

**OTÁVIO MORATELLI
ROBERTA SILVANO PEREIRA
VINÍCIUS ROVER**

**GERENCIAMENTO DE UM TIME DE CHEERLEADING
Trabalho final de Banco de dados
Professor: Alexandre Leopoldo Gonçalves**

ARARANGUÁ - 28/11/2020

SUMÁRIO

- 1. DESCRIÇÃO DO OBJETIVO GERAL DO SISTEMA**
- 2. DESCRIÇÃO DETALHADA DO SISTEMA**
- 3. MODELO CONCEITUAL**
- 4. MODELO LÓGICO**
- 5. SCRIPT DDL**
- 6. CONSULTAS ESPECÍFICAS**
 - 6.1. Quantos atletas existem em cada posição**
 - 6.2. Quantidade de quantos projetos cada diretoria tem**
 - 6.3. Média da mensalidade de cada atleta que possua média maior que R\$20.00**
- 7. DESENVOLVIMENTO DE UMA APLICAÇÃO**
 - 7.1. Seleção da tabela sobre a qual se quer trabalhar**
 - 7.2. Seleção de qual a ação que deverá ser feita sobre a tabela escolhida**
 - 7.3. Caso a ação seja Select, Insert ou Update, é possível escolher as colunas desejadas**
 - 7.4. Ação Select**
 - 7.5. Ação Insert**
 - 7.6. Ação Update**
 - 7.7. Ação Delete**

1. DESCRIÇÃO DO OBJETIVO GERAL DO SISTEMA

O sistema deverá gerenciar uma equipe de *Cheerleading*, auxiliando a sua diretoria a cuidar de seus vários aspectos, como controle de faltas, funções dentro da equipe, gerenciamento de fluxo de caixa, calendário de eventos, informações de cada atleta, entre outros.

2. DESCRIÇÃO DETALHADA DO SISTEMA

O *Cheerleading* é um esporte, reconhecido pelo Comitê Olímpico Internacional (COI), praticado em grupo, que envolve diversas habilidades e é considerado bem perigoso. Ele vem crescendo e se popularizando muito no Brasil nos últimos anos, já contando com mais de 700 times.

Todos os times possuem diversas regras, de modo que visem a evolução e segurança de seus atletas. Os times sempre possuem diretorias que são responsáveis por gerenciar essas regras. O sistema tem como objetivo auxiliar o trabalho da diretoria.

É necessário ter um controle dos atletas no time, possuindo informações como:

- A. nome completo
- B. data de nascimento
- C. tipo sanguíneo
- D. informações para o caso de uma emergência, como nome e telefone. É possível que cada atleta tenha mais de um contato de emergência
- E. quais medicamentos o atleta toma e seus efeitos colaterais
- F. as doenças e alergias que possui
- G. qual seu convênio médico
- H. quais suas posições dentro do time e quais *skills* possui.

Além disso, a diretoria é formada por atletas do time, e é necessário registrar:

- A. qual o respectivo cargo, sendo que cada atleta pode assumir diversas diretorias, mas cada diretoria só possui 1 atleta
- B. quais os projetos desenvolvidos pelas diretorias, com uma descrição e data. Os projetos podem possuir mais de um diretor envolvido.

Os times costumam participar de alguns eventos, e é necessário saber:

- A. o nome do evento, a data, quais atletas serão alocados e em quais posições.

Visto que o esporte é em grupo, e que para algumas habilidades é necessário diversos atletas, é exigido um controle bem rigoroso de presença nos treinos. A cada data de treino é necessário que o sistema registre:

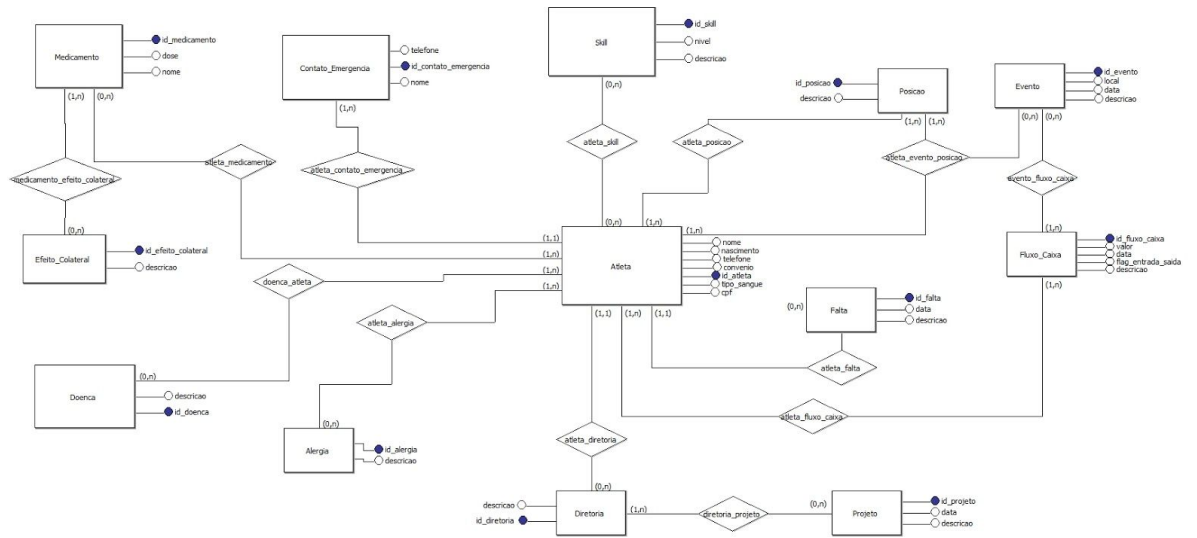
- A. os atletas que faltaram
- B. uma descrição do motivo da falta

Também é necessário ter um controle do fluxo de caixa. Podem existir saídas e entradas de dinheiro, que podem estar relacionadas com:

- A. eventos que o time participa
- B. pagamento de mensalidades pelos atletas
- C. com outras necessidades.

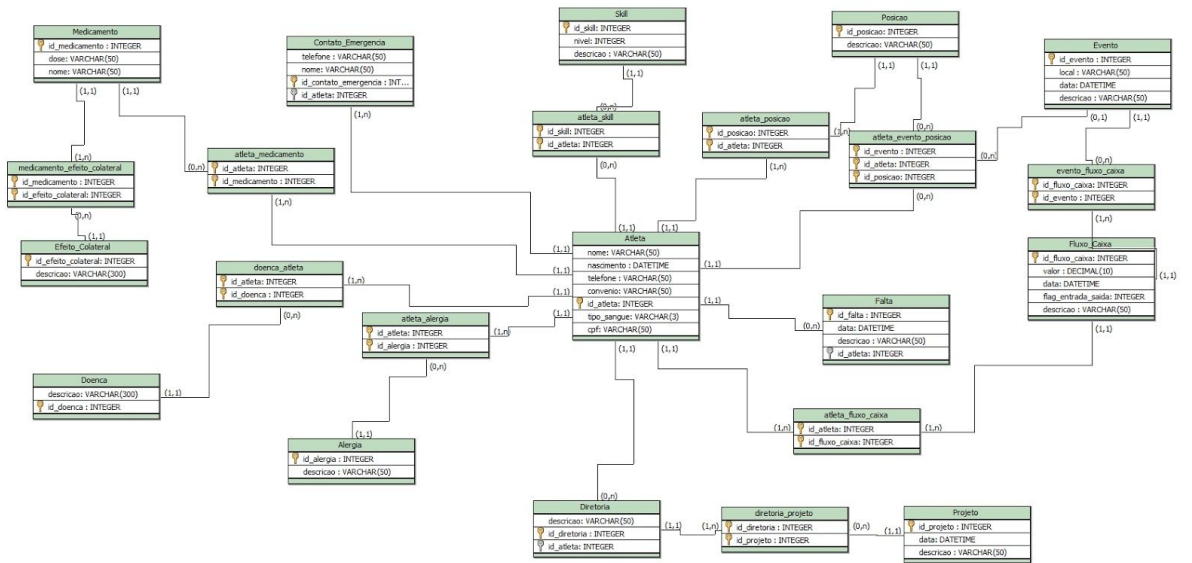
3. MODELO CONCEITUAL

O arquivo está anexado abaixo, mas também se encontra no zip do trabalho.



