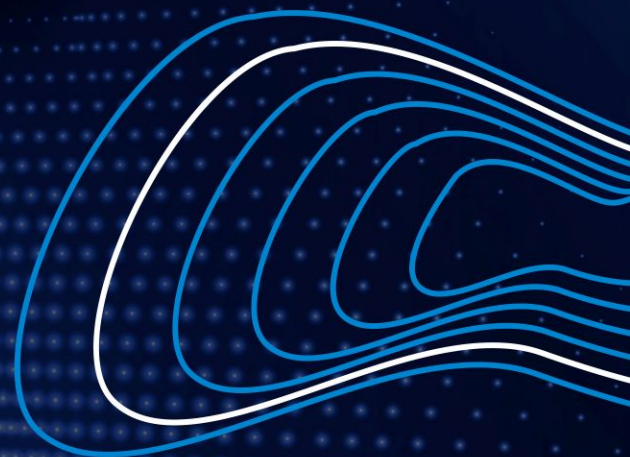


**bsoft** *internetworks*





# Mini Curso - Banco de Dados

Bruno Silva

Marcelo Luis Chociay

Bsoft

Bruno Silva  
Imbituva - PR

Programador Back-End  
Bsoft Docs

Engenharia de  
Software(Unicesumar)

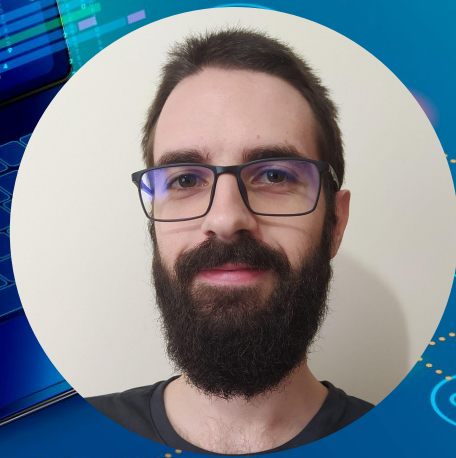




Marcelo Luis Chociay  
Irati - PR

Programador Full-Stack  
Tech Lead Bsoft Docs

Análise e Desenvolvimento de  
Sistemas (IFPR)



# Tópicos

- O que é um Banco de Dados
- Exemplos de Bancos de Dados
- Arquitetura Básica de Sistemas e Bancos de Dados
- O que é SQL
- Principais Comandos SQL
- Functions, Procedures e Triggers
- Desafio: Problema a ser resolvido
- Modelo Conceitual
- Modelo Lógico
- Criação do Banco de Dados (modelo físico)
- Dinâmica online



[t.ly/ANWp](https://t.ly/ANWp) (Github, não é vírus :D)

[sqliteonline.com](https://sqliteonline.com)

