

Universidade Federal de São Paulo – UNIFESP  
Campus: Parque Tecnológico – São José dos Campos



Instituto de Ciência e Tecnologia – ICT

Bacharelado em Ciência e Tecnologia – BCT

**Banco de Dados**

## **O Poderoso TOR**

Prof<sup>a</sup>. Dra. Daniela Leal Musa

**Enzo de Almeida Belfort Rizzi Di Chiara**

**RA: 168813**

**João Pedro da Silva Zampoli**

**RA: 168880**

**Luiza de Souza Ferreira**

**RA: 170453**

**Viviane Flor Park**

**RA: 169259**

São José dos Campos, 2025

## SUMÁRIO

<b>1. Sistema de Gerenciamento de Banco de Dados.....</b>	<b>3</b>
<b>2. Descrição do projeto.....</b>	<b>3</b>
<b>3. Modelo Entidade Relacionamento.....</b>	<b>4</b>
<b>4. Modelo Lógico.....</b>	<b>4</b>
<b>5 Implementação do Banco de Dados.....</b>	<b>5</b>
5.1 Criação das Tabelas.....	5
5.2 Inserção dos Dados nas Tabelas.....	8
<b>6 Desenvolvimento de Consultas SQL.....</b>	<b>18</b>
6.1 Consultas Básicas.....	18
6.2 Consultas com LEFT JOIN.....	19
6.3 Consultas Agregadas.....	20
<b>7 Desenvolvimento da Interface Gráfica.....</b>	<b>22</b>

## **1. Sistema de Gerenciamento de Banco de Dados**

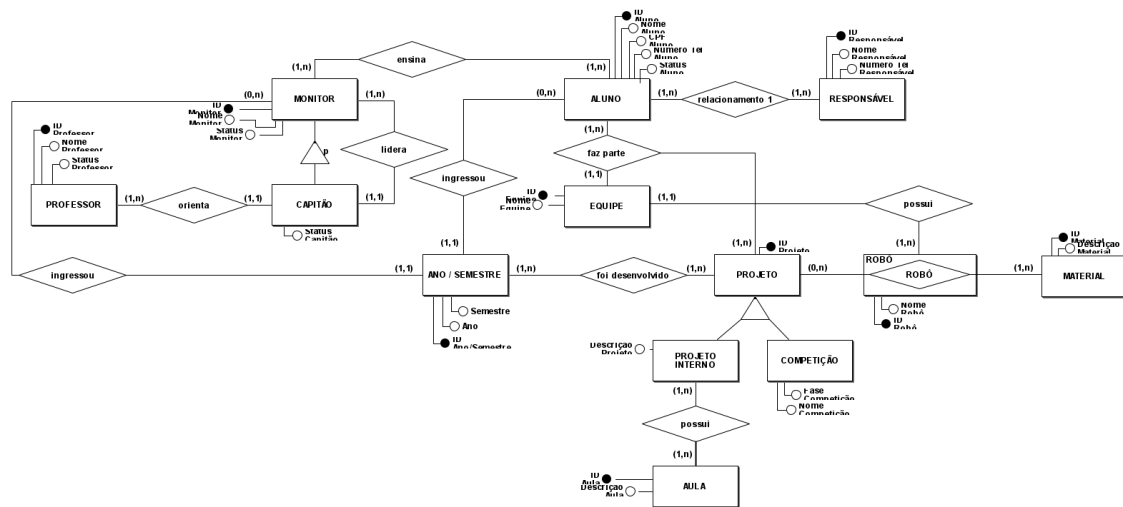
O Sistema de Gerenciamento de Banco de Dados utilizado foi o MariaDB, que foi criado por Michael “Monty” Widenius.

## **2. Descrição do projeto**

O objetivo do projeto é criar um banco de dados para o TOR, que é um projeto de extensão do ICT Unifesp. O projeto tem como missão principal promover conhecimentos de robótica no ensino fundamental público joseense, e portanto tem relação com o ODS 4.

O projeto tem monitores, caracterizados por um código, nome e status de atuação. Um tipo especial de monitor é o capitão, que tem um status de atuação à parte. O capitão lidera os monitores, e é orientado por um professor. Professores são caracterizados por um código, nome e status de atuação. Monitores estão associados ao ano/semestre em que ingressaram no projeto. Monitores ensinam os alunos, caracterizados por um código, nome, CPF, número de telefone e status. Cada aluno tem no mínimo um responsável, que é caracterizado por um código, nome e número de telefone. Alunos estão associados ao ano/semestre em que ingressaram no projeto. Além disso, um aluno faz parte de uma equipe para o desenvolvimento de um projeto. Equipes são caracterizadas por um código e por um nome. Projetos são divididos em duas categorias: projeto interno, caracterizado pela descrição do projeto, e competição, caracterizada pelo nome da competição e a fase. Para cada projeto interno são desenvolvidas uma série de aulas, caracterizadas por um código e por uma descrição da aula. Cada projeto utiliza uma série de materiais, catalogados com um código e uma descrição, para se construir um robô. Cada equipe possui um robô, que é caracterizado por um código e um nome.

