A blue and black logo

Description automatically generated

|  |
| --- |
| Bases de Dados 2023/2024  Licenciatura em Engª. Informática |
| Relatório Técnico |

Turma:

Horário de Laboratório: 10:30 Sexta Feira

Docente: Luís Damas

Grupo:

Nº201902549, Thiers de Mesquita  
Nº202300244, João Jorge

Índice

[1. Sumário 3](#_Toc168993284)

[2. Definição do domínio do problema 3](#_Toc168993285)

[3. Consultas 3](#_Toc168993286)

[4. Modelo Entidade Relação 3](#_Toc168993287)

[4.1 Alterações/Melhorias ao Relatório da 1ª Fase 3](#_Toc168993288)

[4.2 Levantamento das entidades, atributos e relacionamento 3](#_Toc168993289)

[4.3 Diagrama do Modelo Entidade Relação 4](#_Toc168993290)

[5. Modelo Relacional 4](#_Toc168993291)

[6. Diagrama do Modelo Relacional 4](#_Toc168993292)

[7. Consultas à Base de Dados 4](#_Toc168993293)

[7.1 Consultas de resposta ao enunciado 4](#_Toc168993294)

[7.2 Outras Consultas 4](#_Toc168993295)

[*7.3* *Views* 5](#_Toc168993296)

[8. Programação 5](#_Toc168993297)

[*8.1* *Registo de resultados* 5](#_Toc168993298)

[*8.2* *Outros SP* 5](#_Toc168993299)

[*8.3* *Functions* 5](#_Toc168993300)

[*8.4* *Triggers* 6](#_Toc168993301)

[*8.4.1* *Monitorização de falhas* 6](#_Toc168993302)

[*8.4.2* *Outros triggers* 6](#_Toc168993303)

[9. Conclusões 6](#_Toc168993304)

# Sumário

Este projeto visa a criação e gestão de uma base de dados para um sistema de gerenciamento de eventos esportivos. O sistema permite o registro de atletas, equipes, eventos, estádios, patrocinadores, organizadores e resultados de provas. As funcionalidades incluem a inserção, atualização e consulta dessas entidades, bem como a criação de views, procedures, funções e triggers para garantir a integridade dos dados e facilitar operações comuns.

# Definição do domínio do problema

O domínio escolhido é a gestão de eventos de voleibol. O sistema deve permitir o gerenciamento de informações sobre atletas, equipes, eventos, provas, estádios, patrocinadores e organizadores. Os requisitos funcionais incluem o registro de resultados, a associação de atletas a equipes e a vinculação de provas a eventos.

|  |  |
| --- | --- |
| **ID** | **Descrição** |
| R01 | O sistema deve permitir que uma equipe tenha pelo menos um atleta. |
| R02 | |  | | --- | | O sistema deve garantir que as tabelas tenham certos campos como NOT NULL. |  |  | | --- | |  | |
| R03 | O sistema deve permitir que cada evento tenha uma data de início e uma data de fim.   |  | | --- | |  | |
| R04 | |  | | --- | | O sistema deve garantir que cada prova seja associada a um estádio. |  |  | | --- | |  | |
| R05 | |  | | --- | | O sistema deve garantir que cada atleta esteja associado a uma ficha técnica. |  |  | | --- | |  | |
| RM01 | |  | | --- | | O sistema deve calcular a idade do atleta com base na data atual e na data de nascimento do atleta. |  |  | | --- | |  | |
| RM02 | |  | | --- | | O sistema deve calcular o desempenho global do treinador com base nos resultados das equipes que ele treinou. |  |  | | --- | |  | |
| RM03 | |  | | --- | | O sistema deve calcular a duração total de um evento com base nas datas de início e fim. |  |  | | --- | |  | |
| RM04 | |  | | --- | | O sistema deve calcular a capacidade total disponível em um evento com base na capacidade dos estádios onde as provas são realizadas. |  |  | | --- | |  | |
| RM05 | O sistema deve calcular o número total de provas em um evento. |

# Consultas

As consultas principais foram definidas para permitir a extração de dados importantes para o gerenciamento dos eventos. Isso inclui listagens de atletas, equipes, eventos e resultados.

# Modelo Entidade Relação

## Alterações/Melhorias ao Relatório da 1ª Fase

**Renomeação de Entidades:**

"Jogador" foi renomeado para "Atleta.", "Time" foi renomeado para "Equipa.", "Partida" foi renomeado para "Prova.", "Campeonato" foi renomeado para "Evento.", "Entidade Organizadora" foi renomeada para "Organizadora."

**Atributos Removidos:**

Atributos "E-Mail" e "Telemovel" foram removidos da entidade "Pessoa.", Atributos "Altura" e "Peso" foram removidos da entidade "Jogador.", Atributos "Fname" e "Lname" foram removidos da entidade "FichaTecnica.", Atributo "NewAttribute" foi removido da entidade "Resultado."