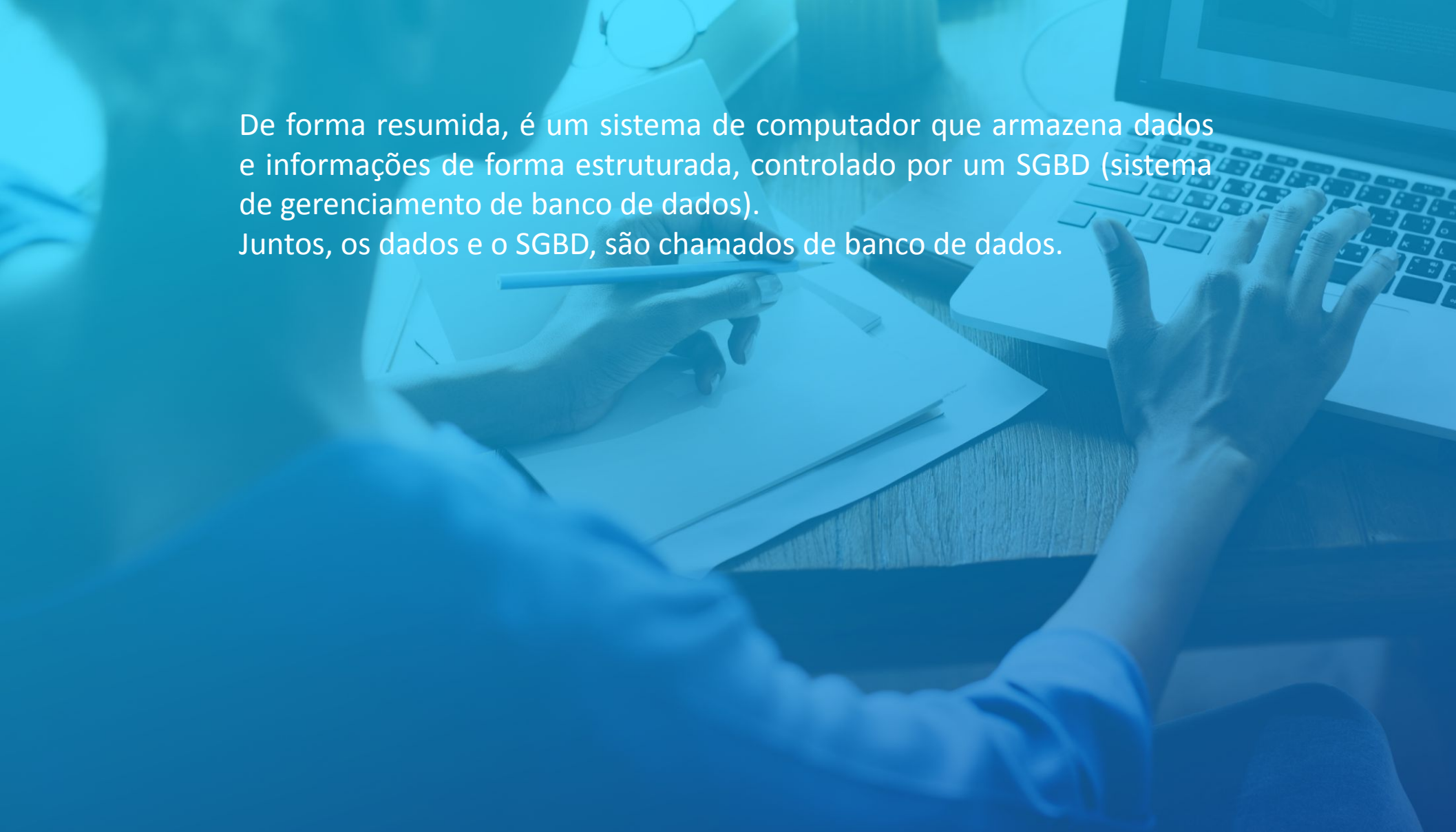
[app.edupulses.io](https://app.edupulses.io)

# Banco de Dados

O que é e qual é sua funcionalidade?





A person is shown from behind, sitting at a desk. They are holding a pen over a piece of paper. A laptop is open to their right. The entire image is covered with a semi-transparent blue filter. Overlaid on the left side of the image is a block of text in white.

De forma resumida, é um sistema de computador que armazena dados e informações de forma estruturada, controlado por um SGBD (sistema de gerenciamento de banco de dados). Juntos, os dados e o SGBD, são chamados de banco de dados.



A person is sitting at a desk, working on a laptop. Their hands are visible, one holding a pen over a piece of paper. The scene is overlaid with a semi-transparent blue filter. Text is overlaid on the left side of the image.

Nos bancos de dados relacionais, os dados são armazenados em colunas e tabelas.

A manipulação dos dados é realizada através da SQL (Linguagem de Consulta Estruturada)

Podemos dizer que uma tabela em um banco de dados se parece com uma planilha no Excel.

# Tipos de Bancos de Dados

Quais são os principais tipos de bancos de dados, e com qual vamos trabalhar hoje?





# Arquitetura Básica de Sistemas e Bancos de Dados

Onde um banco de dados se encaixa em  
uma aplicação real?