4. MODELO LÓGICO

O arquivo está anexado abaixo, mas também se encontra no zip do trabalho.



5. SCRIPT DDL

O arquivo está anexado abaixo, mas também se encontra no zip do trabalho.

```
CREATE TABLE Medicamento (  
  id_medicamento INTEGER PRIMARY KEY NOT NULL,  
  dose VARCHAR(50) NOT NULL,  
  nome VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE Efeito_Colateral (  
  id_efeito_colateral INTEGER PRIMARY KEY NOT NULL,  
  descricao VARCHAR(300) NOT NULL  
);
```

```
CREATE TABLE Doenca (  
  descricao VARCHAR(300) NOT NULL,  
  id_doenca INTEGER PRIMARY KEY NOT NULL  
);
```

```
CREATE TABLE Atleta (  
  nome VARCHAR(50) NOT NULL,  
  nascimento TIMESTAMP NOT NULL,  
  telefone VARCHAR(50) NOT NULL,  
  convenio VARCHAR(50) NOT NULL,  
  id_atleta INTEGER PRIMARY KEY NOT NULL,  
  tipo_sangue VARCHAR(3) NOT NULL,  
  cpf VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE Alergia (  
  id_alergia INTEGER PRIMARY KEY NOT NULL,  
  descricao VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE Diretoria (  
  descricao VARCHAR(50) NOT NULL,  
  id_diretoria INTEGER PRIMARY KEY NOT NULL,  
  id_atleta INTEGER NOT NULL,  
  FOREIGN KEY(id_atleta) REFERENCES Atleta (id_atleta)  
);
```

```
CREATE TABLE Projeto (  
  id_projeto INTEGER PRIMARY KEY NOT NULL,  
  data TIMESTAMP NOT NULL,  
  descricao VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE Contato_Emergencia (  
  id_contato_emergencia INTEGER PRIMARY KEY NOT NULL,  
  id_atleta INTEGER NOT NULL,  
  id_doenca INTEGER NOT NULL,  
  id_efeito_colateral INTEGER NOT NULL,  
  id_alergia INTEGER NOT NULL,  
  id_diretoria INTEGER NOT NULL,  
  id_medicamento INTEGER NOT NULL,  
  id_projeto INTEGER NOT NULL,  
  id_tipo_sangue INTEGER NOT NULL,  
  id_cpf INTEGER NOT NULL  
);
```

```
telefone VARCHAR(50) NOT NULL,  
nome VARCHAR(50) NOT NULL,  
id_contato_emergencia INTEGER PRIMARY KEY NOT NULL,  
id_atleta INTEGER NOT NULL,  
FOREIGN KEY(id_atleta) REFERENCES Atleta (id_atleta)  
);
```

```
CREATE TABLE Skill (  
id_skill INTEGER PRIMARY KEY NOT NULL,  
nivel INTEGER NOT NULL,  
descricao VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE Evento (  
id_evento INTEGER PRIMARY KEY NOT NULL,  
local VARCHAR(50) NOT NULL,  
data TIMESTAMP NOT NULL,  
descricao VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE Falta (  
id_falta INTEGER PRIMARY KEY NOT NULL,  
data TIMESTAMP NOT NULL,  
descricao VARCHAR(50) NOT NULL,  
id_atleta INTEGER NOT NULL,  
FOREIGN KEY(id_atleta) REFERENCES Atleta (id_atleta)  
);
```

```
CREATE TABLE Fluxo_Caixa (  
id_fluxo_caixa INTEGER PRIMARY KEY NOT NULL,  
valor DECIMAL(10) NOT NULL,  
data TIMESTAMP NOT NULL,  
flag_entrada_saida INTEGER NOT NULL,  
descricao VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE atleta_fluxo_caixa (  
id_atleta INTEGER NOT NULL,  
id_fluxo_caixa INTEGER NOT NULL,  