### 3. Modelo Entidade Relacionamento



### 4. Modelo Lógico

Professor(IDProf, NomeProf, StatusProf, #IDCapitão)

Monitor(IDMonitor, NomeMonitor, StatusMonitor, #IDCapitão, #IDAnoSemestre)

Capitão(#IDCapitao, StatusCapitao)

Aluno(IDAluno, NomeAluno, CPFAluno, NumeroTelAluno, StatusAluno, #IDAnoSemestre)

Ensina(#IDMonitor, #IDAluno)

Responsável(IDResponsavel, NomeResponsavel, NumeroTelResponsavel)

Responsável-Aluno(#IDAluno, #IDResponsavel)

Equipe(IDEquipe, NomeEquipe)

Projeto(IDProjeto)

Projeto\_Interno(#IDProjeto, DescriçãoProjeto)

Competicao(#IDProjeto, NomeCompetição, FaseCompetição)

Aula(IDAula, #IDProjeto, DescricaoAula)

Aluno\_Equipe\_Projeto(#IDAluno, #IDProjeto, #IDEquipe)

Ano/Semestre(IDAnoSemestre, Ano, Semestre)

Projeto\_AnoSemestre(#IDAnoSemestre, #IDProjeto)

Robo(IDRobo, NomeRobo, #IDEquipe, #IDMaterial, #IDProjeto)

Material(IDMaterial, DescriçãoMaterial)

## 5 Implementação do Banco de Dados

### 5.1 Criação das Tabelas

As tabelas foram criadas utilizando o *script* mostrado abaixo, resultando na figura 1, que mostra as tabelas inseridas no SGBD.

```
create table Ano_Semestre(  
    IDAnoSemestre int not null auto_increment,  
    Ano year not null,  
    Semestre int not null,  
    primary key(IDAnoSemestre)  
)Engine=INNODB;  
  
create table Monitor(  
    IDMonitor int not null auto_increment,  
    NomeMonitor varchar(80) not null,  
    StatusMonitor varchar(10) not null,  
    IDCapitao int,  
    IDAnoSemestre int not null,  
    primary key(IDMonitor),  
    foreign key(IDCapitao) references Monitor(IDMonitor),  
    foreign key(IDAnoSemestre) references Ano_Semestre(IDAnoSemestre)  
)Engine=INNODB;  
  
create table Capitao(  
    IDCapitao int not null,  
    StatusCapitao varchar(10) not null,  
    primary key(IDCapitao),  
    foreign key(IDCapitao) references Monitor(IDMonitor)  
)Engine=INNODB;  
  
create table Professor(  
    IDProfessor int not null auto_increment,  
    NomeProf varchar(80) not null,  
    StatusProf varchar(10) not null,  
    IDCapitao int,  
    foreign key(IDCapitao) references Capitao(IDCapitao),  
    primary key(IDProfessor)  
)Engine=INNODB;  
  
create table Aluno(  
    IDAluno int not null auto_increment,  
    NomeAluno varchar(80) not null,  
    CPFAluno varchar(11) not null,  
    NumeroTelAluno varchar(11) not null,  
    StatusAluno varchar(10) not null,  
    IDAnoSemestre int not null,
```

```
        primary key(IDAluno),
        foreign key(IDAnoSemestre) references Ano_Semestre(IDAnoSemestre)
    )Engine=INNODB;
```

```
create table Ensina(
    IDAluno int not null,
    IDMonitor int not null,
    primary key(IDAluno, IDMonitor),
    foreign key(IDAluno) references Aluno(IDAluno),
    foreign key(IDMonitor) references Monitor(IDMonitor)
)Engine=INNODB;
```

```
create table Responsavel(
    IDResponsavel int not null auto_increment,
    NomeResponsavel varchar(80) not null,
    NumeroTelResponsavel varchar(11) not null,
    primary key(IDResponsavel)
)Engine=INNODB;
```

```
create table Responsavel_Aluno(
    IDAluno int not null,
    IDResponsavel int not null,
    primary key(IDAluno, IDResponsavel),
    foreign key(IDAluno) references Aluno(IDAluno),
    foreign key(IDResponsavel) references Responsavel(IDResponsavel)
)Engine=INNODB;
```

```
create table Equipe(
    IDEquipe int not null auto_increment,
    NomeEquipe varchar(80) not null,
    primary key(IDEquipe)
)Engine=INNODB;
```

```
create table Projeto(
    IDProjeto int not null auto_increment,
    primary key(IDProjeto)
)Engine=INNODB;
```

```
create table Projeto_Interno(
    IDProjeto int not null,
    DescricaoProjeto varchar(80) not null,
    primary key(IDProjeto),
    foreign key(IDProjeto) references Projeto(IDProjeto)
)Engine=INNODB;
```

```
create table Competicao(
    IDProjeto int not null,
    NomeCompeticao varchar(80) not null,
    FaseCompeticao varchar(50) not null,
```

```
        primary key(IDProjeto),
        foreign key(IDProjeto) references Projeto(IDProjeto)
    )Engine=INNODB;
```

```
create table Material(
    IDMaterial int not null auto_increment,
    DescricaoMaterial varchar(80) not null,
    primary key(IDMaterial)
)Engine=INNODB;
```

```
create table Robo(
    IDRobo int not null auto_increment,
    NomeRobo varchar(80) not null,
    IDEquipe int not null,
    IDMaterial int not null,
    IDProjeto int not null,
    primary key(IDRobo, IDMaterial, IDProjeto),
    foreign key(IDEquipe) references Equipe(IDEquipe),
    foreign key(IDMaterial) references Material(IDMaterial),
    foreign key(IDProjeto) references Projeto(IDProjeto)
)Engine=INNODB;
```

```
create table Aula(
    IDAula int not null auto_increment,
    IDProjeto int not null,
    DescricaoAula varchar(80),
    primary key(IDAula, IDProjeto),
    foreign key(IDProjeto) references Projeto_Interno(IDProjeto)
)Engine=INNODB;
```

```
create table Aluno_Equipe_Projeto(
    IDAluno int not null,
    IDEquipe int not null,
    IDProjeto int not null,
    primary key(IDAluno, IDEquipe, IDProjeto),
    foreign key(IDAluno) references Aluno(IDAluno),
    foreign key(IDEquipe) references Equipe(IDEquipe),
    foreign key(IDProjeto) references Projeto(IDProjeto)
)Engine=INNODB;
```

```
create table Projeto_AnoSemestre(
    IDAnoSemestre int not null,
    IDProjeto int not null,
    primary key(IDAnoSemestre, IDProjeto),
    foreign key(IDAnoSemestre) references Ano_Semestre(IDAnoSemestre),
    foreign key(IDProjeto) references Projeto(IDProjeto)
)Engine=INNODB;
```

