

**Base de Dados  
2023/2024**

**Licenciatura em ENG<sup>a</sup>. Informática**

**Relatório Técnico (Época Exame)**

Turma: L\_EI-02

Horário de Laboratório: 8:30-10:30

Docente: Luís Damas

Grupo:

Nº 202300364, Eduardo Vemba

## Conteúdo

1. Sumário.....	3
2. Definição do Domínio do problema .....	3
3. Consultas .....	3
<b>4. Modelo Entidade Relação .....</b>	<b>4</b>
4.1 Levantamento das Entidades, atributos e relacionamento .....	4
Modelo Relacional .....	7
Justificativa das Escolhas .....	7
4.2 Diagrama do Modelo Entidade Relação.....	7
5. Modelo Relacional.....	8
5.2 Diagrama do Modelo Relacional .....	10
6. Consultas à Base de Dados.....	10
6.2 Outras Consultas .....	11
6.3 Views .....	11
7. Programação .....	12
7.1 stored procedures .....	12
7.2 Functions .....	12
7.3 Triggers.....	13
7.3.1 Monitorização de falhas.....	13
7.3.2 Outros triggers .....	14
8. Conclusões .....	15

## 1. Sumário

Este projeto visa desenvolver uma base de dados para Corridas desportivas. O sistema permitirá Criação de uma prova de corridas entre Homens ou Mulheres .

Este projeto terá como Objetivo o gerenciamento Corridas.

## 2. Definição do Domínio do problema

Área selecionada: O projeto foca no desenvolvimento de um sistema de gerenciamento de corridas, visando atender às necessidades de organizadores de eventos esportivos, atletas e espectadores.

Descrição detalhada da lógica do sistema: O sistema de gerenciamento de corridas proposto tem como objetivo principal facilitar a organização, execução e acompanhamento de eventos de corrida. Ele abrangerá as seguintes funcionalidades principais:

ID	Descrição
RO1	O sistema devera permitir o registo de eventos de corrida
RO2	O sistema deverá permitir a atualização de informações de corredores
RO3	O sistema deverá permitir a consulta de dados sobre eventos de corrida, resultados individuais, rankings
RM01	(Melhoria Proposta) O sistema deverá permitir o calculo de posição de acordo com a posição na corrida
RMO2	(Melhoria Proposta) Melhoria nos Triggers que por vezes eles não aparecem as mensagem pressupostas

## 3. Consultas

**Select \* from pais :** Esta consulta exhibe todos os países que foram inseridos na tabela **pais**, permitindo verificar se todos os países foram corretamente adicionados ao banco de dados.

**Select \* from estadio** : Esta consulta exibe todos os estádios inseridos na tabela **estadio**, permitindo confirmar que todos os estádios e suas respectivas informações foram corretamente adicionados.

**Select \* from atleta** : Esta consulta exibe todos os jogadores inseridos na tabela **atleta**, permitindo confirmar que todos os jogadores e suas respectivas informações foram corretamente adicionados.

**Select \* from jogo** : Esta consulta exibe todos os jogos inseridos na tabela **jogo**, permitindo verificar se todos os jogos foram corretamente registrados no banco de dados.

**Select \* from participacao\_corrida** : Esta consulta exibe todas as participações nas corridas registradas na tabela **participacao\_corrida**, permitindo verificar se todas as participações foram corretamente registradas.

## 4. Modelo Entidade Relação

### 4.1 Levantamento das Entidades, atributos e relacionamento

#### 1.1. Entidade: `pais`

- **Descrição:** Representa os países onde os estádios estão localizados.
- **Atributos:**
  - `codigo` (PK): Código do país (chave primária, `VARCHAR(3)`).
  - `nome`: Nome do país (`VARCHAR(50)`).

#### 1.2. Entidade: `estadio`

- **Descrição:** Representa os estádios onde as provas (jogos) ocorrem.
- **Atributos:**
  - `id` (PK): Identificador único do estádio (chave primária, `INT`).
  - `nome`: Nome do estádio (`VARCHAR(100)`).
  - `capacidade`: Capacidade de público do estádio (`INT`).
  - `localizacao`: Localização do estádio (`VARCHAR(100)`).
  - `country_code` (FK): Código do país onde o estádio está localizado (chave estrangeira, `VARCHAR(3)`).
- **Restrições:**
  - `country_code` deve existir na tabela `pais`.

