperKartsCup`.`corrida`;
INSERT INTO `SuperKartsCup`.`corrida`
(tipo, pista, idevento)
VALUES
('Clássica', 'Autódromo Internacional do Algarve',1),
('DragRace', 'Autódromo Internacional do Algarve',1),
('24', 'Autódromo Internacional do Algarve',1),
('Rally', 'Autódromo Internacional do Algarve',1);
-- SELECT * FROM `SuperKartsCup`.`corrida`;
```

Figura 54: Povoamento da tabela Corrida

```
-- DELETE FROM `SuperKartsCup`.`Arbitro`;
INSERT INTO `SuperKartsCup`.`Arbitro`
(idArbitro, Nome, data_nasc, morada_rua, morada_localidade, morada_porta, morada_codigoPostal, contacto_telemovel, contacto_email, corrida_tipo,idEvento)
VALUES
(1, 'Artur Soares Dias', '1979-07-14', 'Rua da Pontinha', 'Vilar das Almas', '13', '4990-790', '910000001', 'arturito@gmail.com', 'Clássica',1),
(2, 'Nuno Almeida', '1982-05-22', 'Rua das Laranjeiras', 'Lisboa', '8', '1200-190', '910000002', 'nuno.almeida@gmail.com', 'Clássica',1),
(3, 'Tiago Martins', '1980-03-19', 'Rua do Arco', 'Sintra', '21', '2710-120', '910000003', 'tiago.martins@gmail.com', 'Clássica',1),
(4, 'Helder Malheiro', '1985-09-25', 'Rua do Mirante', 'Porto', '15', '4000-250', '910000004', 'helder.malheiro@gmail.com', 'Clássica',1),
(5, 'Fábio Veríssimo', '1983-11-11', 'Rua das Fontes', 'Coimbra', '10', '3000-800', '910000005', 'fabio.verissimo@gmail.com', 'Clássica',1),
(6, 'João Capela', '1977-08-01', 'Rua do Campo', 'Faro', '5', '8000-150', '910000006', 'joao.capela@gmail.com', 'DragRace',1),
(7, 'Jorge Sousa', '1984-02-10', 'Rua Nova', 'Braga', '12', '4700-320', '910000007', 'jorge.sousa@gmail.com', 'DragRace',1),
(8, 'Hugo Miguel', '1986-06-15', 'Rua das Flores', 'Aveiro', '4', '3800-250', '910000008', 'hugo.miguel@gmail.com', 'DragRace',1),
(9, 'Carlos Xistra', '1980-12-05', 'Rua da Pedra', 'Guimarães', '19', '4800-600', '910000009', 'carlos.xistra@gmail.com', 'DragRace',1),
(10, 'Bruno Paixão', '1975-04-09', 'Rua do Pinhal', 'Leiria', '6', '2400-300', '910000010', 'bruno.paixao@gmail.com', 'DragRace',1),
(11, 'Manuel Mota', '1981-01-30', 'Rua da Fonte', 'Setúbal', '7', '2900-150', '910000011', 'manuel.mota@gmail.com', '24',1),
(12, 'António Nobre', '1989-04-12', 'Rua do Sol', 'Almada', '9', '2800-500', '910000012', 'antonio.nobre@gmail.com', '24',1),
(13, 'Luís Godinho', '1988-07-22', 'Rua dos Anjos', 'Évora', '17', '7000-600', '910000013', 'luis.godinho@gmail.com', '24',1),
(14, 'André Narciso', '1990-05-05', 'Rua do Cruzeiro', 'Tomar', '11', '2300-700', '910000014', 'andre.narciso@gmail.com', '24',1),
(15, 'Rui Costa', '1978-09-14', 'Rua dos Passarinhos', 'Funchal', '14', '9000-800', '910000015', 'rui.costa@gmail.com', '24',1),
(16, 'Pedro Proença', '1970-11-03', 'Rua das Hortas', 'Lisboa', '2', '1100-150', '910000016', 'pedro.proenca@gmail.com', 'Rally',1),
(17, 'Duarte Gomes', '1983-03-18', 'Rua do Castelo', 'Porto', '18', '4050-600', '910000017', 'duarte.gomes@gmail.com', 'Rally',1),
(18, 'Vitor Ferreira', '1979-06-09', 'Rua da Serra', 'Viseu', '20', '3500-700', '910000018', 'vitor.ferreira@gmail.com', 'Rally',1),
(19, 'Carlos Macedo', '1987-12-24', 'Rua do Vale', 'Guarda', '3', '6300-800', '910000019', 'carlos.macedo@gmail.com', 'Rally',1),
(20, 'Miguel Ferreira', '1982-08-17', 'Rua do Prado', 'Beja', '22', '7800-900', '910000020', 'miguel.ferreira@gmail.com', 'Rally',1);
-- SELECT * FROM `SuperKartsCup`.`Arbitro`;
```

Figura 55: Povoamento da tabela Arbitro

```
-- DELETE FROM `SuperKartsCup`.`Comissarios`;
INSERT INTO `SuperKartsCup`.`Comissarios`
(idComissarios, Nome, data_nasc, tipo, morada_rua, morada_localidade, morada_porta, morada_codigoPostal, contacto_telemovel, contacto_email, corrida_tipo,idEvento)
VALUES
(1, 'Ricardo Araujo Pereira', '1974-04-28', 'Pista', 'Rua da Fonte', 'Aguçadoura', '1', '4495-001', '910000021', 'ricardofunny@gmail.com', 'Clássica',1),
(2, 'Joana Marques', '1985-03-15', 'Técnico', 'Rua do Farol', 'Lisboa', '12', '1100-002', '910000022', 'joana.humorista@gmail.com', 'Clássica',1),
(3, 'António Zambujo', '1975-09-19', 'Médico', 'Rua das Oliveiras', 'Beja', '34', '7800-003', '910000023', 'zambujo.cantor@gmail.com', 'Clássica',1),
(4, 'Mariza Reis Nunes', '1973-12-16', 'Controlo de Corrida', 'Rua do Mar', 'Faro', '7', '8000-004', '910000024', 'mariza.fado@gmail.com', 'Clássica',1),
(5, 'Carlos Daniel', '1972-07-23', 'Técnico', 'Rua do Jornal', 'Porto', '21', '4000-005', '910000025', 'carlos.noticias@gmail.com', 'Clássica',1),
(6, 'Manuel Luis Goucha', '1954-12-25', 'Médico', 'Rua do Sol', 'Sintra', '3', '2710-006', '910000026', 'mlgoucha@gmail.com', 'DragRace',1),
(7, 'Cristina Ferreira', '1977-09-09', 'Pista', 'Rua das Flores', 'Mafra', '15', '2640-007', '910000027', 'cristina.tv@gmail.com', 'DragRace',1),
(8, 'Cláudia Vieira', '1978-06-20', 'Controlo de Corrida', 'Rua da Alegria', 'Coimbra', '25', '3000-008', '910000028', 'claudia.atriz@gmail.com', 'DragRace',1),
(9, 'José Rodrigues dos Santos', '1964-04-01', 'Técnico', 'Rua da História', 'Almada', '17', '2800-009', '910000029', 'jrs.jornal@gmail.com', 'DragRace',1),
(10, 'Filomena Cautela', '1984-12-16', 'Pista', 'Rua do Teatro', 'Cascais', '19', '2750-010', '910000030', 'filomena.apresentadora@gmail.com', 'DragRace',1),
(11, 'Pedro Teixeira', '1980-12-23', 'Controlo de Corrida', 'Rua das Palmeiras', 'Évora', '9', '7000-011', '910000031', 'pedro.ator@gmail.com', '24',1),
(12, 'Ana Bola', '1952-06-02', 'Técnico', 'Rua do Humor', 'Lisboa', '10', '1200-012', '910000032', 'ana.humorista@gmail.com', '24',1),
(13, 'Herman José', '1954-03-19', 'Médico', 'Rua das Estrelas', 'Porto', '32', '4100-013', '910000033', 'herman.icone@gmail.com', '24',1),
(14, 'Daniel Oliveira', '1981-04-01', 'Pista', 'Rua dos Programas', 'Oeiras', '26', '2780-014', '910000034', 'daniel.entrevistas@gmail.com', '24',1),
(15, 'Tânia Ribas de Oliveira', '1976-06-18', 'Técnico', 'Rua das Criancas', 'Viseu', '8', '3500-015', '910000035', 'tania.apresentadora@gmail.com', '24',1),
(16, 'José Carlos Malato', '1964-03-06', 'Pista', 'Rua da Rádio', 'Ponte de Lima', '5', '4990-016', '910000036', 'malato.radio@gmail.com', 'Rally',1),
(17, 'Fátima Lopes', '1969-05-13', 'Técnico', 'Rua do Estilo', 'Setúbal', '11', '2900-017', '910000037', 'fatima.moda@gmail.com', 'Rally',1),
(18, 'Ana Moura', '1979-09-17', 'Controlo de Corrida', 'Rua da Música', 'Alcobaça', '14', '2460-018', '910000038', 'anamoura.fado@gmail.com', 'Rally',1),
(19, 'Fernando Mendes', '1963-03-09', 'Médico', 'Rua da Sorte', 'Portalegre', '22', '7300-019', '910000039', 'fernando.preco@gmail.com', 'Rally',1),
(20, 'Maria Cerqueira Gomes', '1983-05-06', 'Pista', 'Rua das Hortas', 'Braga', '6', '4700-020', '910000040', 'maria.tv@gmail.com', 'Rally',1);
-- SELECT * FROM `SuperKartsCup`.`Comissarios`;
```

Figura 56: Povoamento da tabela Comissarios