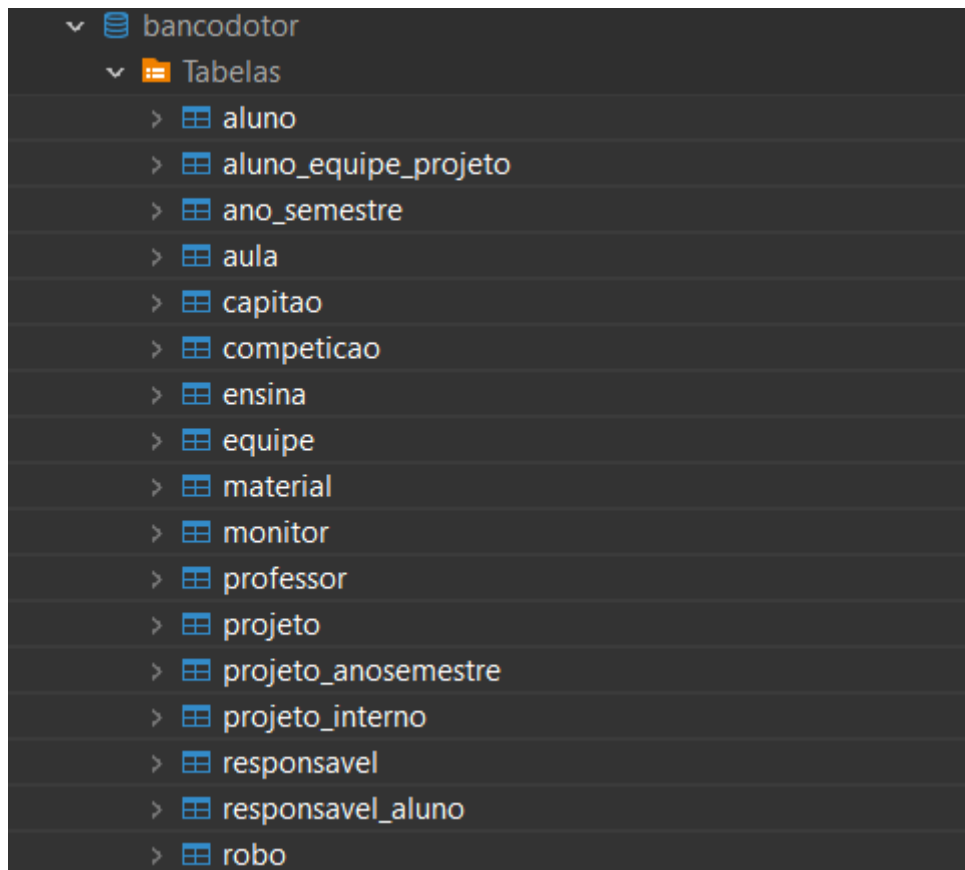


Figura 1: Tabelas adicionadas ao MariaDB.

## 5.2 Inserção dos Dados nas Tabelas

Os dados do banco foram inseridos nas tabelas utilizando os comandos 1 a 17, listados a seguir, resultando nas figuras 2 a 18, que mostram o conteúdo de cada tabela após as inserções.

1. **insert into** Ano\_Semestre(IDAnoSemestre, Ano, Semestre) **values**  
(1, 2022, 1),  
(2, 2022, 2),  
(3, 2023, 1),  
(4, 2023, 2),  
(5, 2024, 1),  
(6, 2024, 2),  
(7, 2025, 1);
2. **insert into** Monitor(IDMonitor, NomeMonitor, StatusMonitor, IDCapitao, IDAnoSemestre) **values**  
(1, 'Viviane', 'Ativo', 1, 4),  
(2, 'Breno', 'Ativo', 1, 4),  
(3, 'Davi', 'Não ativo', 1, 4),  
(4, 'Ícaro', 'Ativo', 1, 5),  
(5, 'Pedro', 'Ativo', 1, 5),  
(6, 'Luana', 'Ativo', 1, 6),



(7, 'Heitor', 'Ativo', 1, 6),  
(8, 'João', 'Não ativo', 1, 6),  
(9, 'Stefani', 'Ativo', 1, 6),  
(10, 'Beatriz', 'Não ativo', 2, 5),  
(11, 'Ana', 'Não ativo', 2, 5),  
(12, 'Luiza', 'Não ativo', 12, 3),  
(13, 'Gabriel', 'Não ativo', 13, 1);

3. **insert into** Capitao(IDCapitao, StatusCapitao) **values**

(1, 'Não ativo'),  
(2, 'Ativo'),  
(12, 'Não ativo'),  
(13, 'Não ativo');

4. **insert into** Professor(IDProfessor, NomeProf, StatusProf, IDCapitao) **values**

(1, 'Fábio', 'Não ativo', 1),  
(2, 'Sérgio', 'Ativo', 2),  
(3, 'Lauro', 'Não ativo', NULL),  
(4, 'Shida', 'Não ativo', NULL);

5. **insert into** Aluno(IDAluno, NomeAluno, CPFAluno, NumeroTelAluno, StatusAluno, IDAnoSemestre) **values**

(1, 'Maria Eduarda', '01010101010', '12901010101', 'Ativo', 1),  
(2, 'Maria Beatriz', '02020202020', '12902020202', 'Ativo', 3),  
(3, 'Davi S', '03030303030', '12903030303', 'Ativo', 1),  
(4, 'Davi G', '04040404040', '12904040404', 'Ativo', 5),  
(5, 'Miguel', '05050505050', '12905050505', 'Ativo', 5),  
(6, 'Miguel G', '06060606060', '12906060606', 'Ativo', 5),  
(7, 'Luís Miguel', '07070707070', '12907070707', 'Ativo', 5),  
(8, 'Kauan', '08080808080', '12908080808', 'Ativo', 3),  
(9, 'Marcelo', '09090909090', '12909090909', 'Ativo', 5),  
(10, 'Nicolas', '10101010101', '12910101010', 'Ativo', 5),  
(11, 'Luan', '11111111111', '12911111111', 'Ativo', 5),  
(12, 'João', '12121212121', '12912121212', 'Ativo', 5),  
(13, 'Maria', '13131313131', '12913131313', 'Não ativo', 5),  
(14, 'Aubrey', '14141414141', '12914141414', 'Não ativo', 1);