### 1.3. Entidade: atleta

- **Descrição:** Representa os jogadores (participantes das provas).
- **Atributos:**
  - `id` (PK): Identificador único do jogador (chave primária, `INT`).
  - `first_name`: Primeiro nome do jogador (`VARCHAR(50)`).
  - `last_name`: Último nome do jogador (`VARCHAR(50)`).
  - `data_nascimento`: Data de nascimento do jogador (`DATE`).
  - `sexo`: Sexo do jogador (`CHAR(1)`), deve ser 'M' ou 'F'.
  - `country_code` (FK): Código do país do jogador (chave estrangeira, `VARCHAR(3)`).
- **Restrições:**
  - `sexo` deve ser 'M' ou 'F'.
  - `country_code` deve existir na tabela `pais`.

### 1.4. Entidade: jogo

- **Descrição:** Representa as provas (jogos) realizadas.
- **Atributos:**
  - `race_id` (PK): Identificador único da prova (chave primária, `INT`).
  - `data`: Data da prova (`DATE`).
  - `time`: Hora da prova (`TIME`).
  - `estadio_id` (FK): Identificador do estádio onde a prova será realizada (chave estrangeira, `INT`).
- **Restrições:**
  - `estadio_id` deve existir na tabela `estadio`.

### 1.5. Entidade: participacao\_corrida

- **Descrição:** Representa a participação dos jogadores nas provas.
- **Atributos:**
  - `race_id` (FK): Identificador da prova (chave estrangeira, `INT`).
  - `id` (FK): Identificador do jogador (chave estrangeira, `INT`).
  - `data`: Data da participação (`DATE`).
  - `pontos`: Pontuação do jogador na prova (`FLOAT`).
  - `posicao`: Posição do jogador na prova (`INT`).
- **Chave Primária Composta:**
  - `race_id + id`
- **Restrições:**
  - `race_id` deve existir na tabela `jogo`.
  - `id` deve existir na tabela `atleta`.
  - `pontos` deve estar entre 0 e 10.

## 2. Relacionamentos

### 2.1. Relacionamento: `pais - estadio`

- **Descrição:** Um país pode ter vários estádios, mas cada estádio pertence a um único país.
- **Participação:** Total no lado do estádio.
- **Cardinalidade:** 1

(um país tem muitos estádios).

### 2.2. Relacionamento: `pais - atleta`

- **Descrição:** Um país pode ter vários jogadores, mas cada jogador pertence a um único país.
- **Participação:** Total no lado do jogador.
- **Cardinalidade:** 1

(um país tem muitos jogadores).

### 2.3. Relacionamento: `estadio - jogo`

- **Descrição:** Um estádio pode ter vários jogos, mas cada jogo ocorre em um único estádio.
- **Participação:** Total no lado do jogo.
- **Cardinalidade:** 1

(um estádio tem muitos jogos).

### 2.4. Relacionamento: `jogo - participacao_corrida`

- **Descrição:** Um jogo pode ter várias participações de jogadores, mas cada participação refere-se a um único jogo.
- **Participação:** Total no lado da participação.
- **Cardinalidade:** 1

(um jogo tem muitas participações).

### 2.5. Relacionamento: `player - participacao_corrida`

- **Descrição:** Um jogador pode participar de vários jogos, mas cada participação refere-se a um único jogador.
- **Participação:** Total no lado da participação.
- **Cardinalidade:** 1

(um jogador tem muitas participações).

