

Faculdade de Engenharia da Universidade do Porto



Bases de Dados

*Copa dos Libertadores da
América*

2LEIC09 – Grupo 909

João Guedes.....**up202108711**

Rui Pando.....**up202108716**

Gabriel Lindo.....**up202007406**

Resumo

No âmbito desta UC, Bases de Dados, foi-nos proposto criar uma base de dados sobre um tema atribuído pela equipa docente da disciplina. O tema que nos foi atribuído foi a Copa dos Libertadores da América. Para tal, foi necessário criar o modelo conceptual, mapear o modelo para um esquema relacional, implementar o esquema numa base de dados SQLite, introduzir dados e, por fim, interrogar a base de dados.

Modelo Conceptual

Descrição

Na *Copa dos Libertadores da América* competem clubes de vários países da América do Sul, classificados em função dos critérios específicos de cada país.

Sobre cada equipa interessa saber quem são os jogadores que participam na partida, de que país a equipa pertence, se foi eliminada ou classificada e quem ganhou a final.

Participam 47 clubes ao longo de 11 meses (datas), durante 3 fases: a primeira fase (chamada Pré-Libertadores), a segunda fase (fase de grupos) e a fase eliminatória (que é constituída pelas: oitavas-de-final, quartas-de-final, sémis-finais e a final).

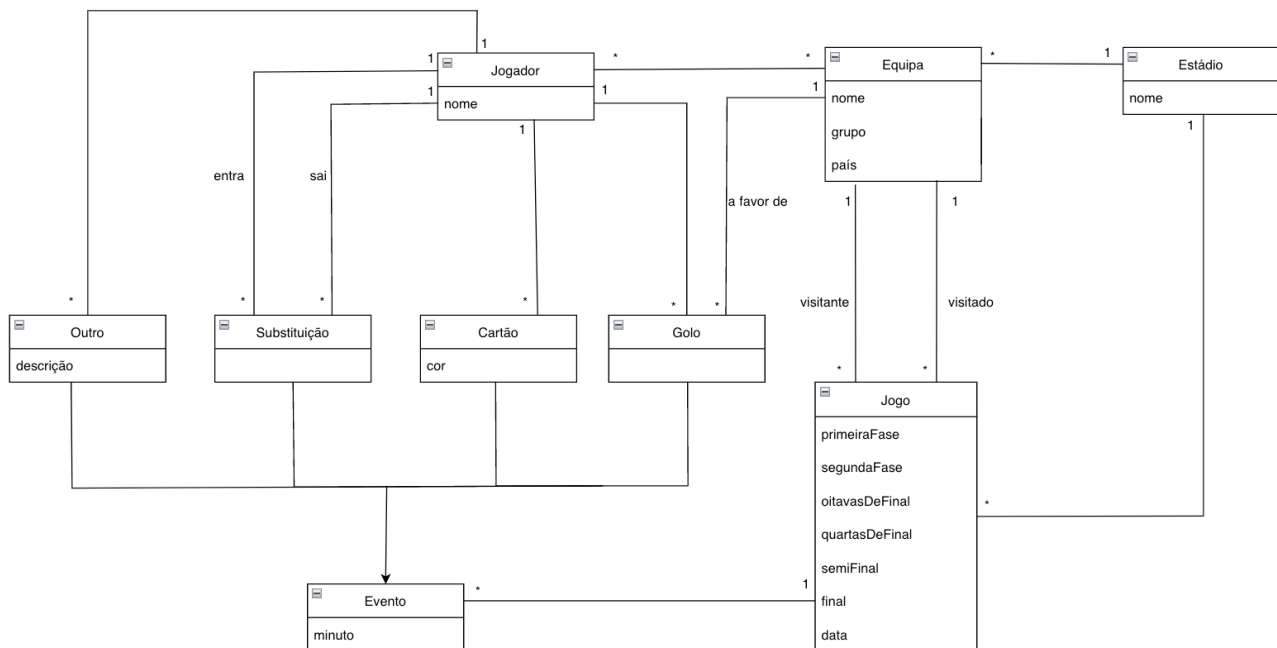
A primeira fase envolve 19 clubes numa pequena fase eliminatória de partidas no sistema ida-e-volta (visitado ou visitante), onde se classificam 4 equipas para a segunda fase.

A segunda fase é dividida em 32 equipas, constituindo 8 grupos com 4 concorrentes cada. As equipas jogam em casa e como visitantes contra todas as equipas do seu respetivo grupo. As duas primeiras equipas de cada grupo são qualificadas para a terceira fase (eliminatória).