PRIMARY KEY(id_atleta,id_fluxo_caixa),  
FOREIGN KEY(id_atleta) REFERENCES Atleta (id_atleta),  
FOREIGN KEY(id_fluxo_caixa) REFERENCES Fluxo_Caixa (id_fluxo_caixa)  
);
```

```
CREATE TABLE medicamento_efeito_colateral (  
id_medicamento INTEGER NOT NULL,  
id_efeito_colateral INTEGER NOT NULL,  
PRIMARY KEY(id_medicamento ,id_efeito_colateral),
```



```
FOREIGN KEY(id_medicamento ) REFERENCES Medicamento (id_medicamento),  
FOREIGN KEY(id_efeito_colateral) REFERENCES Efeito_Colateral  
(id_efeito_colateral)  
);
```

```
CREATE TABLE atleta_medicamento (  
id_atleta INTEGER NOT NULL,  
id_medicamento INTEGER NOT NULL,  
PRIMARY KEY(id_atleta,id_medicamento),  
FOREIGN KEY(id_atleta) REFERENCES Atleta (id_atleta),  
FOREIGN KEY(id_medicamento ) REFERENCES Medicamento (id_medicamento)  
);
```

```
CREATE TABLE atleta_alergia (  
id_atleta INTEGER NOT NULL,  
id_alergia INTEGER NOT NULL,  
PRIMARY KEY(id_atleta,id_alergia),  
FOREIGN KEY(id_atleta) REFERENCES Atleta (id_atleta),  
FOREIGN KEY(id_alergia ) REFERENCES Alergia (id_alergia)  
);
```

```
CREATE TABLE diretoria_projeto (  
id_diretoria INTEGER NOT NULL,  
id_projeto INTEGER NOT NULL,  
PRIMARY KEY(id_diretoria ,id_projeto),  
FOREIGN KEY(id_diretoria ) REFERENCES Diretoria (id_diretoria),  
FOREIGN KEY(id_projeto ) REFERENCES Projeto (id_projeto)  
);
```

```
CREATE TABLE evento_fluxo_caixa (  
id_fluxo_caixa INTEGER NOT NULL,  
id_evento INTEGER NOT NULL,  
PRIMARY KEY(id_fluxo_caixa,id_evento),  
FOREIGN KEY(id_fluxo_caixa) REFERENCES Fluxo_Caixa (id_fluxo_caixa),  
FOREIGN KEY(id_evento ) REFERENCES Evento (id_evento)  
);
```

```
CREATE TABLE atleta_posicao (  
id_posicao INTEGER NOT NULL,  
id_atleta INTEGER NOT NULL,  
PRIMARY KEY(id_posicao,id_atleta),  
FOREIGN KEY(id_atleta) REFERENCES Atleta (id_atleta)  
);
```

```
CREATE TABLE atleta_skill (  
id_skill INTEGER NOT NULL,  
id_atleta INTEGER NOT NULL,  
PRIMARY KEY(id_skill,id_atleta),
```

```
FOREIGN KEY(id_skill) REFERENCES Skill (id_skill),  
FOREIGN KEY(id_atleta) REFERENCES Atleta (id_atleta)  
);
```

```
CREATE TABLE Posicao (  
id_posicao INTEGER PRIMARY KEY NOT NULL,  
descricao VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE atleta_evento_posicao (  
id_evento INTEGER NOT NULL,  
id_atleta INTEGER NOT NULL,  
id_posicao INTEGER NOT NULL,  
PRIMARY KEY(id_evento ,id_atleta,id_posicao),  
FOREIGN KEY(id_evento ) REFERENCES Evento (id_evento),  
FOREIGN KEY(id_atleta) REFERENCES Atleta (id_atleta),  
FOREIGN KEY(id_posicao) REFERENCES Posicao (id_posicao)  
);
```

```
CREATE TABLE doenca_atleta (  
id_atleta INTEGER NOT NULL,  
id_doenca INTEGER NOT NULL,  
PRIMARY KEY(id_atleta,id_doenca),  
FOREIGN KEY(id_atleta) REFERENCES Atleta (id_atleta),  
FOREIGN KEY(id_doenca ) REFERENCES Doenca (id_doenca)  
);
```

```
ALTER TABLE atleta_posicao ADD FOREIGN KEY(id_posicao) REFERENCES  
Posicao (id_posicao);
```