## Modelo Relacional

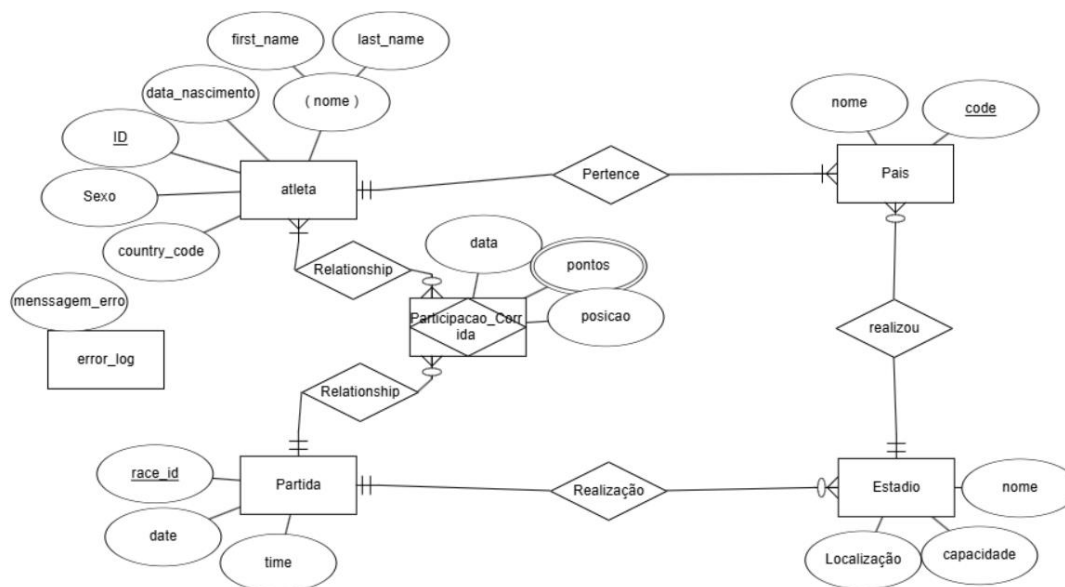
O modelo relacional resultante das entidades e relacionamentos é descrito a seguir:

1. **Pais** (codigo, nome)
2. **Estadio** (id, nome, capacidade, localizacao, country\_code)
3. **Atleta** (id, first\_name, last\_name, data\_nascimento, sexo, country\_code)
4. **Jogo** (race\_id, data, time, estadio\_id)
5. **Participacao\_corrida** (race\_id, id, data, pontos, posicao)

## Justificativa das Escolhas

- **Chaves Primárias (PK):** Seleccionadas para garantir a unicidade de cada registro nas tabelas.
- **Chaves Estrangeiras (FK):** Utilizadas para manter a integridade referencial entre as tabelas relacionadas.
- **Atributos:** Escolhidos para capturar as informações essenciais sobre cada entidade.
- **Restrições de Integridade:** Implementadas para garantir que os dados inseridos atendam aos requisitos do sistema, como a restrição de valores de sexo e a faixa de valores de pontos.

## 4.2 Diagrama do Modelo Entidade Relação



## 5. Modelo Relacional

### 5.1 Especificação do Modelo Relacional

Tabela: `pais`

- **Descrição:** Armazena informações sobre os países.
- **Atributos:**
  - `codigo` (VARCHAR(3)): Código do país (chave primária).
  - `nome` (VARCHAR(50)): Nome do país.
- **Chaves:**
  - **Primária:** `codigo`
  - **Candidatas:** Nenhuma
  - **Estrangeiras:** Nenhuma

Tabela: `estadio`

- **Descrição:** Armazena informações sobre os estádios.
- **Atributos:**
  - `id` (INT): Identificador único do estádio (chave primária).
  - `nome` (VARCHAR(100)): Nome do estádio.
  - `capacidade` (INT): Capacidade do estádio.
  - `localizacao` (VARCHAR(100)): Localização do estádio.
  - `country_code` (VARCHAR(3)): Código do país onde o estádio está localizado (chave estrangeira).
- **Chaves:**
  - **Primária:** `id`
  - **Candidatas:** Nenhuma
  - **Estrangeiras:** `country_code` referencia `pais (codigo)`