```
-- DELETE FROM `SuperKartsCup`.`Funcionarios`;
INSERT INTO `SuperKartsCup`.`Funcionarios`
(idFuncionarios, Nome, data_nasc, morada_rua, morada_localidade, morada_porta, morada_codigoPostal, contacto_telemovel, contacto_email, evento_idevento)
VALUES
(1, 'Jorge Nuno Pinto da Costa', '1937-12-28', 'Rua da Fruta', 'Porto', '2', '1405-043', '910000003', 'ananas@gmail.com', 1),
(2, 'Luis Felipe Vieira', '1949-07-22', 'Rua da Farinha', 'Lisboa', '1', '1000-001', '910000004', 'branca@gmail.com', 1),
(3, 'Ana Sousa', '1985-03-15', 'Rua das Flores', 'Porto', '8', '4050-123', '912345678', 'ana.sousa@gmail.com', 1),
(4, 'Carlos Oliveira', '1978-06-12', 'Avenida Principal', 'Lisboa', '22', '1000-456', '919876543', 'carlos.oliveira@gmail.com', 1),
(5, 'Marta Ferreira', '1990-11-03', 'Rua do Sol', 'Faro', '15', '8000-890', '913213214', 'marta.ferreira@gmail.com', 1),
(6, 'Rui Moreira', '1969-07-28', 'Praça da Liberdade', 'Porto', '3', '4000-123', '914567890', 'rui.moreira@gmail.com', 1),
(7, 'Joana Lima', '1988-04-25', 'Rua Nova', 'Coimbra', '12', '3000-789', '915678321', 'joana.lima@gmail.com', 1),
(8, 'Ricardo Costa', '1982-09-14', 'Rua das Amoreiras', 'Lisboa', '21', '1200-678', '916789432', 'ricardo.costa@gmail.com', 1),
(9, 'Sofia Martins', '1993-01-19', 'Rua da Paz', 'Aveiro', '9', '3800-890', '917890123', 'sofia.martins@gmail.com', 1),
(10, 'Luís Rodrigues', '1975-10-05', 'Rua da Alegria', 'Setúbal', '7', '2900-234', '918901234', 'luis.rodrigues@gmail.com', 1),
(11, 'Patrícia Mendes', '1987-08-08', 'Rua do Carmo', 'Braga', '18', '4700-345', '919012345', 'patricia.mendes@gmail.com', 1),
(12, 'Tiago Almeida', '1995-05-30', 'Rua das Palmeiras', 'Guimarães', '6', '4800-567', '920123456', 'tiago.almeida@gmail.com', 1),
(13, 'Diana Figueiredo', '1981-12-02', 'Rua da Luz', 'Porto', '20', '4050-678', '921234567', 'diana.figueiredo@gmail.com', 1),
(14, 'Filipe Nunes', '1976-03-10', 'Avenida da República', 'Lisboa', '1', '1000-890', '922345678', 'filipe.nunes@gmail.com', 1),
(15, 'Beatriz Vieira', '1992-06-15', 'Rua da Escola', 'Faro', '14', '8000-456', '923456789', 'beatriz.vieira@gmail.com', 1),
(16, 'André Carvalho', '1984-09-21', 'Rua do Parque', 'Coimbra', '33', '3000-123', '924567890', 'andre.carvalho@gmail.com', 1),
(17, 'Sara Batista', '1996-02-11', 'Praça Nova', 'Lisboa', '30', '1200-345', '925678901', 'sara.batista@gmail.com', 1),
(18, 'Leonor Cruz', '1989-07-01', 'Rua da Serra', 'Braga', '27', '4700-890', '926789012', 'leonor.cruz@gmail.com', 1),
(19, 'Manuel Matos', '1983-11-22', 'Rua do Mercado', 'Setúbal', '5', '2900-678', '927890123', 'manuel.matos@gmail.com', 1),
(20, 'João Fonseca', '1980-05-07', 'Rua da Fonte', 'Guimarães', '4', '4800-890', '929012345', 'joao.fonseca@gmail.com', 1);
-- SELECT * FROM `SuperKartsCup`.`Funcionarios`;
```

Figura 57: Povoamento da tabela Funcionarios

```

-- DELETE FROM `SuperKartsCup`.`turno`;
INSERT INTO `SuperKartsCup`.`turno`
(idturno, idfuncionarios, data_inicio, data_fim)
VALUES
(1, 1, '2024-09-22 06:00:00', '2024-09-22 13:00:00'),
(2, 2, '2024-09-22 06:00:00', '2024-09-22 13:00:00'),
(3, 3, '2024-09-23 06:00:00', '2024-09-23 13:00:00'),
(4, 4, '2024-09-23 06:00:00', '2024-09-23 13:00:00'),
(5, 5, '2024-09-24 06:00:00', '2024-09-24 13:00:00'),
(6, 6, '2024-09-24 06:00:00', '2024-09-24 13:00:00'),
(7, 7, '2024-09-25 06:00:00', '2024-09-25 13:00:00'),
(8, 8, '2024-09-25 06:00:00', '2024-09-25 13:00:00'),
(9, 9, '2024-09-26 06:00:00', '2024-09-26 13:00:00'),
(10, 10, '2024-09-26 06:00:00', '2024-09-26 13:00:00'),
(11, 11, '2024-09-22 13:00:00', '2024-09-22 20:00:00'),
(12, 12, '2024-09-22 13:00:00', '2024-09-22 20:00:00'),
(13, 13, '2024-09-23 13:00:00', '2024-09-23 20:00:00'),
(14, 14, '2024-09-23 13:00:00', '2024-09-23 20:00:00'),
(15, 15, '2024-09-24 13:00:00', '2024-09-24 20:00:00'),
(16, 16, '2024-09-24 13:00:00', '2024-09-24 20:00:00'),
(17, 17, '2024-09-25 13:00:00', '2024-09-25 20:00:00'),
(18, 18, '2024-09-25 13:00:00', '2024-09-25 20:00:00'),
(19, 19, '2024-09-26 13:00:00', '2024-09-26 20:00:00'),
(20, 20, '2024-09-26 13:00:00', '2024-09-26 20:00:00'),
(21, 1, '2024-09-27 06:00:00', '2024-09-27 13:00:00');
-- SELECT * FROM `SuperKartsCup`.`turno`;

```

Figura 58: Povoamento da tabela Turno