6. CONSULTAS ESPECÍFICAS

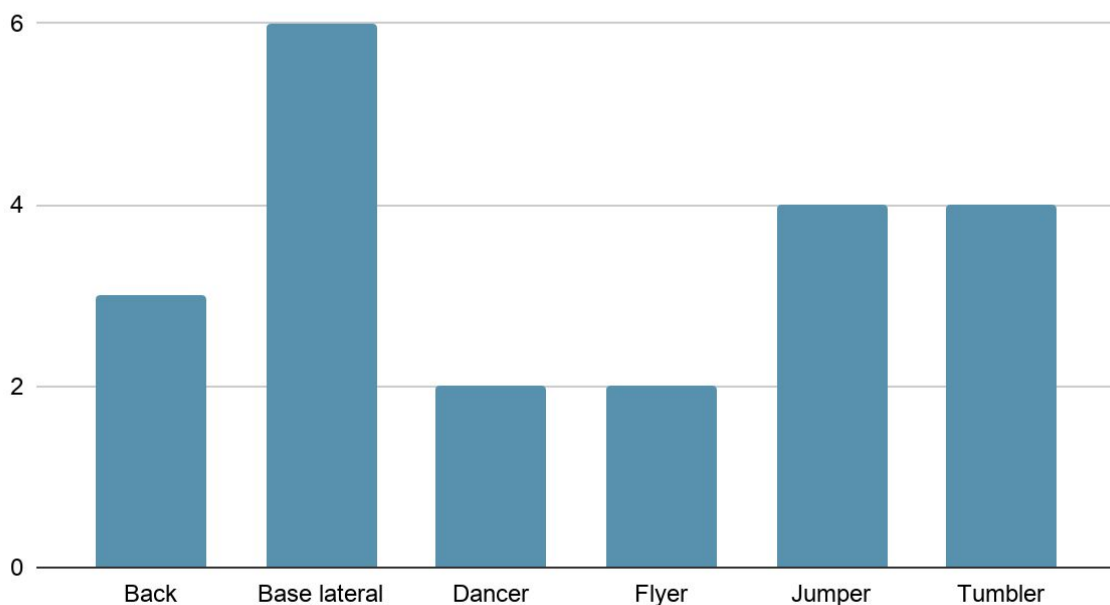
Para realizar as consultas específicas, foram utilizadas funções de agregação e 3 tabelas em cada consulta. Para uma melhor visualização dos dados, utilizamos também a função *string_agg()* nas consultas 1 e 2.

6.1. QUANTOS ATLETAS EXISTEM EM CADA POSIÇÃO

```
SELECT p.descricao AS posicao, count(a.id_atleta) AS qtd_atleta,  
string_agg(a.nome, ' ') AS atletas FROM atleta_posicao AS ap  
LEFT JOIN atleta AS a ON a.id_atleta = ap.id_atleta  
LEFT JOIN posicao AS p ON p.id_posicao = ap.id_posicao  
GROUP BY p.descricao  
ORDER BY p.descricao;
```

posicao character varying (50)	qtd_atleta bigint	atletas text
back	3	Raul Brum Weschenfelder, Augusto Prudencio, Victor Seabra
base lateral	6	Otavio Moratelli, Nicolas Beraldo, Camila Rodrigues, Bruna Gonçalves, Augusto Prudencio, Ricardo Battaglin
dancer	2	Roberta Silvano Pereira, Emanuelle Lizuka
flyer	2	Roberta Silvano Pereira, Emanuelle Lizuka
jumper	4	Victor Seabra, Raul Brum Weschenfelder, Ricardo Battaglin, Emanuelle Lizuka
tumbler	4	Ricardo Battaglin, Roberta Silvano Pereira, Raul Brum Weschenfelder, Augusto Prudencio