6. **insert into** Ensina(IDMonitor, IDAluno) **values**

(4, 1),  
(4, 2),  
(4, 3),  
(1, 1),  
(1, 2),  
(1, 3),  
(2, 1),  
(2, 2),  
(2, 3),  
(5, 6),  
(5, 7),

(10, 5),  
(10, 4),  
(10, 11),  
(12, 14);

7. **insert into** Responsavel(IDResponsavel, NomeResponsavel, NumeroTelResponsavel) **values**

(1, 'José', '12900100100'),  
(2, 'Karla', '12900200200'),  
(3, 'Geraldo', '12900300300'),  
(4, 'Osvaldo', '12900400400'),  
(5, 'Paula', '12900500500'),  
(6, 'Helen', '12900600600'),  
(7, 'Gabriela', '12900700700'),  
(8, 'Thaís', '12900800800'),  
(9, 'Josefina', '12900900900'),  
(10, 'Lara', '12901001001'),  
(11, 'Marcos', '12901101101'),  
(12, 'Carlos', '12901201201'),  
(13, 'Renata', '12901301301'),  
(14, 'Sabrina', '12901401401');

8. **insert into** Responsavel\_Aluno(IDAluno, IDResponsavel) **values**

(1, 7),  
(2, 10),  
(3, 1),  
(4, 14),  
(5, 3),  
(6, 9),  
(7, 11),  
(8, 6),  
(9, 2),  
(10, 12),  
(11, 4),  
(12, 5),  
(13, 8),  
(14, 13);

9. **insert into** Equipe(IEquipe, NomeEquipe) **values**

(1, 'Scuderia Ferrari HP'),  
(2, 'Os Galáticos'),  
(3, 'Autobots'),  
(4, 'Silvertech'),  
(5, 'Equipe');

10. **insert into** Projeto(IDProjeto) **values**

(1),  
(2),  
(3),

(4),  
(5),  
(6),  
(7),  
(8),  
(9);

11. **insert into** Projeto\_Interno(IDProjeto, DescricaoProjeto) **values**

(1, 'Braço robótico de três eixos'),  
(2, 'Aulas introdutórias de Arduino'),  
(8, 'Impressora 3D'),  
(9, 'Garra Robótica');

12. **insert into** Competicao(IDProjeto, NomeCompeticao, FaseCompeticao) **values**

(3, 'OBR 2023', 'Regional'),  
(4, 'OBR 2024', 'Regional'),  
(5, 'OBR 2024', 'Estadual'),  
(6, 'MOTORS 2023', 'Regional'),  
(7, 'MOTORS 2024', 'Estadual');

13. **insert into** Material(IDMaterial, DescricaoMaterial) **values**

(1, 'LEGO'),  
(2, 'Bloco NXT'),  
(3, 'Computador'),  
(4, 'Carregador'),  
(5, 'Cabo'),  
(6, 'Atuador'),  
(7, 'Sensor de cor'),  
(8, 'Sensor ultrassônico'),  
(9, 'Sensor de toque'),  
(10, 'Pista');

14. **insert into** Robo(IDRobo, NomeRobo, IDEquipe, IDMaterial, IDProjeto) **values**

(1, 'Loki', 1, 1, 4),  
(2, 'Loki', 1, 1, 5),  
(3, 'Odin', 2, 2, 2),  
(4, 'Hela', 3, 7, 1);

15. **insert into** Aula(IDAula, IDProjeto, DescricaoAula) **values**

(1, 1, 'Criação da base para o braço'),  
(2, 1, 'Redução e potência'),  
(3, 1, 'Apresentação do braço'),  
(4, 2, 'Introdução à placa'),  
(5, 2, 'Arduino IDE e C++'),  
(6, 2, 'Protoboard e projetos');

16. **insert into** Aluno\_Equipe\_Projeto(IDAluno, IDProjeto, IDEquipe) **values**

(1, 4, 1),  
(2, 4, 1),

(3, 4, 1),  
 (1, 5, 1),  
 (2, 5, 1),  
 (3, 5, 1),  
 (9, 7, 5),  
 (10, 7, 5),  
 (5, 7, 5),  
 (11, 7, 5);

17. **insert into** Projeto\_AnoSemestre(IDAnoSemestre, IDProjeto) **values**

(6, 1),  
 (7, 2),  
 (3, 3),  
 (5, 4),  
 (6, 5),  
 (4, 6),  
 (6, 7);

123 IDAnoSemestre	Ano	123 Semestre
1	2022	1
2	2022	2
3	2023	1
4	2023	2
5	2024	1
6	2024	2
7	2025	1

Figura 2: Tabela Ano\_Semestre.

123 IDMonitor	A-Z NomeMonitor	A-Z StatusMonitor	123 IDCapitao	123 IDAnoSemestre
1	Viviane	Ativo	1	4
2	Breno	Ativo	1	4
3	Davi	Não ativo	1	4
4	Ícaro	Ativo	1	5
5	Pedro	Ativo	1	5
6	Luana	Ativo	1	6
7	Heitor	Ativo	1	6
8	João	Não ativo	1	6
9	Stefani	Ativo	1	6
10	Beatriz	Não ativo	2	5
11	Ana	Não ativo	2	5
12	Luíza	Não ativo	12	3
13	Gabriel	Não ativo	13	1

Figura 3: Tabela Monitor.