```

-- DELETE FROM `SuperKartsCup`.`espectadores`;
INSERT INTO `SuperKartsCup`.`espectadores`
(idEspectadores, Nome, data_nasc, morada_rua, morada_localidade, morada_porta, morada_codigoPostal)
VALUES ('1', 'David Beckham', '1975-05-02', 'Rua do Relógio', 'Perre', '7', '4925-001');
-- SELECT * FROM `SuperKartsCup`.`espectadores`;

```

Figura 59: Povoamento da tabela Espectadores

```

-- DELETE FROM `SuperKartsCup`.`contacto_espectador`;
INSERT INTO `SuperKartsCup`.`contacto_espectador`
(idEspectadores, contacto)
VALUES
('1', '917021126'),
('1', '967145522');
-- SELECT * FROM `SuperKartsCup`.`contacto_espectador`;

```

Figura 60: Povoamento da tabela Contacto_Espectador

```
-- DELETE FROM `SuperKartsCup`.`Equipas`;
INSERT INTO `SuperKartsCup`.`Equipas`
(idEquipas, Nome, morada_rua, morada_localidade, morada_porta, morada_codigoPostal, contacto_telemovei, contacto_email)
VALUES
(1, 'RedBull Portugal', 'Rua da Tourina', 'Porto', '9', '1405-043', '914001665', 'redbull@contact.com'),
(2, 'PORSCHÉ', 'Rua de Merece', 'Calvelo', '1', '4990-580', '910011213', 'porsche@contact.com'),
(3, 'AUDI', 'Rua da Di', 'Porto', '134', '1405-042', '914076665', 'audi@contact.com'),
(4, 'Mercedes', 'Rua da Meca', 'Porto', '14', '1405-041', '915643665', 'mercedes@contact.com'),
(5, 'Ferrari', 'Rua da Fe', 'Porto', '873', '1405-040', '914001543', 'ferrari@contact.com'),
(6, 'Sauber', 'Rua da Saude', 'Lisboa', '923', '1000-012', '914005435', 'sauber@contact.com'),
(7, 'Renault', 'Rua da Torre', 'Viseu', '39', '1505-043', '923001665', 'refr@contact.com'),
(8, 'Lotus', 'Rua da Ponte', 'Porto', '45', '1405-156', '912991665', 'lotus@contact.com'),
(9, 'McLaren', 'Rua do Mc', 'Braga', '23', '2021-043', '914010556', 'mclaren@contact.com'),
(10, 'Delta Café', 'Rua da Cafeina', 'Lisboa', '6273', '1000-001', '914001695', 'delta@contact.com');
-- SELECT * FROM `SuperKartsCup`.`Equipas`;
```

Figura 61: Povoamento da tabela Equipas

As tabelas Funcionarios da Equipa, Pilotos e Classificacao, não se encontram completas, pois como têm 100, 37 e 40 entradas, respetivamente, não era possível tirar um *screenshot* capaz de captar todas as entradas das mesmas.

```
-- DELETE FROM `SuperKartsCup`.`Funcionarios da Equipa`;
INSERT INTO `SuperKartsCup`.`Funcionarios da Equipa`
(idFuncionariosDaEquipas, Nome, data_nasc, tipo_funcequipa, morada_rua, morada_localidade, morada_porta, morada_codigoPostal, contacto_telemovei, contacto_email, equipas_idequipas)
VALUES
(1, 'Jorge Palma', '1937-12-28', 'Chefe', 'Rua do Pão', 'Porto', '4', '1405-043', '910000014', 'palmas@gmail.com', 1),
(2, 'Marta Vieira', '1985-05-10', 'Mecânico', 'Rua das Rosas', 'Porto', '10', '1405-100', '920000014', 'martavieira@gmail.com', 1),
(3, 'Paulo Costa', '1990-07-15', 'Engenheiro', 'Avenida Central', 'Porto', '25', '1405-200', '930000014', 'paulocosta@gmail.com', 1),
(4, 'Sofia Andrade', '1982-02-20', 'Analista', 'Rua das Flores', 'Porto', '33', '1405-300', '940000014', 'sofiaandrade@gmail.com', 1),
(5, 'Carlos Nunes', '1979-09-12', 'PitCrew', 'Rua da Torre', 'Porto', '12', '1405-400', '950000014', 'carlosnunes@gmail.com', 1),
(6, 'Ana Moreira', '1995-11-25', 'Mecânico', 'Rua do Sol', 'Porto', '44', '1405-500', '960000014', 'anamoreira@gmail.com', 1),
(7, 'João Almeida', '1986-03-30', 'Mecânico', 'Rua do Mar', 'Porto', '7', '1405-600', '970000014', 'joaoalmeida@gmail.com', 1),
(8, 'Maria Silva', '1992-04-12', 'Engenheiro', 'Rua da Lua', 'Porto', '18', '1405-700', '980000014', 'mariasilva@gmail.com', 1),
(9, 'Rui Martins', '1986-08-18', 'Analista', 'Rua do Vento', 'Porto', '29', '1405-800', '990000014', 'ruimartins@gmail.com', 1),
(10, 'Cláudia Sousa', '1991-06-05', 'PitCrew', 'Rua das Ondas', 'Porto', '2', '1405-900', '910000015', 'claudiasousa@gmail.com', 1),
(11, 'Sergio Godinho', '1945-08-31', 'Chefe', 'Rua da Mala', 'Lisboa', '123', '1000-001', '910000904', 'serjao@gmail.com', 2),
```

Figura 62: Povoamento da tabela Funcionarios da Equipa