Tabela: `atleta`

- **Descrição:** Armazena informações sobre os jogadores.
- **Atributos:**
  - `id` (INT): Identificador único do jogador (chave primária).
  - `first_name` (VARCHAR(50)): Primeiro nome do jogador.
  - `last_name` (VARCHAR(50)): Último nome do jogador.
  - `data_nascimento` (DATE): Data de nascimento do jogador.
  - `sexo` (CHAR(1)): Sexo do jogador, deve ser 'M' ou 'F'.
  - `country_code` (VARCHAR(3)): Código do país do jogador (chave estrangeira).
- **Chaves:**
  - **Primária:** `id`
  - **Candidatas:** Nenhuma
  - **Estrangeiras:** `country_code` referencia `pais (codigo)`



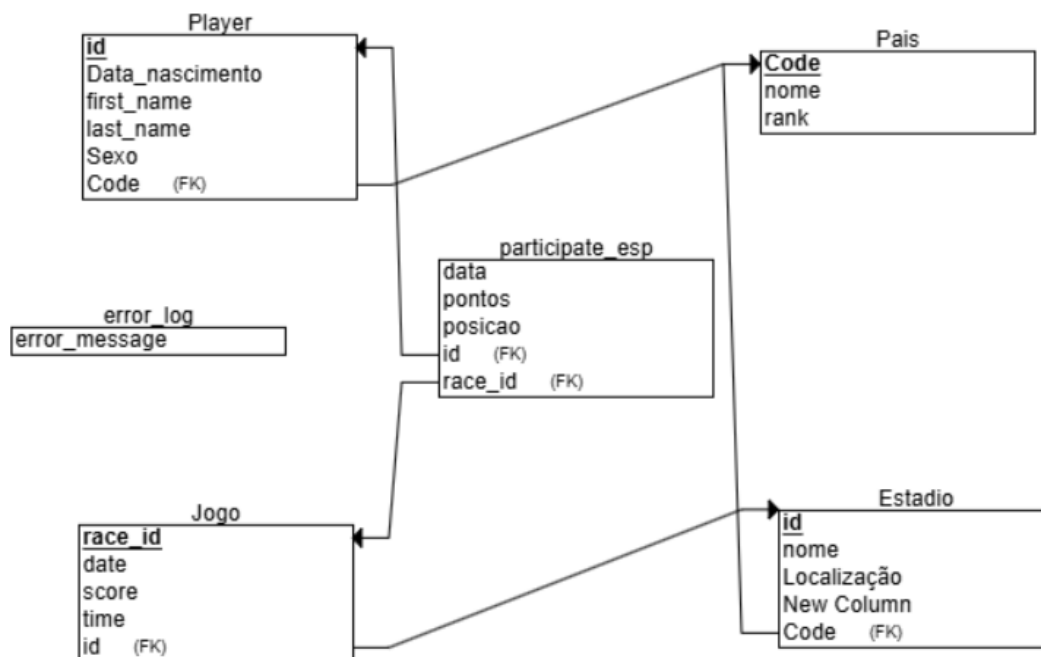
Tabela: `jogo`

- **Descrição:** Armazena informações sobre os jogos.
- **Atributos:**
  - `race_id` (INT): Identificador único do jogo (chave primária).
  - `data` (DATE): Data do jogo.
  - `time` (TIME): Hora do jogo.
  - `estadio_id` (INT): Identificador do estádio onde o jogo será realizado (chave estrangeira).
- **Chaves:**
  - **Primária:** `race_id`
  - **Candidatas:** Nenhuma
  - **Estrangeiras:** `estadio_id` referencia `estadio(id)`

Tabela: `participacao_corrida`

- **Descrição:** Armazena informações sobre a participação dos jogadores nas corridas.
- **Atributos:**
  - `race_id` (INT): Identificador do jogo (chave estrangeira, parte da chave primária composta).
  - `id` (INT): Identificador do jogador (chave estrangeira, parte da chave primária composta).
  - `data` (DATE): Data da participação.
  - `pontos` (FLOAT): Pontuação do jogador na corrida.
  - `posicao` (INT): Posição do jogador na corrida.
- **Chaves:**
  - **Primária:** (`race_id`, `id`)
  - **Candidatas:** Nenhuma
  - **Estrangeiras:**
    - `race_id` referencia `jogo(race_id)`
    - `id` referencia `atleta(id)`