123 IDCapitao	A-Z StatusCapitao
1	Não ativo
2	Ativo
12	Não ativo
13	Não ativo

Figura 4: Tabela Capitao.

123 IDProfessor	A-Z NomeProf	A-Z StatusProf	123 IDCapitao
1	Fábio	Não ativo	1
2	Sérgio	Ativo	2
3	Lauro	Não ativo	[NULL]
4	Shida	Não ativo	[NULL]

Figura 5: Tabela Professor.

123 IDAluno	A-Z NomeAluno	A-Z CPFAluno	A-Z NumeroTelAluno	A-Z StatusAluno	123 IDAnoSemestre
1	Maria Eduarda	01010101010	12901010101	Ativo	1
2	Maria Beatriz	02020202020	12902020202	Ativo	3
3	Davi S	03030303030	12903030303	Ativo	1
4	Davi G	04040404040	12904040404	Ativo	5
5	Miguel	05050505050	12905050505	Ativo	5
6	Miguel G	06060606060	12906060606	Ativo	5
7	Luís Miguel	07070707070	12907070707	Ativo	5
8	Kauan	08080808080	12908080808	Ativo	3
9	Marcelo	09090909090	12909090909	Ativo	5
10	Nicolas	10101010101	12910101010	Ativo	5
11	Luan	11111111111	12911111111	Ativo	5
12	João	12121212121	12912121212	Ativo	5
13	Maria	13131313131	12913131313	Não ativo	5
14	Aubrey	14141414141	12914141414	Não ativo	1

Figura 6: Tabela Aluno.

123 IDAluno	123 IDMonitor
1	1
2	1
3	1
1	2
2	2
3	2
1	4
2	4
3	4
6	5
7	5
4	10
5	10
11	10
14	12

Figura 7: Tabela Ensina.

123 IDResponsavel	A-Z NomeResponsavel	A-Z NumeroTelResponsavel
1	José	12900100100
2	Karla	12900200200
3	Geraldo	12900300300
4	Osvaldo	12900400400
5	Paula	12900500500
6	Helen	12900600600
7	Gabriela	12900700700
8	Thaís	12900800800
9	Josefina	12900900900
10	Lara	12901001001
11	Marcos	12901101101
12	Carlos	12901201201
13	Renata	12901301301
14	Sabrina	12901401401

Figura 8: Tabela Responsavel.

123 IDAluno	123 IDResponsavel
3	1
9	2
5	3
11	4
12	5
8	6
1	7
13	8
6	9
2	10
7	11
10	12
14	13
4	14

Figura 9: Tabela Responsavel\_Aluno.

123 IDEquipe	A-Z NomeEquipe
1	Scuderia Ferrari HP
2	Os Galáticos
3	Autobots
4	Silvertech
5	Equipe

Figura 10: Tabela Equipe.

123 IDProjeto
1
2
3
4
5
6
7
8
9

Figura 11: Tabela Projeto.

123 IDProjeto	A-Z DescricaoProjeto
1	Braço robótico de três eixos
2	Aulas introdutórias de Arduino
8	Impressora 3D
9	Garra Robótica

Figura 12: Tabela Projeto\_Interno.

123 IDProjeto	A-Z NomeCompeticao	A-Z FaseCompeticao
3	OBR 2023	Regional
4	OBR 2024	Regional
5	OBR 2024	Estadual
6	MOTORS 2023	Regional
7	MOTORS 2024	Estadual

Figura 13: Tabela Competicao.

123 IDMaterial	A-Z DescricaoMaterial
1	LEGO
2	Bloco NXT
3	Computador
4	Carregador
5	Cabo
6	Atuador
7	Sensor de cor
8	Sensor ultrassônico
9	Sensor de toque
10	Pista

Figura 14: Tabela Material.

123 IDRobo	A-Z NomeRobo	123 IDEquipe	123 IDMaterial	123 IDProjeto
1	Loki	1	1	4
2	Loki	1	1	5
3	Odin	2	2	2
4	Hela	3	7	1

Figura 15: Tabela Robo.



123 IDAula	123 IDProjeto	A-Z DescricaoAula
1	1	Criação da base para o braço
2	1	Redução e potência
3	1	Apresentação do braço
4	2	Introdução à placa
5	2	Arduino IDE e C++
6	2	Protoboard e projetos

Figura 16: Tabela Aula.

123 IDAluno	123 IDEquipe	123 IDProjeto
1	1	4
1	1	5
2	1	4
2	1	5
3	1	4
3	1	5
5	5	7
9	5	7
10	5	7
11	5	7

Figura 17: Tabela Aluno\_Equipe\_Projeto.

123 IDAnoSemestre	123 IDProjeto
6	1
7	2
3	3
5	4
6	5
4	6
6	7

Figura 18: Tabela Projeto\_AnoSemestre

## 6 Desenvolvimento de Consultas SQL

### 6.1 Consultas Básicas

Os comandos de consultas básicas estão listados a seguir de 1 a 5, juntamente com os seus respectivos enunciados e seus resultados. As figuras 19 a 23 representam o resultado de cada consulta básica.

1. **Selecione o nome dos monitores que são ou já foram capitães do TOR, indicando o status atual deles.**

```
SELECT m.NomeMonitor, c.StatusCapitao  
FROM Monitor m, Capitao c  
WHERE m.IDMonitor = c.IDCapitao;
```

A-Z NomeMonitor ▼	A-Z StatusCapitao ▼
Viviane	Não ativo
Breno	Ativo
Luiza	Não ativo
Gabriel	Não ativo

Figura 19: Resultado da consulta básica 1.