```
-- DELETE FROM `SuperKartsCup`.`Pilotos`;
INSERT INTO `SuperKartsCup`.`Pilotos`
(numero_piloto, Nome, data_nasc, morada_rua, morada_localidade, morada_porta, morada_codigoPostal, contacto_telemovei, contacto_email, equipas_idequipas)
VALUES
(1, 'Max Verstappen', '1997-09-30', 'Rua da Erva', 'Porto', '9', '1405-043', '910001665', 'eusouveloz@gmail.com', 1),
(23, 'Luís Correia', '2002-01-15', 'Rua das Nuvens', 'Porto', '12', '1405-150', '910002001', 'luis.correia@gmail.com', 1),
(7, 'Carla Mendes', '2001-11-22', 'Rua do Horizonte', 'Porto', '7', '1405-200', '910002002', 'carla.mendes@gmail.com', 1),
(3, 'Ivo Costa Sousa', '2004-09-15', 'Rua de Merece', 'Calvelo', '85', '4990-580', '913071038', 'ivoo.c.sousa@gmail.com', 2),
(5, 'Tiago Pereira', '1998-07-12', 'Rua do Sol', 'Lisboa', '25', '1000-010', '910003001', 'tiago.pereira@gmail.com', 2),
(6, 'Ana Silva', '1996-04-30', 'Rua das Flores', 'Lisboa', '18', '1000-015', '910003002', 'ana.silva@gmail.com', 2),
(88, 'João Rodrigues', '2003-06-20', 'Rua Nova', 'Lisboa', '30', '1000-020', '910003003', 'joao.rodrigues@gmail.com', 2),
(45, 'Beatriz Lima', '1999-05-19', 'Rua do Campo', 'Braga', '10', '4700-003', '910004001', 'beatriz.lima@gmail.com', 3),
(91, 'Pedro Fernandes', '2000-08-09', 'Rua da Montanha', 'Braga', '5', '4700-008', '910004002', 'pedro.fernandes@gmail.com', 3),
```

Figura 63: Povoamento da tabela Pilotos

```
-- DELETE FROM `SuperKartsCup`.`Classificacao`;
INSERT INTO `SuperKartsCup`.`Classificacao`
(idEvento, Corrida_tipo, Posicao, numero_Piloto, Pontos)
VALUES
(1,'Clássica', 1, 3, 10),
(1,'Clássica', 2, 1, 8),
(1,'Clássica', 3, 45, 7),
(1,'Clássica', 4, 12, 6),
(1,'Clássica', 5, 41, 5),
(1,'Clássica', 6, 36, 4),
(1,'Clássica', 7, 51, 3),
(1,'Clássica', 8, 56, 2),
(1,'Clássica', 9, 61, 1),
(1,'Clássica', 10, 67, 0),
(1,'DragRace', 1, 13, 9),
(1,'DragRace', 2, 5, 7),
(1,'DragRace', 3, 23, 3),
(1,'DragRace', 4, 91, 3),
(1,'DragRace', 5, 68, 0),
(1,'DragRace', 6, 52, 0),
(1,'DragRace', 7, 62, 0),
(1,'DragRace', 8, 41, 0),
(1,'DragRace', 9, 37, 0),
(1,'DragRace', 10, 58, 0),
(1,'24', 1, 1, 9),
(1,'24', 2, 88, 8),
(1,'24', 3, 14, 7),
(1,'24', 4, 4, 7),
(1,'24', 5, 46, 5),
```

Figura 64: Povoamento da tabela Classificacao


```

-- DELETE FROM `SuperKartsCup`.`material de corrida`;
INSERT INTO `SuperKartsCup`.`material de corrida`
(idMaterialDeCorrida, Nome, Quantidade, stock_minimo, equipas_idequipas)
VALUES
('1', 'Pneus', '40', '4', '1'),
('2', 'Gasolina', '10', '1', '1'),
('3', 'Pneus', '40', '4', '2'),
('4', 'Gasolina', '10', '1', '2'),
('5', 'Pneus', '40', '4', '3'),
('6', 'Gasolina', '10', '1', '3'),
('7', 'Pneus', '40', '4', '4'),
('8', 'Gasolina', '10', '1', '4'),
('9', 'Pneus', '40', '4', '5'),
('10', 'Gasolina', '10', '1', '5'),
('11', 'Pneus', '40', '4', '6'),
('12', 'Gasolina', '10', '1', '6'),
('13', 'Pneus', '40', '4', '7'),
('14', 'Gasolina', '10', '1', '7'),
('15', 'Pneus', '40', '4', '8'),
('16', 'Gasolina', '10', '1', '8'),
('17', 'Pneus', '40', '4', '9'),
('18', 'Gasolina', '10', '1', '9'),
('19', 'Pneus', '40', '4', '10'),
('20', 'Gasolina', '10', '1', '10');
-- SELECT * FROM `SuperKartsCup`.`material de corrida`;

```

Figura 65: Povoamento da tabela Material de Corrida

```

-- DELETE FROM `SuperKartsCup`.`Patrocinadores`;
INSERT INTO `SuperKartsCup`.`Patrocinadores`
(idPatrocinadores, Nome, TipoMaterial, morada_rua, morada_localidade, morada_porta, morada_codigoPostal, contacto_telemovei, contacto_email)
VALUES
('1', 'Pneus Malheiro', 'Pneus', 'Rua da Borracha', 'Porto', '799', '1405-043', '910020023', 'penusmalheiro@gmail.com'),
('2', 'Repsol', 'Gasolina', 'Rua do Negro', 'Lisboa', '867', '1000-001', '963133789', 'repsol@gmail.com');
-- SELECT * FROM `SuperKartsCup`.`Patrocinadores`;

```

Figura 66: Povoamento da tabela Patrocinadores

```

-- DELETE FROM `SuperKartsCup`.`lembrancas`;
INSERT INTO `SuperKartsCup`.`lembrancas`
(idLembrancas, Nome, Quantidade, Preço, evento_idevento)
VALUES ('1', 'Camisola', '100', '25', '1');
-- SELECT * FROM `SuperKartsCup`.`lembrancas`;

```

Figura 67: Povoamento da tabela Lembrancas

```
-- DELETE FROM `SuperKartsCup`.`venda`;
INSERT INTO `SuperKartsCup`.`venda`
  (idVenda, Valor, Data_venda, funcionario, lembrancas, espectadores)
VALUES ('1', '75', '2024-11-03 11:32:42', '1', '1', '1');
-- SELECT * FROM `SuperKartsCup`.`venda`;
```

Figura 68: Povoamento da tabela Venda

```
-- DELETE FROM `SuperKartsCup`.`fornecedor`;
INSERT INTO `SuperKartsCup`.`fornecedor`
  (idFornecedor, Nome, tipo_lembranca, morada_rua, morada_localidade, morada_porta, morada_codigoPostal, contacto_telemovei, contacto_email)
VALUES ('1', 'GUCCI', 'Camisola', 'Rua da Italia', 'Porto', '66', '1405-043', '910888666', 'guxiguxi@gmail.com');
-- SELECT * FROM `SuperKartsCup`.`fornecedor`;
```

Figura 69: Povoamento da tabela Fornecedor

```
-- DELETE FROM `SuperKartsCup`.`encomenda`;
INSERT INTO `SuperKartsCup`.`encomenda`
  (idEncomenda, Valor, Quantidade, Estado, idfuncionario, idfornecedor, idlembranca)
VALUES ('1', '500', '100', '1', '2', '1', '1');
-- SELECT * FROM `SuperKartsCup`.`encomenda`;
```

Figura 70: Povoamento da tabela Encomenda

```

-- DELETE FROM `SuperKartsCup`.`oferecido`;
INSERT INTO `SuperKartsCup`.`oferecido`
(idmaterialdecorrida, idPatrocinadores)
VALUES
('1', '1'),
('2', '2'),
('3', '1'),
('4', '2'),
('5', '1'),
('6', '2'),
('7', '1'),
('8', '2'),
('9', '1'),
('10', '2'),
('11', '1'),
('12', '2'),
('13', '1'),
('14', '2'),
('15', '1'),
('16', '2'),
('17', '1'),
('18', '2'),
('19', '1'),
('20', '2');
-- SELECT * FROM `SuperKartsCup`.`oferecido`;

```

Figura 71: Povoamento da tabela Oferecido