Posição x Atleta

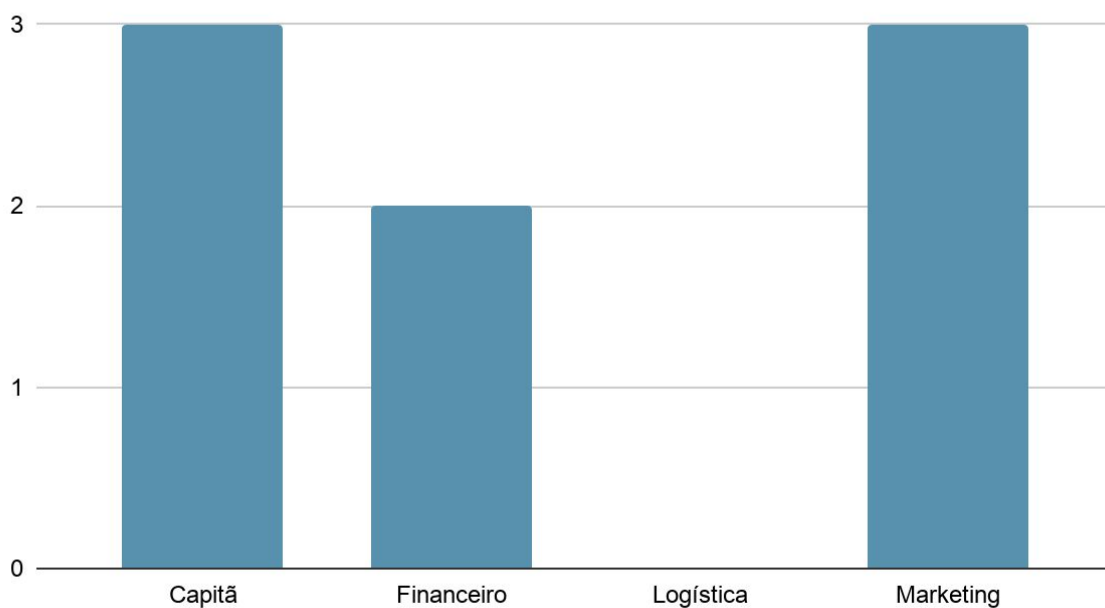


6.2. QUANTIDADE DE QUANTOS PROJETOS CADA DIRETORIA TEM

```
SELECT d.descricao AS diretorio, count(p.id_projeto) AS qtd_projeto,  
string_agg(p.descricao, ' ') AS projetos FROM diretorio AS d  
LEFT JOIN diretorio_projeto AS dp ON dp.id_diretorio = d.id_diretorio  
LEFT JOIN projeto AS p ON p.id_projeto = dp.id_projeto  
GROUP BY d.descricao  
ORDER BY d.descricao;
```

diretorio character varying (50)	qtd_projeto bigint	projetos text
Capitã	3	Extensão, Festa Junina, Tryout
Financeiro	2	Meias, Festa Junina
Logística	0	[null]
Mkt	3	Meias, Festa Junina, Tryout