2. **Selecione os alunos que participaram do TOR no semestre de 2024/1, indicando o nome do aluno e o seu status.**

```
SELECT a.NomeAluno, a.StatusAluno  
FROM Aluno a JOIN Ano_Semestre ans ON (a.IDAnoSemestre =  
ans.IDAnoSemestre AND ans.Ano = 2024 AND ans.Semestre = 1);
```

A-Z NomeAluno ▼	A-Z StatusAluno ▼
Davi G	Ativo
Miguel	Ativo
Miguel G	Ativo
Luís Miguel	Ativo
Marcelo	Ativo
Nicolas	Ativo
Luan	Ativo
João	Ativo
Maria	Não ativo

Figura 20: Resultado da consulta básica 2.

3. **Retorne o nome do robô, o nome da equipe que o construiu e o nome da competição em que ele participou.**

```
SELECT r.NomeRobo, e.NomeEquipe, c.NomeCompeticao
FROM Robo r JOIN Equipe e ON (r.IDEquipe = e.IDEquipe) JOIN Competicao c ON
(r.IDProjeto = c.IDProjeto);
```

A-Z NomeRobo ▼	A-Z NomeEquipe ▼	A-Z NomeCompeticao ▼
Loki	Scuderia Ferrari HP	OBR 2024
Loki	Scuderia Ferrari HP	OBR 2024

Figura 21: Resultado da consulta básica 3.

4. **Retorne o nome dos alunos e o CPF dos alunos, nos quais os responsáveis tenham um número de telefone que termine com 1.**

```
SELECT a.NomeAluno, a.CPFAluno
FROM Aluno a JOIN Responsavel_Aluno ra ON (a.IDAluno = ra.IDAluno) JOIN
Responsavel r ON (ra.IDResponsavel = r.IDResponsavel)
WHERE r.NumeroTelResponsavel LIKE '%1';
```

A-Z NomeAluno ▼	A-Z CPFAluno ▼
Maria Beatriz	02020202020
Luís Miguel	07070707070
Nicolas	10101010101
Aubrey	14141414141
Davi G	04040404040

Figura 22: Resultado da consulta básica 4.

5. **Indique o nome dos capitães, seguido de seus status, e o nome de seus professores orientadores, com seus status.**

```
SELECT m.NomeMonitor, c.StatusCapitao, p.NomeProf, p.StatusProf
FROM Capitao c, Professor p, Monitor m
WHERE c.IDCapitao = m.IDMonitor AND p.IDCapitao = c.IDCapitao;
```

A-Z NomeMonitor ▼	A-Z StatusCapitao ▼	A-Z NomeProf ▼	A-Z StatusProf ▼
Viviane	Não ativo	Fábio	Não ativo
Breno	Ativo	Sérgio	Ativo

Figura 23: Resultado da consulta básica 5.

## 6.2 Consultas com LEFT JOIN

Os dois comandos de LEFT JOIN, seus enunciados e seus resultados estão listados a seguir. As figuras 24 e 25 representam os resultados das consultas com LEFT JOIN.

1. **Mostrar o nome dos robôs que participam de competições, indicando o nome da equipe e o nome da competição. Robôs que não participaram de competições devem aparecer.**

```
SELECT r.NomeRobo, e.NomeEquipe, c.NomeCompeticao
FROM Robo r LEFT JOIN Equipe e ON (r.IDEquipe = e.IDEquipe) LEFT JOIN
Competicao c ON (r.IDProjeto = c.IDProjeto);
```

A-Z NomeRobo	A-Z NomeEquipe	A-Z NomeCompeticao
Loki	Scuderia Ferrari HP	OBR 2024
Loki	Scuderia Ferrari HP	OBR 2024
Odin	Os Galáticos	[NULL]
Hela	Autobots	[NULL]

Figura 24: Resultado da consulta 1 com LEFT JOIN.

2. **Retorne o ID e a descrição dos projetos internos que são aulas, mostrando também a descrição da aula e o ID da aula. Projetos internos que não são aulas devem aparecer.**

```
SELECT pi.IDProjeto, pi.DescricaoProjeto, a.DescricaoAula, a.IDAula
FROM Projeto_Interno pi LEFT JOIN Aula a ON (pi.IDProjeto = a.IDProjeto);
```

123 IDProjeto	A-Z DescricaoProjeto	A-Z DescricaoAula	123 IDAula
1	Braço robótico de três eixos	Criação da base para o braço	1
1	Braço robótico de três eixos	Redução e potência	2
1	Braço robótico de três eixos	Apresentação do braço	3
2	Aulas introdutórias de Arduino	Introdução à placa	4
2	Aulas introdutórias de Arduino	Arduino IDE e C++	5
2	Aulas introdutórias de Arduino	Protoboard e projetos	6
8	Impressora 3D	[NULL]	[NULL]
9	Garra Robótica	[NULL]	[NULL]

Figura 25: Resultado da consulta 2 com LEFT JOIN.

## 6.3 Consultas Agregadas

Os comandos de Consultas Agregadas, seus respectivos enunciados e seus resultados estão listados a seguir de 1 a 5. As figuras 26 a 30 representam os resultados das consultas agregadas.

1. **Obter o nome dos capitães na ordem crescente por nome e o número de monitores que cada capitão liderou, apenas para os que lideraram mais de 1 aluno.**

```
SELECT c.NomeMonitor, COUNT(m.IDMonitor)
FROM Monitor m JOIN Monitor c ON (m.IDCapitao = c.IDMonitor)
GROUP BY c.NomeMonitor
HAVING COUNT(m.IDMonitor) > 1
ORDER BY c.NomeMonitor ASC;
```

A-Z NomeMonitor ▼	123 COUNT(m.IDMonitor) ▼
Breno	2
Viviane	9

Figura 26: Resultado da consulta agregada 1.

2. **Retorne o nome das equipes que têm mais de 3 alunos, ordenando os nomes das equipes de forma decrescente.**

```
SELECT e.NomeEquipe
FROM Equipe e, Aluno_Equipe_Projeto aep
WHERE e.IDEquipe = aep.IDEquipe
GROUP BY e.NomeEquipe
HAVING COUNT(aep.IDAluno) > 4
ORDER BY e.NomeEquipe DESC;
```

A-Z NomeEquipe ▼
Scuderia Ferrari HP

Figura 27: Resultado da consulta agregada 2.