## 5.2 Diagrama do Modelo Relacional



## 6. Consultas à Base de Dados

### 6.1 Consultas de resposta ao enunciado

Comando SQL (SELECT)	Descrição	
Select * from jogo	Lista de jogos realizados Poir sexo	
Select  atleta.first_name, atleta.last_name, pais.nome from pais join atleta on atleta.country_code = pais.codigo	Lista o nome dos atletas e os seus determinados paises	

## 6.2 Outras Consultas

## 6.3 Views

View	Descrição
rank_jogador	Declara um determinado rank de acordo ao avg de pontos nas corridas

create view rank\_jogador as

```
select p.id,p.first_name,p.last_name,avg(pc.pontos) as avg_pontos,  
case  
when avg(pontos) >= 8 then 'Rank Mundial'  
when avg(pontos) >= 6 then 'Rank Nacional'  
when avg(pontos) >= 4 then 'Rank Capital'  
else 'Rank Regional'  
end as TOP  
from participacao_corrida pc  
join atleta p  
on pc.id = p.id  
group by p.id, p.first_name, p.last_name;
```

## 7. Programação

### 7.1 stored procedures

SP	Descrição
Sp_criar_prova	Criação de provas de corrida de acordo com o sexo

DELIMITER //

```
CREATE PROCEDURE sp_criar_prova( IN p_data DATE, IN p_time TIME, IN
p_estadio_id INT, IN p_sexo VARCHAR(1) )
```

```
BEGIN DECLARE v_race_id INT;
```

```
IF p_sexo NOT IN ('M','F') THEN
```

```
SIGNAL SQLSTATE '45000'
```

```
SET MESSAGE_TEXT = 'Sexo inválido, use M de masculino ou F de feminino.';
```

```
END IF;
```

```
INSERT INTO jogo(data, time, estadio_id)
```

```
VALUES (p_data, p_time, p_estadio_id); SET v_race_id = LAST_INSERT_ID();
```

```
CREATE TEMPORARY TABLE temp_players AS SELECT id FROM player
```

```
WHERE sexo = p_sexo;
```

```
INSERT INTO participacao_corrida(id, race_id, data) SELECT id, v_race_id, p_data
```

```
FROM temp_players; DROP TEMPORARY TABLE IF EXISTS temp_players;
```

```
END //
```

```
DELIMITER ;
```

### 7.2 Functions

Function	Descrição
Calcular_idade	Calcula a idade do jogador de acordo a data de nascimento

```

DELIMITER //

create function calcular_idade(data_nascimento date)

returns int

deterministic

begin

    declare idade int;

    set idade = timestampdiff(year,data_nascimento,curdate());

    return idade;

end //

DELIMITER ;

```

## 7.3 Triggers

### 7.3.1 Monitorização de falhas

Trigger	Descrição
Sexo_autorizado	Monitoriza inserções para garantir sexo válido

```

delimiter //

create trigger sexo_autorizado

after insert on atleta

for each row

Begin

    declare msg varchar(50);

    if new.sexo not in('M','F') then

        set msg = 'Tentou usar um sexo incorreto.';

        insert into error_log (error_message) values (msg);

        SIGNAL SQLSTATE '45000'

```

```
SET MESSAGE_TEXT = 'Sexo inválido, use M de masculino ou F de
feminino.';
```

```
end if;
```

```
end //
```

```
delimiter ;
```

### 7.3.2 Outros triggers

Trigger	Descrição
Jogo_clonado	Verifica inserção de jogos e clona se necessário

```
delimiter //
```

```
create trigger jogo_clonado
```

```
after insert on jogo
```

```
for each row
```

```
begin
```

```
declare msg varchar(30);
```

```
if not exists(select 1 from jogo where race_id = new .race_id) then
```

```
set msg = 'ID da prova não encontrado';
```

```
insert into error_log (error_message) values (msg);
```

```
signal sqlstate '49000'
```

```
set message_text = 'ID da prova não encontrado.';
```

```
end if;
```

```
end //
```

```
delimiter ;
```

## 8. Conclusões

O desenvolvimento e implementação deste banco de dados ilustraram a importância de uma modelagem bem planejada e de uma implementação cuidadosa para garantir a eficiência, a integridade e a funcionalidade dos dados.