Projetos x Diretoria

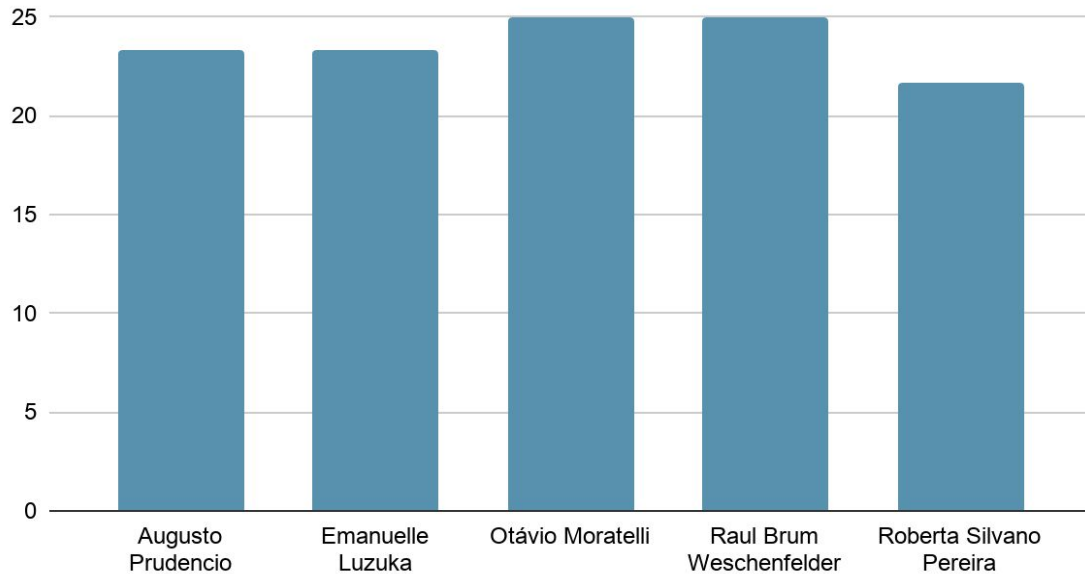


6.3. MÉDIA DA MENSALIDADE DE CADA ATLETA QUE POSSUA MÉDIA MAIOR QUE R\$20.00

```
SELECT a.nome, avg(fc.valor) AS media_mensalidade FROM atleta AS a
LEFT JOIN atleta_fluxo_caixa AS afc ON afc.id_atleta = a.id_atleta
LEFT JOIN fluxo_caixa AS fc ON fc.id_fluxo_caixa = afc.id_fluxo_caixa
WHERE fc.descricao = 'mensalidade'
GROUP BY a.nome
HAVING avg(fc.valor) > 20 ORDER BY a.nome;
```

nome	media_mensalidade
character varying (50)	numeric
Augusto Prudencio	23.333333333333333
Emanuelle Lizuka	23.333333333333333
Otávio Moratelli	25.000000000000000
Raul Brum Weschenfelder	25.000000000000000
Roberta Silvano Pereira	21.666666666666667

Atleta x Mensalidade



7. DESENVOLVIMENTO DE UMA APLICAÇÃO

Segue abaixo alguns prints da aplicação em execução.

7.1. Seleção da tabela sobre a qual se quer trabalhar

```
OPÇÕES DE TABELA:
[1] - alergias
[2] - atleta
[3] - atleta_alergia
[4] - atleta_evento_posicao
[5] - atleta_fluxo_caixa
[6] - atleta_medicamento
[7] - atleta_posicao
[8] - atleta_skill
[9] - contato_emergencia
[10] - diretoria
[11] - diretoria_projeto
[12] - doenca
[13] - doenca_atleta
[14] - efeito_colateral
[15] - evento
[16] - evento_fluxo_caixa
[17] - falta
[18] - fluxo_caixa
[19] - medicamento
[20] - medicamento_efeito_colateral
[21] - posicao
[22] - projeto
[23] - skill
[24] - Sair

Escolha a tabela desejada: 2
Você escolheu a tabela [2] - atleta
```

7.2. Seleção de qual a ação que deverá ser realizada

```
OPÇÕES DE AÇÕES:
[1] - SELECT
[2] - INSERT
[3] - UPDATE
[4] - DELETE
[5] - CONSULTA ESPECÍFICA 1: Quantos atletas existem em cada posição
[6] - CONSULTA ESPECÍFICA 2: Quantidade de projetos que cada diretoria tem
[7] - CONSULTA ESPECÍFICA 3: Média da mensalidade de cada atleta que possua média maior que R$20.00
[8] - SAIR
[9] - RETORNAR AO MENU DE TABELAS

Escolha a ação desejada: 1
Você escolheu a opção [1] - SELECT
```

7.3. Caso a ação seja Select, Insert ou Update, é possível escolher as colunas desejadas

É possível selecionar as seguintes colunas da tabela atleta. [Colunas id são selecionadas automaticamente]:

- [*] - Todas
- [0] - nascimento
- [1] - id_atleta
- [2] - nome
- [3] - tipo_sangue
- [4] - cpf
- [5] - convenio
- [6] - telefone
- [7] - Voltar ao menu de tabelas
- [8] - Voltar ao menu de ações

Insira o número das colunas que você deseja selecionar no formato número_coluna1, número_coluna2, etc. Por exemplo: 1, 2, 3 [Valores inválidos são ignorados]: *

As colunas que você selecionou foram: ['nascimento', 'id_atleta', 'nome', 'tipo_sangue', 'cpf', 'convenio', 'telefone']

7.4. Ação Select

As colunas que você selecionou foram: ['nascimento', 'id_atleta', 'nome']

RESULTADO:

NASCIMENTO	ID_ATELETA	NOME
1996-07-01	1	Roberta Silvano Pereira
1999-10-10	2	Raul Brum Weschenfelder
2000-02-15	3	Otavio Moratelli
1999-08-25	4	Augusto Prudencio
2000-04-09	5	Ricardo Battaglin
1997-05-03	6	Nicolas Beraldo
2000-12-15	7	Victor Seabra
2000-01-01	8	Emanuelle Lizuka
2000-06-23	9	Camila Rodrigues
2001-08-23	10	Bruna Gonçalves

7.5. Ação Insert

```
Você
A ta Problems (Ctrl+Shift+M) - Total 0 Problems
Insção possui os seguintes dados:

RESULTADO:

Essa tabela está vazia

Pressione enter para continuar

A tabela falta possui chave estrangeira! Seguem os códigos da tabela de referência atleta abaixo:
ID_ATLETA      NOME
1      Roberta Silvano Pereira
2      Raul Brum Weschenfelder
3      Otavio Moratelli
4      Augusto Prudencio
5      Ricardo Battaglin
6      Nicolas Beraldo
7      Victor Seabra
8      Emanuelle Lizuka
9      Camila Rodrigues
10     Bruna Gonçalves

INSIRA OS VALORES:
Identificador id falta preenchido automaticamente
DATA: 10-11-2020
ID_ATLETA [CHAVE ESTRANGEIRA]: 3
DESCRICAO: estava doente
Insert realizado com sucesso!

Deseja dar insert em outro registro desta mesma tabela? [Y/N]
```


7.6. Ação Update

```
Você escolheu a opção [3] - UPDATE

É possível selecionar as seguintes colunas da tabela skill. [Colunas id são selecionadas automaticamente]:
[*] - Todas
[0] - id_skill
[1] - nivel
[2] - descricao
[3] - Voltar ao menu de tabelas
[4] - Voltar ao menu de ações

Insira o número das colunas que você deseja selecionar no formato número_coluna1, número_coluna2, etc. Por exemplo: 1, 2, 3 [Valores inválidos são ignorados]:2

As colunas que você selecionou foram: ['id_skill', 'descricao']

A tabela que você deseja dar update possui os seguintes dados:
ID_SKILL  NIVEL  DESCRICAO
1         2      subida no quick
2         7      mortalzinho
3         2      extensão 2 pés
4         1  tic toc conectado abaixo da prep
5         2      lib na altura de prep

Insira o id do registro (id_skill) que você deseja dar update: 4
Insira os valores atualizados:
ID_SKILL:  Você não possui permissão para alterar o identificador
DESCRICAO: tic toc conectado
Update realizado com sucesso!

Deseja dar update em outro registro desta mesma tabela? [Y/N] █
```

7.7. Ação Delete

```
Você escolheu a opção [4] - DELETE

A tabela evento contem os seguintes dados:
ID_EVENTO  DATA  LOCAL  DESCRICAO
1  2021-10-10  cricumã  jisc
2  2021-10-17  curitiba  campeonato paranaense

Insira o id do registro (id_evento) que você deseja deletar: 1

Tem certeza que deseja deletar id_evento = [1]? [Y/N] y
id_evento = [1] deletada com sucesso!

Deseja deletar outro registro desta mesma tabela? [Y/N] █
```