3. **Selecione o nome dos monitores que ensinaram mais de 2 alunos, ordenando seus nomes por ordem crescente.**

```
SELECT m.NomeMonitor
FROM Monitor m JOIN Ensina e ON (m.IDMonitor = e.IDMonitor)
GROUP BY m.NomeMonitor
HAVING COUNT(e.IDAluno) > 2
ORDER BY m.NomeMonitor ASC;
```

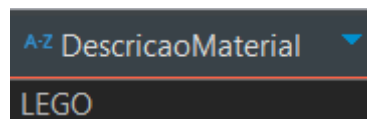
A-Z NomeMonitor ▼
Beatriz
Breno
Ícaro
Viviane

Figura 28: Resultado da consulta agregada 3.

4. **Indique a descrição do material que foi utilizado em mais de 1 robô, ordenando as descrições por ordem decrescente.**

```
SELECT m.DescricaoMaterial
FROM Material m, Robo r
WHERE m.IDMaterial = r.IDMaterial
GROUP BY m.DescricaoMaterial
HAVING COUNT(r.IDRobo) > 1
```

ORDER BY m.DescricaoMaterial DESC;

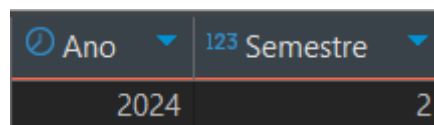


A-Z DescricaoMaterial
LEGO

Figura 29: Resultado da consulta agregada 4.

5. Retorne o ano e semestre em que foram feitos mais de 2 projetos, ordenando os anos e semestres de forma crescente.

```
SELECT a.Ano, a.Semestre
FROM Ano_Semestre a JOIN Projeto_AnoSemestre p ON (a.IDAnoSemestre =
p.IDAnoSemestre)
GROUP BY p.IDAnoSemestre
HAVING COUNT(p.IDProjeto) > 2
ORDER BY a.Ano, a.Semestre ASC;
```



Ano	Semestre
2024	2

Figura 30: Resultado da consulta agregada 5.

## 7 Desenvolvimento da Interface Gráfica

Para o desenvolvimento da interface gráfica, foram utilizados algumas ferramentas muito utilizadas no cenário de Desenvolvimento Web, foram elas:

- Node.js: Ambiente de execução de JavaScript.
- TypeScript: Para execução junto com o React, o que permitiu uma melhor manipulação das páginas e dos estados.
- NPM: Gerenciador de pacotes para Node.js.
- Biblioteca React: Criada pelo Facebook e comumente utilizada para criação de UIs (User Interfaces).
- Framework Next.js: Para proporcionar um desenvolvimento de rápida prototipação, fácil manipulação de caminhos e rotas além de possuir diversas otimizações e implementações QoL que facilitam o desenvolvimento de páginas.
- Bibliotecas MySQL e MySQL2: Embora focadas para a integração com Bancos MySQL, serviram muitíssimo bem para uma integração com um Banco MariaDB, como o utilizado neste projeto.
- Bibliotecas HeroUI e Tailwind CSS: Utilizadas para produzir e padronizar o visual da interface.

A interface possui uma simples apresentação do TOR e a exibição de alguns dados que são obtidos no Banco de Dados e transmitidos para a interface por meio de um sistema de requisição de APIs dentro do próprio projeto.

Por exemplo, essa é a forma cujo a qual são obtidos os dados da quantidade de alunos ativos no TOR:

```
1 import mysql from 'mysql2/promise';
2
3 let connection;
4
5 export const createConnection = async () => {
6   if (!connection || connection.connection._closing || connection.connection.state === 'disconnected') {
7     connection = await mysql.createConnection({
8       host: process.env.DB_HOST,
9       port: process.env.DB_PORT,
10      user: process.env.DB_USER,
11      password: process.env.DB_PASSWORD,
12      database: process.env.DB_DATABASE
13    });
14   }
15   return connection;
16 };
17
```

Figura 31: Código para a criação de conexão com o Banco de Dados, utiliza-se um arquivo .env para a passagem das variáveis secretas de ambiente.

```
1 import { createConnection } from "../../lib/db";
2 import { NextResponse } from "next/server";
3
4 export async function GET(req) {
5   try {
6     const db = await createConnection();
7     if (!db) throw new Error("Falha ao conectar ao banco de dados");
8
9     const sql = 'SELECT COUNT(*) AS count FROM aluno WHERE StatusAluno = "Ativo"';
10    const [rows] = await db.query(sql);
11
12    if (!rows || rows.length === 0) throw new Error("Nenhum resultado encontrado");
13
14    return NextResponse.json({ count: rows[0].count });
15  } catch (error) {
16    console.error("Erro na API /AlunosAtivos:", error);
17    return NextResponse.json({ error: error.message }, { status: 500 });
18  }
19 }
20
```

Figura 32: Código da API de obtenção de números de alunos ativos



Figura 33: Interface gráfica com informações obtidas do banco de dados.