```

-- DELETE FROM `SuperKartsCup`.`comunica`;
INSERT INTO `SuperKartsCup`.`comunica`
(funcionarios_idfuncionarios, equipes_idequipes)
VALUES
('1', '1'),
('2', '2'),
('3', '3'),
('4', '4'),
('5', '5'),
('6', '6'),
('7', '7'),
('8', '8'),
('9', '9'),
('10', '10');
-- SELECT * FROM `SuperKartsCup`.`comunica`;

```

Figura 72: Povoamento da tabela Comunica

```
-- DELETE FROM `SuperKartsCup`.`gereF`;
INSERT INTO `SuperKartsCup`.`gereF`
(funcionarios_idfuncionariogere,funcionarios_idfuncionarios)
VALUES
('2', '1');
-- SELECT * FROM `SuperKartsCup`.`gereF`;
```

Figura 73: Povoamento da tabela gereF

5.4. Tradução das interrogações do utilizador em SQL

Neste tópico, foram analisados todos os requisitos de maneira a apresentar detalhadamente as consultas SQL. Estas consultas são utilizadas para extrair, de forma relevante, informações da base de dados do evento. Cada interrogação foi cuidadosamente projetada para responder a perguntas-chave relacionadas à gestão e organização do evento, abordando dados sobre equipas, vendas, stock de artigos, árbitros e fornecedores. Através destas consultas, é possível obter uma visão estruturada e precisa das informações armazenadas, permitindo uma análise eficaz dos dados e um suporte fundamentado à tomada de decisões no contexto do evento.

A primeira consulta mostra quais equipas estão registadas para um evento específico, unindo as tabelas Equipas, Pilotos e Classificacao. Relaciona-se com os requisitos RD6 e RD7, onde se mencionam as características das equipas e sua existência no evento.

```
-- Quais são as equipas registadas para o evento?
SELECT DISTINCT
Equipas.idEquipas,
Equipas.Nome AS NomeEquipa
FROM
Equipas
INNER JOIN
Pilotos ON Equipas.idEquipas = Pilotos.Equipas_idEquipas
INNER JOIN
Classificacao ON Pilotos.numero_Piloto = Classificacao.numero_Piloto
WHERE
Classificacao.idEvento = 1
ORDER BY
Equipas.idEquipas;
```

Figura 74: Mostrar as equipas registadas para um evento específico

A segunda consulta mostra o histórico de vendas, composto por id e nome do funcionário, id, valor e data da venda, id e nome da respetiva lembrança e id e nome do espectador que realizou a compra, de um funcionário específico através do seu nome. Esta consulta faz-se através da união as tabelas Venda, Funcionarios, Lembrancas e Espectadores. Enquadra-se

nos requisitos RD14, RM19 e RC20, que tratam de vendas de artigos turísticos, histórico de vendas, e gestão por um usuário específico.

```
-- Qual é o histórico de vendas realizadas por um funcionário específico?
SELECT
    Funcionarios.idFuncionarios,
    Funcionarios.Nome AS NomeFuncionario,
    Venda.idVenda,
    Venda.Valor,
    Venda.Data_venda,
    Venda.lembrecas,
    lembrecas.nome AS NomeLembrecas,
    Venda.espectadores,
    Espectadores.Nome AS NomeEspectador
FROM
    Venda
INNER JOIN
    Funcionarios ON Venda.funcionario = Funcionarios.idFuncionarios
INNER JOIN
    Lembrecas ON Venda.lembrecas = Lembrecas.idLembrecas
INNER JOIN
    Espectadores ON Venda.espectadores = Espectadores.idEspectadores
WHERE
    Funcionarios.Nome = 'Jorge Nuno Pinto da Costa';
```

Figura 75: Mostrar o histórico de vendas realizadas por um funcionário

A terceira consulta lista todos os artigos disponíveis em stock e suas respectivas quantidades para um evento específico. Está ligado ao requisito RM24, que se refere à gestão do stock de artigos.

```
-- Quais são os artigos disponíveis em stock e suas respectivas quantidades para um evento específico?
SELECT
    Lembrecas.idLembrecas,
    Lembrecas.Nome,
    Lembrecas.Quantidade,
    Lembrecas.Preço
FROM
    Lembrecas
WHERE
    Lembrecas.Quantidade > 0
    AND Lembrecas.Evento_idEvento = 1;
```

Figura 76: Listar todos os artigos em stock e suas respectivas quantidades de um evento

A quarta consulta mostra os árbitros designados para uma prova específica de um evento específico. Relaciona-se com RD36 e RD38, que indicam a necessidade de árbitros para provas e suas características.

```

-- Que árbitros estão designados para uma prova específica de um evento específico?
SELECT
    Arbitro.idArbitro,
    Arbitro.Nome,
    Arbitro.data_nasc,
    Arbitro.morada_rua,
    Arbitro.morada_localidade,
    Arbitro.morada_porta,
    Arbitro.morada_codigoPostal,
    Arbitro.contacto_telemovel,
    Arbitro.contacto_email
FROM
    Arbitro
WHERE
    Arbitro.idEvento = 1
    AND Arbitro.Corrída_Tipo = 'Clássica';

```

Figura 77: Mostrar os árbitros designados para uma prova específica de um evento

A última consulta, lista todos os fornecedores, com todas as suas informações, que têm encomendas pendentes, mostrando também, obviamente, toda a informação da encomenda correspondente. Este processo só é possível devido à união entre as tabelas Encomenda, Lembranca e Fornecedor. Refere-se aos requisitos RD26, RM32 e RC34, que tratam de fornecedores, gestão de encomendas, e notificações sobre atrasos ou encomendas pendentes.

```

-- Quais são os dados dos fornecedores com encomendas pendentes?
SELECT
    Encomenda.idEncomenda,
    Encomenda.Valor,
    Encomenda.Quantidade,
    Encomenda.idLembranca,
    Lembrancas.Nome AS NomeLembranca,
    Encomenda.idfornecedor,
    fornecedor.Nome,
    fornecedor.tipo_lembanca,
    fornecedor.morada_rua,
    fornecedor.morada_localidade,
    fornecedor.morada_porta,
    fornecedor.morada_codigoPostal,
    fornecedor.contacto_telemovei,
    fornecedor.contacto_email
FROM
    Encomenda
INNER JOIN
    Lembrancas ON Encomenda.idLembranca = Lembrancas.idLembrancas
INNER JOIN
    Fornecedor ON Encomenda.idfornecedor = Fornecedor.idFornecedor
WHERE
    Encomenda.Estado = 1;

```

Figura 78: Listar todos os fornecedores com encomendas pendentes

5.5. Definição e caracterização das vistas de utilização em SQL

A definição e caracterização de vistas é essencial para uma utilização eficiente de uma base de dados. Estas vistas foram criadas com o propósito de fornecer informações importantes sobre as equipas, sobre os funcionários responsáveis pela comunicação das equipas, sobre as encomendas e sobre os pilotos. Para a caracterização das mesmas, foi usado um conjunto de seleções e junções das tabelas da base de dados.

A primeira vista cria uma tabela virtual que contém todas as informações relacionadas com a estrutura de uma equipa, fornecendo id, nome, morada, contacto, nome dos pilotos e funcionários pertencentes à mesma.