A partir desta terceira e última fase, a competição prossegue em subfases, desde as oitavas-de-final até à final. Com cada partida, uma equipa joga como visitante ou como visitante.

Sobre cada jogo, devemos saber quais foram os resultados obtidos, quem marcou os golos, que cartões foram dados (vermelho ou amarelo), quem foi substituído e outros eventos que possam ser importantes para o decorrer do jogo.

Diagrama UML



Esquema Relacional

- **Equipa** (idEquipa, nome, grupo, país, idEstádio -> Estádio)
- **Jogador** (idJogador, nome)
- **Estádio** (idEstádio, nome)
- **Jogo** (idJogo, primeiraFase, segundaFase, oitavasDeFinal, quartasDeFinal, semiFinal, final, data, idEstádio -> Estádio)
- **Evento** (idEvento, minuto, idJogo -> Jogo)
- **Golo** (idEvento -> Evento, idJogador -> Jogador)
- **Cartão** (idEvento -> Evento, cor, idJogador -> Jogador)
- **Substituição** (idEvento -> Evento)
- **Outro** (idEvento -> Evento, descrição, idJogador -> Jogador)
- **JogadorEquipa** (idJogador -> Jogador, idEquipa -> Equipa)
- **Entra** (idEvento -> Evento, idJogador -> Jogador)
- **Sai** (idEvento -> Evento, idJogador -> Jogador)
- **Visitante** (idJogo -> Jogo, idEquipa -> Equipa)
- **Visitado** (idJogo -> Jogo, idEquipa -> Equipa)
- **AFavorDe** (idEvento -> Evento, idEquipa -> Equipa)

Analise Dependências Funcionais & Formas Normais

Equipa (idEquipa, nome, grupo, país, idEstádio -> Estádio)

idEquipa → nome, grupo, país, idEstádio

3ª Forma Normal: Segue

Forma Normal Boyce-Codd: Segue

Jogador (idJogador, nome)

idJogador → nome

3ª Forma Normal: Segue

Forma Normal Boyce-Codd: Segue

Estádio (idEstádio, nome)

idEstádio → nome

3ª Forma Normal: Segue

Forma Normal Boyce-Codd: Segue

Jogo (idJogo, primeiraFase, segundaFase, oitavasDeFinal, quartasDeFinal, semiFinal, final, data, idEstádio -> Estádio)

idJogo → primeiraFase, segundaFase, oitavasDeFinal, semiFinal, final, data, idEstádio -> Estádio

3ª Forma Normal: Segue

Forma Normal Boyce-Codd: Segue

Evento (idEvento, minuto, idJogo -> Jogo)

idEvento → minuto, idJogo -> Jogo

3ª Forma Normal: Segue

Forma Normal Boyce-Codd: Segue

Golo (idEvento -> Evento, idJogador -> Jogador)

idEvento → idJogador

3ª Forma Normal: Segue

Forma Normal Boyce-Codd: Segue

Cartão (idEvento -> Evento, cor, idJogador -> Jogador)

idEvento → cor, idJogador

3ª Forma Normal: Segue

Forma Normal Boyce-Codd: Segue

Substituição (idEvento -> Evento)

...

3ª Forma Normal: Segue

Forma Normal Boyce-Codd: Segue

Outro (idEvento -> Evento, descrição, idJogador -> Jogador)
idEvento → descrição, idJogador
3ª Forma Normal: Segue
Forma Normal Boyce-Codd: Segue

JogadorEquipa (idJogador -> Jogador, idEquipa -> Equipa)
...
3ª Forma Normal: Segue
Forma Normal Boyce-Codd: Segue

Entra (idEvento -> Evento, idJogador -> Jogador)
idEvento → idJogador
3ª Forma Normal: Segue
Forma Normal Boyce-Codd: Segue

Sai (idEvento -> Evento, idJogador -> Jogador)
idEvento → idJogador
3ª Forma Normal: Segue
Forma Normal Boyce-Codd: Segue

Visitante (idJogo -> Jogo, idEquipa -> Equipa)
idJogo → idEquipa
3ª Forma Normal: Segue
Forma Normal Boyce-Codd: Segue

Visitado (idJogo -> Jogo, idEquipa -> Equipa)
idJogo → idEquipa
3ª Forma Normal: Segue
Forma Normal Boyce-Codd: Segue

AFavorDe (idEvento -> Evento, idEquipa -> Equipa)
idEvento → idEquipa
3ª Forma Normal: Segue
Forma Normal Boyce-Codd: Segue

Formal Normal Boyce-Codd: consiste em definir uma Dependência Funcional como tendo no seu lado esquerdo uma *superkey*; $A \rightarrow B$ (A é *superkey*).