**Atributos Adicionados**:

Atributo "IdTreinador" foi adicionado à entidade "Equipa.", Atributo "IdEstadio" foi adicionado à entidade "Prova."

**Entidades Removidas:** A entidade "Pais" foi removida.

**Alterações nos Relacionamentos:**

Relacionamento "Inscreve" foi alterado para incluir "Atleta" e "Treinador" inscrevendo em "Equipa."

Relacionamento "Inclui" entre "Evento" e "Prova" foi removido.

Relacionamento "Participa" entre "Equipa" e "Evento" foi adicionado.

Relacionamento "Tem" entre "Prova" e "Resultado" foi mantido.

Relacionamento "Realiza-se" entre "Prova" e "Estadio" foi mantido.

Relacionamento "Patrocina" entre "Patrocinador" e "Equipa" foi mantido.

Relacionamento "Organiza" entre "Organizadora" e "Evento" foi mantido.

## Levantamento das entidades, atributos e relacionamento

**- Pessoa**: (IdPessoa PK), Nome, DataNascimento, Nacionalidade, Morada, Cidade, Bairro, Posição

**- Atleta**: (IdAtleta PK), IdPessoa (FK), Capitão

**- Treinador**: (IdTreinador PK), IdPessoa (FK), AnosExperiencia, DesempenhoGlobal)

**- FichaTecnica**: (IdAtleta PK, FK), IdTreinador (FK)

**- Equipa**: (IdEquipa PK), NomeEquipa, IdTreinador (FK)

**- Evento**: (IdEvento PK), NomeEvento, DataInicio, DataFim, TotalProvas

**- Prova**: (IdProva PK), NomeProva, Data, Hora, Jornada, Duracao, LocalProva, IdEstadio (FK)

**- Resultado**: (IdResultado PK), Ranking

**- Estadio**: (IdEstadio PK), Nome, Capacidade, Localizacao

**- Organizadora**: (IdOrganizadora PK), Nome

**- Patrocinador**: (IdPatrocinador PK), Nome

## Diagrama do Modelo Entidade Relação

Diagrama, Esquemático

Descrição gerada automaticamente

# Modelo Relacional

## Especificação do Modelo Relacional

**Atleta(IdAtleta, nome\_atleta, data\_nasc, Nacionalidade, Cidade, Bairro, Posicao)**

- PK(IdAtleta)

**Equipa(IdEquipa, nome\_equipa, desempenho\_global, IdTreinador)**

- PK(IdEquipa)

- FK(IdTreinador) -> Treinador(IdTreinador)

**Evento(IdEvento, nome\_evento, Local, data\_inicio, data\_fim, total\_provas)**

- PK(IdEvento)

**Prova(IdProva, nome\_prova, Local\_prova, data, hora, duracao, rodadas, IdEstadio)**

- PK(IdProva)

- FK(IdEstadio) -> Estadio(IdEstadio)

**Resultado(IdResultado, classificacao, ranking)**

- PK(IdResultado)

**Prova\_Evento(IdProva, IdEvento)**

- PK(IdProva, IdEvento)

- FK(IdProva) -> Prova(IdProva)

- FK(IdEvento) -> Evento(IdEvento)

**ProvaResultado(IdProva, IdResultado)**

- PK(IdProva, IdResultado)

- FK(IdProva) -> Prova(IdProva)

- FK(IdResultado) -> Resultado(IdResultado)

**AtletaEquipa(IdAtleta, IdEquipa)**

- PK(IdAtleta, IdEquipa)

- FK(IdAtleta) -> Atleta(IdAtleta)

- FK(IdEquipa) -> Equipa(IdEquipa)

**FichaTecnica(IdFicha, curriculo, historico)**

- PK(IdFicha)

**AtletaFicha(IdAtleta, IdFicha)**

- PK(IdAtleta, IdFicha)

- FK(IdAtleta) -> Atleta(IdAtleta)

- FK(IdFicha) -> FichaTecnica(IdFicha)

**Genero(IdGenero, descricao)**

- PK(IdGenero)

**AtletaGenero(IdAtleta, IdGenero)**

- PK(IdAtleta, IdGenero)

- FK(IdAtleta) -> Atleta(IdAtleta)

- FK(IdGenero) -> Genero(IdGenero)

**Estadio(IdEstadio, Nome, Capacidade, Localizacao)**

- PK(IdEstadio)

**Treinador(IdTreinador, Nome, DataNasc, Cidade, Bairro, AnosExperiencia)**

- PK(IdTreinador)

**Patrocinador(IdPatrocinador, Nome)**

- PK(IdPatrocinador)

**Organizadora(IdOrganizadora, Nome)**

- PK(IdOrganizadora)