```

-- DROP VIEW InfoEquipas;
CREATE VIEW InfoEquipas AS
SELECT
    Equipas.idEquipas,
    Equipas.Nome AS NomeEquipa,
    Equipas.morada_rua,
    Equipas.morada_localidade,
    Equipas.morada_porta,
    Equipas.morada_codigoPostal,
    Equipas.contacto_telemovel,
    Equipas.contacto_email,
    Pilotos.Nome AS NomePiloto,
    `Funcionarios da Equipa`.Nome AS NomeFuncionario,
    `Funcionarios da Equipa`.tipo_funcequipa
FROM
    Equipas
INNER JOIN
    Pilotos ON Equipas.idEquipas = Pilotos.Equipas_idEquipas
INNER JOIN
    `Funcionarios da Equipa` ON Equipas.idEquipas = `Funcionarios da Equipa`.Equipas_idEquipas;

```

Figura 79: Vista que mostra todas as informações de uma determinada equipa

A segunda vista cria uma tabela virtual que apresenta o nome e o id do funcionário responsável pela comunicação com a sua equipa. Além disso, a tabela inclui o nome e o id da equipe correspondente.

```

-- DROP VIEW ComunicaFuncionarioEquipa;
CREATE VIEW ComunicaFuncionarioEquipa AS
SELECT
    Comunica.Funcionarios_idFuncionarios,
    Funcionarios.Nome AS NomeFuncionario,
    Comunica.Equipas_idEquipas,
    Equipas.Nome AS NomeEquipa
FROM
    Comunica
INNER JOIN
    Funcionarios ON Comunica.Funcionarios_idFuncionarios = Funcionarios.idFuncionarios
INNER JOIN
    Equipas ON Comunica.Equipas_idEquipas = Equipas.idEquipas;

```

Figura 80: Vista que mostra o funcionário responsável pela comunicação com uma determinada equipa

A terceira vista cria uma tabela com todas as informações relacionadas a uma encomenda.


```

-- DROP VIEW InfoEncomenda;
CREATE VIEW InfoEncomenda AS
SELECT
    Encomenda.idEncomenda,
    Encomenda.Valor,
    Encomenda.idlembanca,
    Lembrancas.Nome AS NomeLembra,
    Encomenda.Quantidade,
    Encomenda.idfuncionario,
    Funcionarios.Nome AS NomeFuncionario,
    Encomenda.idfornecedor,
    Fornecedor.Nome AS NomeFornecedor,
    Fornecedor.contacto_telemovei AS FornecedorTel,
    Fornecedor.contacto_email AS FornecedorEmail
FROM
    Encomenda
INNER JOIN
    Lembrancas ON Encomenda.idlembanca = Lembrancas.idLembrancas
INNER JOIN
    Funcionarios ON Encomenda.idfuncionario = Funcionarios.idFuncionarios
INNER JOIN
    Fornecedor ON Encomenda.idfornecedor = Fornecedor.idFornecedor;

```

Figura 81: Vista que mostra toda a informação de uma encomenda

A quarta vista cria uma tabela virtual que contém todas as informações de um determinado piloto, desde o seu número de pista até aos pontos obtidos nas suas corridas em determinado evento.

```

-- DROP VIEW InfoPiloto;
CREATE VIEW InfoPiloto AS
SELECT DISTINCT
    Pilotos.numero_Piloto,
    Pilotos.Nome AS NomePiloto,
    Pilotos.data_nasc,
    Classificacao.Pontos,
    Corrida.tipo,
    Pilotos.contacto_telemovei,
    Pilotos.contacto_email,
    Pilotos.morada_rua,
    Pilotos.morada_localidade,
    Pilotos.morada_porta,
    Pilotos.morada_codigoPostal,
    Pilotos.Equipas_idEquipas,
    Equipas.Nome AS NomeEquipa,
    Evento.Nome AS NomeEvento
FROM
    Pilotos
INNER JOIN
    Classificacao ON Pilotos.numero_Piloto = Classificacao.numero_Piloto
INNER JOIN
    Corrida ON Classificacao.Corrida_tipo = Corrida.tipo
INNER JOIN
    Evento ON Corrida.idEvento = Evento.idEvento
INNER JOIN
    Equipas ON Pilotos.Equipas_idEquipas = Equipas.idEquipas;

```

Figura 82: Vista que mostra toda a informação de um piloto

A última vista criada, também cria uma tabela virtual, mas esta apenas contém a equipa vencedora e os seus pontos de determinado evento.

```

-- DROP VIEW EquipaVencedora;
CREATE VIEW EquipaVencedora AS
SELECT DISTINCT
    Evento.idEvento,
    Evento.Nome AS NomeEvento,
    Equipas.idEquipas,
    Equipas.Nome AS NomeEquipaVencedora,
    Equipas.Pontos
FROM
    Evento
INNER JOIN
    Corrida ON Evento.idEvento = Corrida.idEvento
INNER JOIN
    Classificacao ON Corrida.idEvento = Classificacao.idEvento
INNER JOIN
    Pilotos ON Classificacao.numero_Piloto = Pilotos.numero_Piloto
INNER JOIN
    Equipas ON Pilotos.Equipas_idEquipas = Equipas.idEquipas
WHERE
    Equipas.Pontos = (SELECT MAX(Pontos) FROM Equipas);

```

Figura 83: Vista que mostra a equipa vencedora de um determinado evento

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

Para melhor utilização e organização da base de dados é importantíssimo a existência de perfis de utilização. Para a SuperKarts Cup criamos sete perfis de utilizadores diferentes, atribuindo-lhes a suas respetivas permissões, de forma a garantir a segurança e integridade da base de dados.

O primeiro perfil criado, foi o perfil Gerente, este perfil tem total permissão para fazer qualquer tipo seleção, inserção ou alteração na base de dados.

```
-- DROP USER 'Gerente'@'%';
CREATE USER 'Gerente'@'%' IDENTIFIED BY 'Sr.Roberto';

GRANT ALL PRIVILEGES ON `SuperKartsCup`.* TO 'Gerente'@'%';
```

Figura 84: Criação de utilizadores "Gerente" e devidas permissões

O próximo perfil a ser criado foi o perfil Funcionario, este é responsável pelas vendas e encomendas do evento, podendo ainda ler e inserir os turnos de trabalho. Por fim os Funcionarios podem ler toda a base de dados do evento.

```
-- DROP USER 'Funcionario'@'%';
CREATE USER 'Funcionario'@'%' IDENTIFIED BY 'funcionario2024';

GRANT SELECT, INSERT, UPDATE ON `SuperKartsCup`.`Venda` TO 'Funcionario'@'%';
GRANT SELECT, INSERT, UPDATE ON `SuperKartsCup`.`Encomenda` TO 'Funcionario'@'%';
GRANT SELECT, INSERT ON `SuperKartsCup`.`Turno` TO 'Funcionario'@'%';
GRANT SELECT ON `SuperKartsCup`.* TO 'Funcionario'@'%';
```

Figura 85: Criação de utilizadores "Funcionario" e devidas permissões

O terceiro perfil a ser criado foi o perfil Arbitro, este perfil tem permissão para ver as tabelas, Corrida, Comissarios, Equipas, Pilotos e ainda permissão para ler, inserir ou alterar os dados da tabela Classificacao, pois, os Árbitros são os responsáveis pelo ajuizamento da prova e atribuição de pontos das equipas.