3ª Forma Normal: define-se como tendo, “para cada Dependência Funcional, ou o lado esquerdo tem um *superkey* ou o lado direito tem *prime attributes* (atributo que é membro duma *key*).”

Lista e Forma de Implementação das Restrições

Tabela **EQUIPA**:

- **ID** - CONSTRAINT EQUIPA_PK PRIMARY KEY(ID): não pode haver duas equipas com o mesmo ID;
- **NOME** - NOT NULL UNIQUE: não pode haver duas equipas com o mesmo nome;
- **PAIS** - NOT NULL: a cada equipa deve estar associado um país a que pertença.

Tabela **JOGADOR**:

- **ID** - CONSTRAINT JOGADOR_PK PRIMARY KEY(ID): não pode haver dois jogadores com o mesmo ID;
- **EQUIPA** - CONSTRAINT EQUIPA_FK FOREIGN KEY(EQUIPA) REFERENCES EQUIPA(NOME): a cada jogador deve estar associada uma equipa;
- **NOME** - NOT NULL UNIQUE: não pode haver dois jogadores com o mesmo nome.

Tabela **JOGO**:

- **ID** - CONSTRAINT JOGO_PK PRIMARY KEY(ID): não pode haver dois jogos com o mesmo ID;
- **JORNADA** - NOT NULL: cada jogo deve-se dar numa determinada jornada;
- **DATA** - NOT NULL: cada jogo deve-se dar numa determinada data;
- **equipaCASA** - CONSTRAINT EQUIPA_FK FOREIGN KEY(equipaCASA) REFERENCES EQUIPA(NOME): em cada jogo há uma equipa que joga em casa;
- **equipaVIS** - CONSTRAINT EQUIPA_FK FOREIGN KEY(equipaVIS) REFERENCES EQUIPA(NOME): em cada jogo há uma equipa a visitar;
- **CASares** - NOT NULL: em cada jogo, a equipa que jogou em casa deve ter obtido um resultado, mesmo que este seja 0;
- **VISres** - NOT NULL: em cada jogo, a equipa que jogou a visitar deve ter obtido um resultado, mesmo que este seja 0;

Tabela **GOLO**:

- **ID** - CONSTRAINT GOLO_PK PRIMARY KEY(ID): não pode haver dois golos com o mesmo ID;
- **MINUTO** - NOT NULL: cada golo deve ter-se dado num determinado minuto do jogo;
- **EQUIPAfav** - CONSTRAINT EQUIPA_FK FOREIGN KEY(EQUIPAfav) REFERENCES EQUIPA(NOME): cada golo deve ter favorecido o resultado de uma das duas equipas a jogar;
- **MARCADOR** - CONSTRAINT JOGADOR_FK FOREIGN KEY(MARCADOR) REFERENCES JOGADOR(NOME): cada golo deve ter sido marcado por um jogador de uma das duas equipas a jogar;
- **IDjogo** - CONSTRAINT JOGO_FK FOREIGN KEY(IDjogo) REFERENCES JOGO(ID): cada golo deve ter sido marcado num determinado jogo.

Tabela **EVENTO**:

- **ID** - CONSTRAINT EVENTO_PK PRIMARY KEY(ID): não pode haver dois eventos com o mesmo ID;
- **TIPO** - NOT NULL, CONSTRAINT CHK_EVENTO CHECK(TIPO='Cartão' OR TIPO='Substituição'): cada evento deve ter sido ou um cartão, ou uma substituição;
- **MINUTO** - NOT NULL: cada evento deve ter-se dado num determinado minuto do jogo;
- **IDjogo** - CONSTRAINT JOGO_FK FOREIGN KEY(IDjogo) REFERENCES JOGO(ID): cada evento deve ter ocorrido num determinado jogo na tabela JOGO.

Tabela **CARTAO**:

- **ID** - CONSTRAINT CARTAO_PK PRIMARY KEY(ID): não pode haver dois cartões com o mesmo ID;
- **COR** - NOT NULL, CONSTRAINT CHK_CARTAO CHECK(COR='Amarelo' OR COR='Vermelho'): cada cartão deve ter sido ou amarelo, ou vermelho;
- **EQUIPA** - CONSTRAINT EQUIPA_FK FOREIGN KEY(EQUIPA) REFERENCES EQUIPA(NOME): cada cartão deve ter sido aplicado a um jogador, pertencente a uma determinada equipa;
- **JOGADOR** - CONSTRAINT JOGADOR_FK FOREIGN KEY(JOGADOR) REFERENCES JOGADOR(NOME): cada cartão deve ter sido aplicado a um jogador;
- **IDevento** - CONSTRAINT EVENTO_FK FOREIGN KEY(IDevento) REFERENCES EVENTO(ID): cada cartão deve corresponder a um determinado evento na tabela EVENTO.