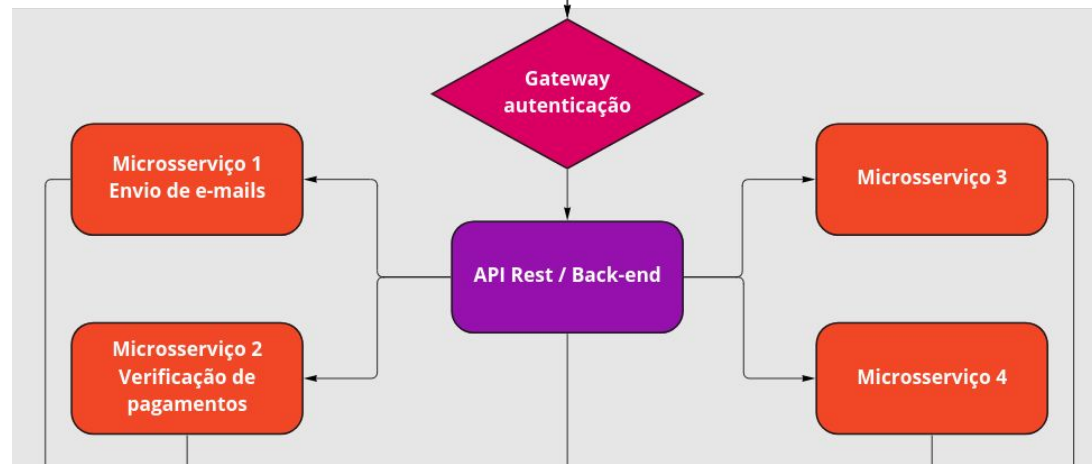




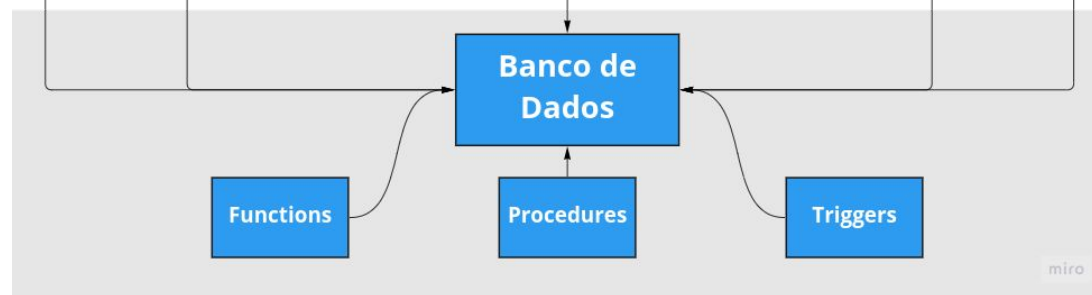
Camada Front-end  
Interface do usuário



Camada Back-end  
Regras de negócio  
etc



Camada Database  
Dados

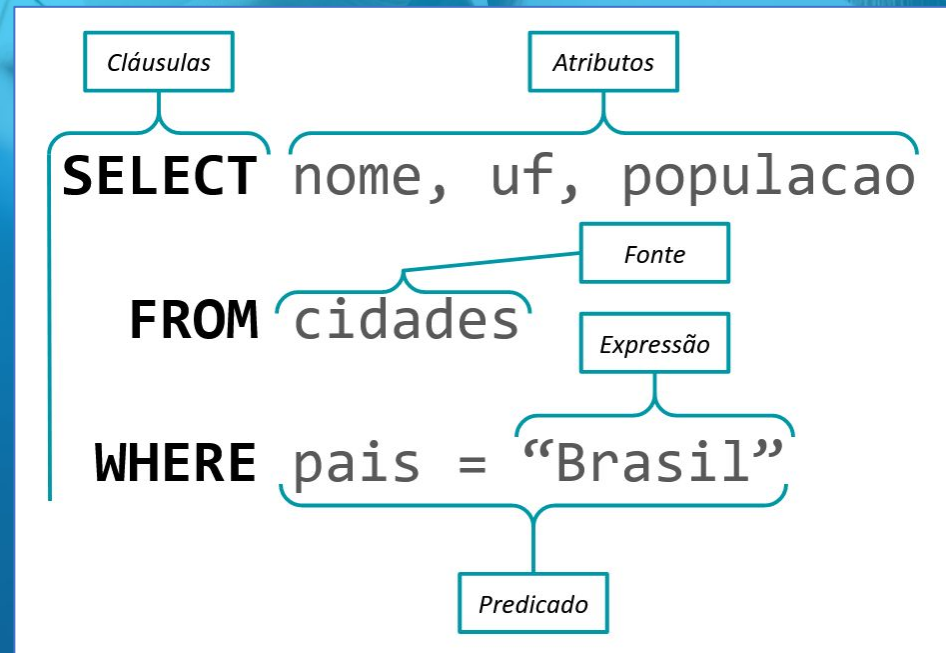


# Structured Query Language

O que é SQL?



A Linguagem de Consulta Estruturada (Structured Query Language) é uma linguagem de programação usada pela maioria dos bancos de dados relacionais para consultar, manipular (inserir, alterar, excluir, etc), definir dados e fornecer controle de acesso.



# Linguagem SQL

Quais os principais comandos?