```
-- DROP USER 'Arbitro'@'%';
CREATE USER 'Arbitro'@'%' IDENTIFIED BY 'arbitro2024';

GRANT SELECT ON `SuperKartsCup`.`Arbitro` TO 'Arbitro'@'%';
GRANT SELECT, INSERT, UPDATE ON `SuperKartsCup`.`Classificacao` TO 'Arbitro'@'%';
GRANT SELECT ON `SuperKartsCup`.`Corrida` TO 'Arbitro'@'%';
GRANT SELECT ON `SuperKartsCup`.`Comissarios` TO 'Arbitro'@'%';
GRANT SELECT ON `SuperKartsCup`.`Equipas` TO 'Arbitro'@'%';
GRANT SELECT ON `SuperKartsCup`.`Pilotos` TO 'Arbitro'@'%';
GRANT EXECUTE ON PROCEDURE AtualizarPontosEquipas TO 'Arbitro'@'%';
```

Figura 86: Criação de utilizadores "Arbitro" e devidas permissões

O quarto perfil criado foi o perfil Comissario, este perfil controla tudo o que envolve a corrida.

```
-- DROP USER 'Comissario'@'%';
CREATE USER 'Comissario'@'%' IDENTIFIED BY 'comissario2024';

GRANT SELECT ON `SuperKartsCup`.`Comissarios` TO 'Comissario'@'%';
GRANT SELECT ON `SuperKartsCup`.`Classificacao` TO 'Comissario'@'%';
GRANT SELECT ON `SuperKartsCup`.`Arbitro` TO 'Comissario'@'%';
GRANT SELECT ON `SuperKartsCup`.`Corrida` TO 'Comissario'@'%';
GRANT SELECT ON `SuperKartsCup`.`Equipas` TO 'Comissario'@'%';
GRANT SELECT ON `SuperKartsCup`.`Pilotos` TO 'Comissario'@'%';
GRANT SELECT ON `SuperKartsCup`.`Funcionarios da Equipa` TO 'Comissario'@'%';
GRANT SELECT ON `SuperKartsCup`.`Material de Corrida` TO 'Comissario'@'%';
```

Figura 87: Criação de utilizadores "Comissario" e devidas permissões

O próximo perfil criado foi o perfil Chefe de Equipa, este perfil pode ler tudo o que é relacionado as corridas, mas apenas pode inserir ou alterar dados nas tabelas Pilotos e Funcionarios de Equipa.

```
-- DROP USER 'Chefe de Equipa'@'%';
CREATE USER 'Chefe de Equipa'@'%' IDENTIFIED BY 'chefe2024';

GRANT SELECT ON `SuperKartsCup`.`Classificacao` TO 'Chefe de Equipa'@'%';
GRANT SELECT ON `SuperKartsCup`.`Corrida` TO 'Chefe de Equipa'@'%';
GRANT SELECT ON `SuperKartsCup`.`Equipas` TO 'Chefe de Equipa'@'%';
GRANT SELECT, INSERT, UPDATE ON `SuperKartsCup`.`Pilotos` TO 'Chefe de Equipa'@'%';
GRANT SELECT, INSERT, UPDATE ON `SuperKartsCup`.`Funcionarios da Equipa` TO 'Chefe de Equipa'@'%';
GRANT SELECT ON `SuperKartsCup`.`Material de Corrida` TO 'Chefe de Equipa'@'%';
```

Figura 88: Criação de utilizadores "Chefe de Equipa" e devidas permissões

O sexto perfil foi criado para os pilotos do evento, este foi apelidado de Piloto, tendo permissões para ler todo que seja relacionado com ele e com a sua equipa.

```
-- DROP USER 'Piloto'@'%';
CREATE USER 'Piloto'@'%' IDENTIFIED BY 'piloto2024';

GRANT SELECT ON `SuperKartsCup`.`Pilotos` TO 'Piloto'@'%';
GRANT SELECT ON `SuperKartsCup`.`Classificacao` TO 'Piloto'@'%';
GRANT SELECT ON `SuperKartsCup`.`Corrida` TO 'Piloto'@'%';
GRANT SELECT ON `SuperKartsCup`.`Equipas` TO 'Piloto'@'%';
GRANT SELECT ON `SuperKartsCup`.`Funcionarios da Equipa` TO 'Piloto'@'%';
GRANT SELECT ON `SuperKartsCup`.`Material de Corrida` TO 'Piloto'@'%';
```

Figura 89: Criação de utilizadores "Piloto" e devidas permissões

O último perfil, foi criado para os funcionários das equipas, denominado de Funcionario de Equipa, este perfil tem permissões para ler todo o que relaciona a ele e a sua equipa, mas pode alterar os dados da tabela Material de Corrida.

```
-- DROP USER 'Funcionario de Equipa'@'%';
CREATE USER 'Funcionario de Equipa'@'%' IDENTIFIED BY 'funcionariodeequipa2024';

GRANT SELECT ON `SuperKartsCup`.`Pilotos` TO 'Funcionario de Equipa'@'%';
GRANT SELECT ON `SuperKartsCup`.`Classificacao` TO 'Funcionario de Equipa'@'%';
GRANT SELECT ON `SuperKartsCup`.`Corrida` TO 'Funcionario de Equipa'@'%';
GRANT SELECT ON `SuperKartsCup`.`Equipas` TO 'Funcionario de Equipa'@'%';
GRANT SELECT ON `SuperKartsCup`.`Funcionarios da Equipa` TO 'Funcionario de Equipa'@'%';
GRANT SELECT, UPDATE ON `SuperKartsCup`.`Material de Corrida` TO 'Funcionario de Equipa'@'%';
```

Figura 90: Criação de utilizadores "Funcionario de Equipa" e devidas permissões

5.7. Indexação do sistema de dados

A indexação é um mecanismo fundamental nos sistemas de gestão de dados, projetado para melhorar a eficiência das operações de consulta e recuperação de informações. Este processo cria estruturas adicionais, chamadas índices, que permitem localizar registos de forma rápida, sem a necessidade de percorrer toda a tabela.

No caso da SuperKarts Cup, foram criados cinco índices em tabelas estratégicas para facilitar e melhorar a eficiência da consulta da mesma.

```

CREATE INDEX idx_evento_data_inicio ON Evento (Data_Inicio);

CREATE INDEX idx_funcionarios_nome ON Funcionarios (Nome);

CREATE INDEX idx_pilotos_nome ON Pilotos (Nome);

CREATE INDEX idx_equipas_nome ON Equipas (Nome);

CREATE INDEX idx_encomenda_estado ON Encomenda (Estado);

```

Figura 91: Aplicação de indexação na SuperKarts Cup

5.8. Procedimentos implementados

Para a automatização e simplificação da base de dados, foram criados três procedimentos

O primeiro procedimento a ser implementado foi um gatilho. Este gatilho foi criado para quando uma encomenda fosse criada, o gatilho cria uma entrada na tabela notificação.

```

DELIMITER $$
-- DROP TRIGGER cria_encomenda;
CREATE TRIGGER cria_encomenda
AFTER INSERT ON Encomenda
FOR EACH ROW
BEGIN
    INSERT INTO Notificacao (descricao, idEncomenda)
    VALUES (
        CONCAT('Nova encomenda criada com ID: ', NEW.idEncomenda,
            ', Valor: ', NEW.Valor,
            ', Quantidade: ', NEW.Quantidade),
        NEW.idEncomenda
    );
END$$
DELIMITER ;

```

Figura 92: Gatilho responsável por criar uma notificação quando uma encomenda é criada

O segundo procedimento implementado foi também um gatilho. O gatilho agora criado também cria uma entrada na tabela Notificacao, mas desta vez é quando uma encomenda é atualizada, notificando os funcionários.