Tabela **SUBSTITUICAO**:

- **ID** - CONSTRAINT SUBSTITUICAO_PK PRIMARY KEY(ID): não pode haver duas substituições com o mesmo ID;
- **EQUIPA** - CONSTRAINT EQUIPA_FK FOREIGN KEY(EQUIPA) REFERENCES EQUIPA(NOME): cada substituição deve ter ocorrido dentro de uma determinada equipa;
- **JOGADORent** - CONSTRAINT JOGADOR_FK FOREIGN KEY(JOGADORent) REFERENCES JOGADOR(NOME): em cada substituição, deve haver um jogador a entrar em campo;
- **JOGADORsai** - CONSTRAINT JOGADOR_FK FOREIGN KEY(JOGADORsai) REFERENCES JOGADOR(NOME): em cada substituição, deve haver um jogador a sair de campo;
- **IDevento** - CONSTRAINT EVENTO_FK FOREIGN KEY(IDevento) REFERENCES EVENTO(ID): cada substituição deve corresponder a um determinado evento na tabela EVENTO.

Tabela **ESTADIO**:

- **ID** - CONSTRAINT ESTADIO_PK PRIMARY KEY(ID): não pode haver dois estádios com o mesmo ID;
- **NOME** - NOT NULL: cada jogo deve ter decorrido num estádio, que por sua vez deve ter um nome;
- **PAIS** - NOT NULL: cada estádio deve ser localizado num determinado país;
- **IDjogo** - CONSTRAINT JOGO_FK FOREIGN KEY(IDjogo) REFERENCES JOGO(ID): cada estádio deve estar associado a um jogo da tabela JOGO.

Interrogação da Base de Dados

1. Quantos jogos decorreram durante a Copa dos Libertadores?
(*int1.sql*)
2. Quais foram as cinco equipas que mais golos marcaram?
(*int2.sql*)
3. Quais jogadores foram substituídos depois do intervalo (45 minutos) e em qual jogo?
(*int3.sql*)
4. Quantos cartões amarelos foram dados nas semifinais?
(*int4.sql*)
5. Em quais estádios foram jogados os últimos jogos e quais eventos ocorreram nesses mesmos jogos e em quais minutos?
(*int5.sql*)
6. Quais foram os cinco jogadores que mais golos marcaram, quantos golos marcaram e de qual equipa eram?
(*int6.sql*)
7. A equipa Palmeiras perdeu em que jogos e contra quais equipas?
(*int7.sql*)
8. Quais equipas tiveram menos golos nas quartas-de-final, em ordem decrescente?
(*int8.sql*)
9. Quais equipas entraram nas oitavas-de-final, mas foram desqualificadas antes de chegar às quartas-de-final?
(*int9.sql*)
10. Quais jogadores do Atlético PR não receberam nenhum cartão (amarelo ou vermelho) durante a Copa e marcaram golos?
(*int10.sql*)

Adição de gatilhos à base de dados

1º Gatilho - Quando é eliminada uma entrada na tabela **JOGADOR**, quando não houver mais nenhum jogador na sua respetiva equipa, a entrada dessa equipa na tabela **EQUIPA** é eliminada também.

2º Gatilho - Quando é inserida uma nova entrada na tabela **GOLO**, quando o jogador que marcou o golo ainda não tem uma entrada respetiva na tabela **JOGADOR**, ou seja, o seu nome ainda não existe na base de dados como jogador, é inserida uma nova entrada na tabela **JOGADOR** com o valor de **ID** seguinte, o **NOME** como o do jogador que marcou o golo, e a sua **EQUIPA** como a equipa favorecida pelo golo (nota-se que se o jogador que marcou o golo não pertence, na realidade, à equipa favorecida pelo golo, como é no caso de um autogolo, esta entrada na tabela **JOGADOR** inserida pelo gatilho torna-se incorreta).

3º Gatilho - Quando é inserida uma nova entrada na tabela **GOLO**, quando a equipa favorecida por este golo não é nenhuma das duas equipas a jogar no jogo em questão, a ação é abortada, e a mensagem '**Equipa favorecida pelo golo deve ser uma das equipas a participar no jogo em questão**' aparece na linha de comandos.

Avaliação da Participação dos Vários Elementos do Grupo

O projeto foi dividido igualmente entre os elementos do grupo: João Guedes e Gabriel Lindo.