# Subconjuntos SQL

## DQL

SELECT

## DML

INSERT  
UPDATE  
DELETE

## DDL

CREATE  
ALTER  
DROP

## DCL

GRANT  
REVOKE

## DTL

BEGIN  
COMMIT  
ROLLBACK

# Subconjuntos SQL

## DQL

SELECT

Linguagem de  
Consulta de  
Dados



## DML

INSERT  
UPDATE  
DELETE

Linguagem de  
Manipulação  
de Dados

## DDL

CREATE  
ALTER  
DROP

Linguagem de  
Definição de  
Dados

## DCL

GRANT  
REVOKE

Linguagem de  
Controle de  
Dados

## DTL

BEGIN  
COMMIT  
ROLLBACK

Linguagem de  
Transação de  
Dados

# CREATE

- Permite criar um novo banco de dados, uma nova tabela ou criar índices

Exemplo de uso:

```
CREATE DATABASE 'MeuBD'
```

```
CREATE TABLE funcionarios(  
    nome VARCHAR(100),  
    idade VARCHAR(3),  
    salario FLOAT  
)
```

# INSERT

- Permite criar novos registros

Exemplo de uso:

```
INSERT INTO funcionarios (nome, idade, salario) VALUES ('Marcos Silva','25',1890.50)
```

```
INSERT INTO funcionarios VALUES ('Marcos Silva','25',1890.50)
```



# SELECT

- Possibilita a consulta de dados de uma tabela, função ou procedure
- Pode ser estruturado permitindo consultas simples ou mais complexas

Exemplo de uso:

```
SELECT nome, sobrenome, idade  
FROM clientes;
```

# SELECT

- Possibilita a consulta de dados de uma tabela, função ou procedure
- Pode ser estruturado permitindo consultas simples ou mais complexas

Exemplo de uso:

```
SELECT nome, sobrenome, idade  
FROM clientes;
```

```
SELECT nome, sobrenome, idade  
FROM clientes  
WHERE id_cidade IN (SELECT id  
                     FROM cidades  
                     WHERE uf = 'PR');
```

# UPDATE

- Permite atualizar valores de registros existentes

Exemplo de uso:

```
UPDATE funcionarios  
  SET idade = '25',  
      salario = 2150.50  
 WHERE nome = 'Marcos Silva'
```

# DELETE

- Permite eliminar registros de uma tabela

Exemplo de uso:

```
DELETE FROM funcionarios  
WHERE nome = 'Marcos Silva'
```



# ALTER

- Efetua alterações na estrutura de uma tabela

Exemplo de uso:

```
ALTER TABLE funcionarios ADD data_nascimento DATE;
```

```
ALTER TABLE funcionarios DROP idade;
```

```
ALTER TABLE funcionarios MODIFY nome VARCHAR(150);
```

# Cláusulas

Permite criar condições para selecionar ou modificar uma consulta

## **FROM**

Especifica a tabela que será consultada ou sofrerá modificação

## **WHERE**

Indica as condições que serão usadas para localizar os registros selecionados

## **GROUP BY**

Separa os registros em grupos específicos

## **ORDER BY**

Ordena os registros selecionados

## **LIMIT**

Limita a quantidade de registros selecionados no resultado

## **COUNT**

Conta quantos registros foram selecionados

# JOINS

JOINS são uma maneira de combinar registros e colunas de uma ou mais tabelas, usando valores comuns de cada uma delas, através da junção das colunas determinadas no JOIN.

## **INNER JOIN**

Retorna todos os registros em comum de ambas as tabelas que satisfazem a condição de junção

## **LEFT JOIN**

Retorna todos os registros da tabela da esquerda e os registros correspondentes da tabela da direita

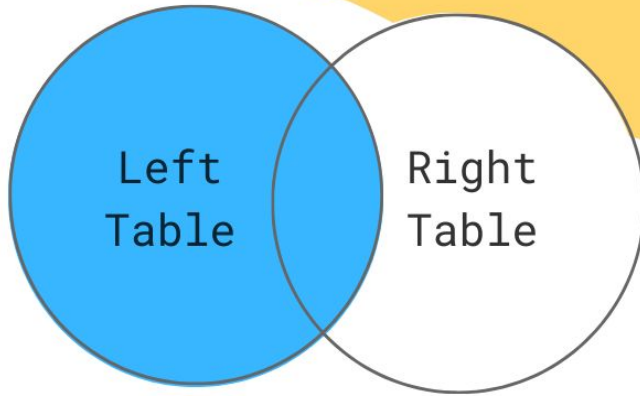
## **RIGHT JOIN**

Retorna todos os registros da tabela da direita e os registros correspondentes da tabela da esquerda