```

DELIMITER $$
-- DROP TRIGGER atualiza_encomenda;
CREATE TRIGGER atualiza_encomenda
AFTER UPDATE ON Encomenda
FOR EACH ROW
BEGIN
    INSERT INTO Notificacao (descricao, idEncomenda)
    VALUES (
        CONCAT('Encomenda com ID ', NEW.idEncomenda,
            ' foi alterada. Estado atual: ', NEW.Estado,
            ', Valor: ', NEW.Valor,
            ', Quantidade: ', NEW.Quantidade),
        NEW.idEncomenda
    );
END$$
DELIMITER ;

```

Figura 93: Gatilho responsável por criar uma notificação quando uma encomenda é atualizada

O último procedimento foi criado para atribuir/atualizar os pontos das equipas de forma automática. Este procedimento realiza a soma de todos os pontos adquiridos pelos pilotos de certa equipa e de forma automática, atribui esses pontos a respetiva equipa.

```

DELIMITER $$
-- DROP PROCEDURE AtualizarPontosEquipas();
CREATE PROCEDURE AtualizarPontosEquipas()
BEGIN
    UPDATE Equipas
    SET Equipas.Pontos = (
        SELECT SUM(Classificacao.Pontos)
        FROM Classificacao
        JOIN Pilotos ON Classificacao.numero_Piloto = Pilotos.numero_Piloto
        WHERE Pilotos.Equipas_idEquipas = Equipas.idEquipas
    );
END$$
DELIMITER ;

```

Figura 94: Procedimento responsável por atribuir/atualizar os pontos das equipas

5.9. Plano de segurança e recuperação de dados

Para garantir a segurança dos dados, armazenados na base de dados da SuperKarts Cup, é composto um plano robusto de segurança e recuperação em caso de falhas ou desastres.

Em primeiro lugar realizamos quatro práticas de segurança com o objetivo de proteger os dados contra acessos não autorizados e outras ameaças. A primeira prática que devemos realizar é o **Controlo dos acessos**, esta prática pode ser posta em prática de três maneiras:

- **Princípio do menor privilégio:** Em que se garante que cada utilizador ou sistema tenha apenas as permissões mínimas necessárias para realizar as suas tarefas. Isto reduz os riscos de acessos indevidos ou erros acidentais.
- **Autenticação forte:** Utilizar palavras-passe complexas (mistura de letras, números e caracteres especiais) e implementar autenticação de dois fatores (2FA) sempre que possível. O 2FA adiciona uma camada extra de segurança.
- **Gestão de permissões:** Gerir e rever regularmente as permissões atribuídas a utilizadores, grupos e aplicações para garantir que estão alinhadas às necessidades atuais.

A segunda prática que devemos adotar para termos uma base de dados impenetrável é a **Criptografia**. A Criptografia deve ser aplicada tanto nos dados em repouso, como nos dados em trânsito:

- **Dados em repouso:** Para proteger dados armazenados, utilizar TDE (Transparent Data Encryption), que encripta automaticamente os dados no disco sem necessidade de alterações nas aplicações.
- **Dados em Trânsito:** Utilizar protocolos como TLS (Transport Layer Security) ou SSL (Secure Sockets Layer) para proteger a comunicação entre sistemas e impedir intercepções ou ataques intermediários.

A terceira prática de segurança é a **Monitorização e Auditoria**, esta prática consiste em:

- **Registos da auditoria:** Ativar registos que registem atividades importantes na base de dados, como acessos, alterações de permissões e consultas sensíveis.
- **Monitorização contínua:** Implementar ferramentas para observar atividades em tempo real e identificar comportamentos anómalos ou suspeitos, como acessos fora de horário ou de localizações incomuns.

A última prática de segurança que devemos adotar é **Atualizações**, ou seja:

- **Correções regulares:** Aplicar atualizações e *patches* de segurança no sistema de gestão de base de dados e no sistema operativo. Isto protege contra vulnerabilidades conhecidas.

Depois de garantir que as quatro práticas de segurança estão a ser cumpridas a rigor, passamos para a realização do plano de cópias de segurança, para garantirmos a recuperação dos dados em caso de falha, perda ou até mesmo desastre. O plano consiste em três passos, em que o primeiro dos mesmos é realizar três **Tipos de cópias de segurança**:

- **Cópia completa:** Faz uma cópia de todos os dados, ideal para criar pontos de referência sólidos. Geralmente realizada em períodos específicos (diariamente, semanalmente).
- **Cópia diferencial:** Guarda as alterações feitas desde a última cópia completa, reduzindo o tempo de execução e o espaço necessário.
- **Cópia de logs de transação:** Captura os registros das transações, permitindo recuperar os dados até um ponto específico no tempo.

O segundo passo apelidamos de **Estratégia de armazenamento**, este passo consiste em:

- **Redundância:** Manter as cópias de segurança em diferentes locais, um no local, para uma recuperação rápida, fora do local, ou seja, em locais externos, para uma proteção contra desastres físicos e na nuvem, o que oferece escalabilidade e acessibilidade.
- **Frequências de cópias:** Realizar cópias completas diárias e cópias dos logs de transação em intervalos curtos (15 a 30 minutos) para minimizar a perda dos dados.

O último passo é a **Validação**. Este passo consiste em:

- **Testes regulares de restauração:** Realizar testes periódicos para garantir que as cópias estão íntegras e podem ser restauradas com sucesso.

Em terceiro lugar, para termos maneira de restabelecer o sistema e minimizar o impacto de falhas ou desastres, elaboramos um plano de recuperação. Este plano consiste na realização de três etapas, **Identificação do problema**, **Etapas de recuperação** e **Alta disponibilidade**, ou seja:

- **Identificação do problema:** Utilizar registros, auditorias e ferramentas de monitorização para diagnosticar a causa da falha, como corrupção de dados, falha de hardware ou ataque informático.
- **Etapas de recuperação:** Em primeiro lugar, devemos selecionar a cópia mais recente e íntegra, de seguida, devemos, antes de reintegrar o sistema, restaurar os dados num ambiente isolado para verificar a integridade e por fim, após o resultado dos testes serem bem-sucedidos, reintroduzir os dados no sistema.
- **Alta disponibilidade:** Replicação dos dados, ou seja, utilizar replicação síncrona ou assíncrona para manter cópias atualizadas dos dados em servidores secundários e *clusters* de alta disponibilidade, ou seja, configurar os *clusters* com *failover* automático para garantir que o sistema continue a funcionar mesmo que o servidor principal falhe.

Chegando ao último passo do plano de segurança e recuperação de dados, devemos garantir que o sistema continue operacional mesmo durante ou após um desastre. A este passo chamamos continuidade operacional. A continuidade operacional consiste em duas coisas:

- **Objetivos de recuperação:** Existem dois objetivos de recuperação, um deles é o RPO (Recovery Point Objective) que determina o tempo máximo de dados que pode ser perdido, guiando a frequência das cópias de segurança e o RTO (Recovery Time Objective) que define o tempo máximo aceitável para que o sistema seja restaurado após uma falha.

- **Testes de PRD (Plano de Recuperação de Desastre):** Realizar simulações regulares de desastres para testar e validar a eficiência do plano de recuperação. Isto garante que a equipa esteja preparada para lidar com emergências.

Estas práticas ajudam a proteger os dados, garantir a recuperação em situações de falha e minimizar interrupções nas operações críticas.

Lista de Siglas e Acrónimos

1. **SQL** – *Structured Query Language*
2. **TDE** - *Transparent Data Encryption*
3. **TLS** - *Transport Layer Security*
4. **SSL** - *Secure Sockets Layer*
5. **RPO** - *Recovery Point Objective*
6. **RTO** - *Recovery Time Objective*
7. **PRD** - Plano de Recuperação de Desastre