**ProvaOrganizadora(IdProva, IdOrganizadora)**

- PK(IdProva, IdOrganizadora)

- FK(IdProva) -> Prova(IdProva)

- FK(IdOrganizadora) -> Organizadora(IdOrganizadora)

**ProvaPatrocinador(IdProva, IdPatrocinador)**

- PK(IdProva, IdPatrocinador)

- FK(IdProva) -> Prova(IdProva)

- FK(IdPatrocinador) -> Patrocinador(IdPatrocinador)

## Diagrama do Modelo Relacional

*Imagem do diagrama do modelo relacional, gerado no MySQL Workbench.*

# Consultas à Base de Dados

## Consultas de resposta ao enunciado

|  |  |  |
| --- | --- | --- |
| **Comando SQL (SELECT)** | **Descrição** | **Critério** |
| SELECT A.nome\_atleta, E.nome\_equipa FROM Atleta A JOIN AtletaEquipa AE ON A.IdAtleta = AE.IdAtleta JOIN Equipa E ON AE.IdEquipa = E.IdEquipa; | Lista os atletas e suas respectivas equipes. | ***1.1*** |
| *SELECT nome\_atleta AS 'Nome do Atleta', data\_nasc AS 'Data de Nascimento', Nacionalidade AS 'Nacionalidade'*  *FROM Atleta*  *WHERE Nacionalidade = 'Brasil';* | *Lista de atletas brasileiros* | ***1.2*** |
| *SELECT E.nome\_equipa AS 'Nome da Equipa', A.nome\_atleta AS 'Nome do Atleta'*  *FROM Equipa E*  *JOIN AtletaEquipa AE ON E.IdEquipa = AE.IdEquipa*  *JOIN Atleta A ON AE.IdAtleta = A.IdAtleta*  *WHERE E.desempenho\_global > 85;* | *Equipas com desempenho global acima de 85* | ***2.1*** |
|  |  |  |

## Outras Consultas

*Apresentar o comando SQL da consulta e descrever sucintamente o seu objetivo.*

|  |  |
| --- | --- |
| **Comando SQL (SELECT)** | **Descrição** |
| *Select …*  *From participante*  *….* | *Lista de participantes com ….* |
|  |  |

## *Views*

*Apresentar as views implementadas e descrever sucintamente o seu objetivo.*

|  |  |
| --- | --- |
| **Nome View** | **Descrição** |
| *V\_Participantes* | *Lista os participantes que ….* |
|  |  |

# Programação

*Apresentar o código desenvolvido em Stored Procedures (SP), Functions e triggers*

## *Registo de resultados*

*Apresentar os SP, que implementam a lógica de negócio correspondente ao registo de resultados. Fazer a correspondência com os SP identificados no enunciado.*

|  |  |
| --- | --- |
| **SP implementado** | **SP enunciado** |
| *sp\_criar\_jogo(in data date,….)*  *….* | *1* |
|  |  |

## *Outros SP*

*Apresentar os SP implementados e descrever sucintamente o seu objetivo.*

|  |  |
| --- | --- |
| **SP** | **Descrição** |
| *sp\_ipma(in local varchar(50),*  *in dia tinyint)*  *….* | *Apresenta a previsão meteorológica para um determinado local e dia* |
|  |  |

## *Functions*

*Apresentar as Functions implementadas e descrever sucintamente o seu objetivo.*

|  |  |
| --- | --- |
| **SP** | **Descrição** |
| *fn\_ipma\_temp(in local varchar(50),*  *in dia tinyint, in hora tinyint) returns tinyint*  *….* | *Devolve a temperatura prevista para um determinado local, dia e hora* |
|  |  |

## *Triggers*

*Apresentar os triigers implementados e descrever sucintamente a sua lógica.*

## *Monitorização de falhas*

*Apresentar e descrever os triggers implementados para responder ao tópico Monitorização de falhas descrito no enunciado.*

|  |  |
| --- | --- |
| **Trigger** | **Trigger enunciado** |
| *Log\_resultado before update on resultados ….*  *….* | *Result\_change* |
|  |  |

## *Outros triggers*

*Apresentar e descrever outros triggers implementados e descrever o seu objetivo.*

|  |  |
| --- | --- |
| **Trigger** | **Descrição** |
| *CH\_resultado before update on resultados ….* | *Verifica se a pontuação de um resultado…* |
|  |  |

# Conclusões

O desenvolvimento deste projeto permitiu implementar uma base de dados funcional para a gestão de eventos esportivos, focando no voleibol. As principais funcionalidades incluem o registro de entidades e resultados, consultas detalhadas e o uso de views, procedures e functions para facilitar a manipulação dos dados. Desafios encontrados incluíram garantir a integridade referencial e lidar com constraints, mas foram superados com a implementação de procedures robustas.