A obtenção de dados para a tabela de monitores ativos consistiu em um select entre três tabelas: Monitor, Ano\_Semestre e Capitao, para isso foram utilizados dois joins de forma a criar a ligação entre essas tabelas por meio de suas chaves estrangeiras:

```
select m.IDMonitor, m.NomeMonitor, anosem.Ano, anosem.Semestre,
c.StatusCapitao from monitor m join ano_semestre anosem on (m.IDAnoSemestre =
anosem.IDAnoSemestre) left join capitao c on (m.IDMonitor = c.IDCapitao) where
m.StatusMonitor = "Ativo"
```

Já para a tabela de Ex-monitores houve somente uma relação entre as tabelas Monitor e Ano\_Semestre como possível ver abaixo:

```
SELECT m.IDMonitor, m.NomeMonitor, anosem.Ano, anosem.Semestre FROM
monitor m JOIN ano_semestre anosem ON (m.IDAnoSemestre = anosem.IDAnoSemestre)
WHERE m.StatusMonitor = "Não Ativo"
```



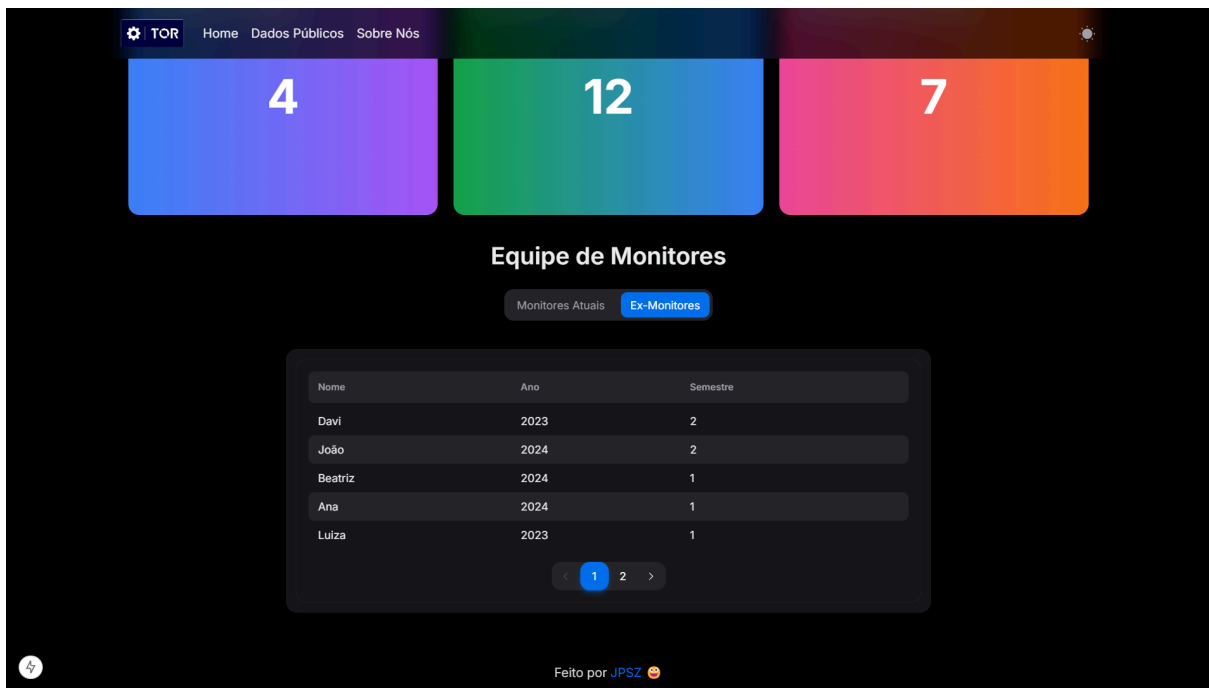


Figura 34: Interface gráfica com a 2ª tabela, de ex-monitores, implementada.

Há ainda uma terceira tabela sendo requisitada pela página, uma tabela que exhibe todos os dados da tabela Robô do Banco de Dados:

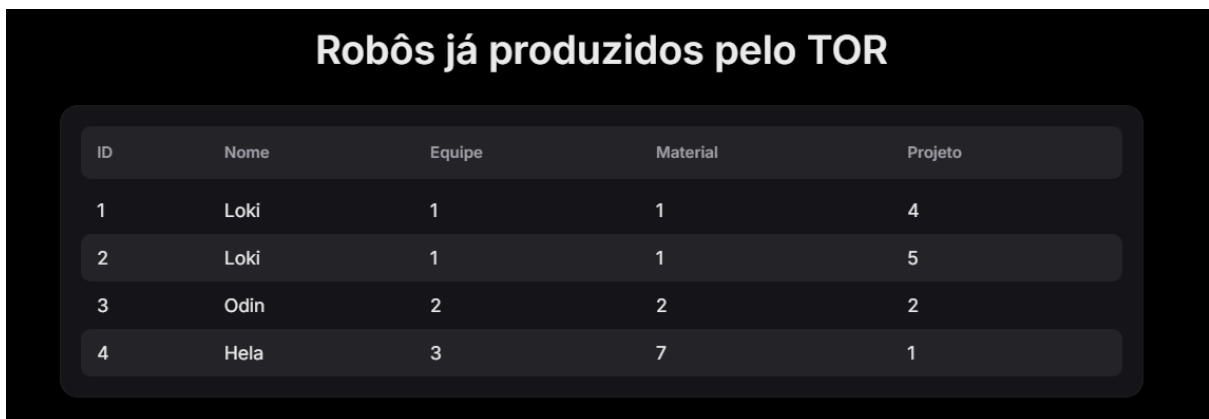


Figura 35: Interface Gráfica com a 3ª tabela, exibindo os robôs produzidos pelo TOR

A seguir, o código da rota que retorna todos os dados dos robôs com a query de select dentro dela:

```
1 import { createConnection } from "../../../../../lib/db";
2 import { NextResponse } from "next/server";
3
4 export async function GET(req) {
5   let db;
6   try {
7     db = await createConnection();
8     const sql = 'SELECT * from robo join ';
9     const [rows] = await db.query(sql);
10
11     if (!rows || rows.length === 0) throw new Error("Nenhum resultado encontrado");
12
13     return NextResponse.json(rows);
14   } catch (error) {
15     console.error("Erro na API /Robos:", error);
16     return NextResponse.json({ error: error.message }, { status: 500 });
17   } finally {
18     if (db) {
19       await db.end();
20     }
21   }
22 }
```

Figura 36: Código de implementação da API que obtém a tabela Robô