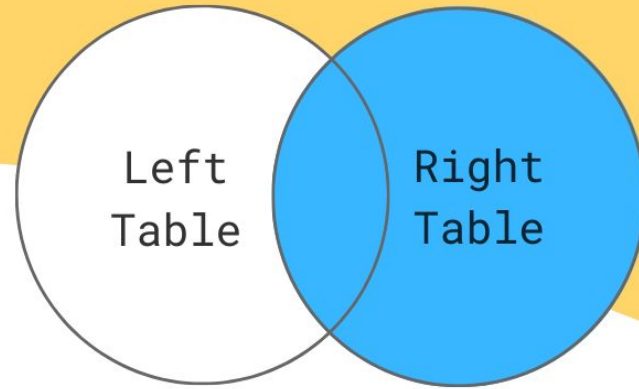
## **FULL JOIN**

Retorna todos os registros quando houver uma correspondência na tabela da esquerda ou da direita

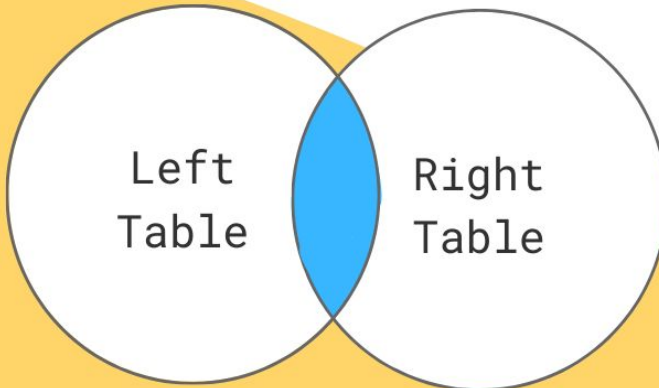
## LEFT JOIN



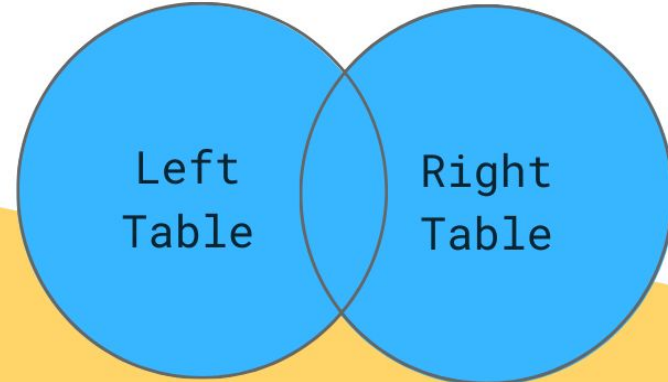
## RIGHT JOIN



## INNER JOIN



## FULL JOIN





# PRIMARY KEY e FOREIGN KEY

Definição da chave primária e  
chave(s) estrangeira(s)

## **PRIMARY KEY (chave primária)**

Define a coluna principal (identificador único) de uma tabela. A coluna definida como PRIMARY KEY não pode ter valores repetidos. Cada tabela só pode ter uma PK

## **FOREIGN KEY (chave estrangeira)**

Define uma coluna que possui um relacionamento com outra tabela, ou seja, é uma referência para a chave primária de outra tabela. Cada tabela pode ter várias FK's

# Functions, Procedures e Triggers



# Funções e Procedimentos

São rotinas que podem executar diversas funções, receber dados de entrada e retornar algum valor

```
CREATE FUNCTION somar_valores (valora DECIMAL(10,2), valorb INT)
RETURNS DECIMAL(10,2)
RETURN valora + valorb;
```

```
SELECT somar_valores(10.5, 5)
```

```
! somar_valores(10.5, 5)
```

```
15.50
```

```
CREATE PROCEDURE contar_funcionarios (OUT parametro1 INT)
BEGIN
    SELECT COUNT(*) INTO parametro1 FROM funcionarios;
END
```

```
CALL contar_funcionarios(@quantidade);
```

```
SELECT @quantidade;
```

# Triggers

São procedimentos armazenados no banco que podem ser ativados automaticamente após uma ação ou processamento

## **AFTER {UPDATE | INSERT | DELETE}**

Determina que o gatilho deve ser acionado depois do UPDATE, INSERT ou DELETE

## **BEFORE {UPDATE | INSERT | DELETE}**

Determina que o gatilho deve ser acionado antes do UPDATE, INSERT ou DELETE

```
CREATE TRIGGER calcular_salario  
AFTER INSERT ON funcionarios  
BEGIN  
...Código...  
END;
```



# Desafio

Está pronto para criar um banco de dados?



A person is shown from the side, sitting at a desk. They are wearing a blue long-sleeved shirt. Their right hand is on a laptop keyboard, and their left hand is holding a pen over a piece of paper. There are several papers and a pair of glasses on the desk. The entire image is covered with a semi-transparent blue filter, and white text is overlaid on the left side.

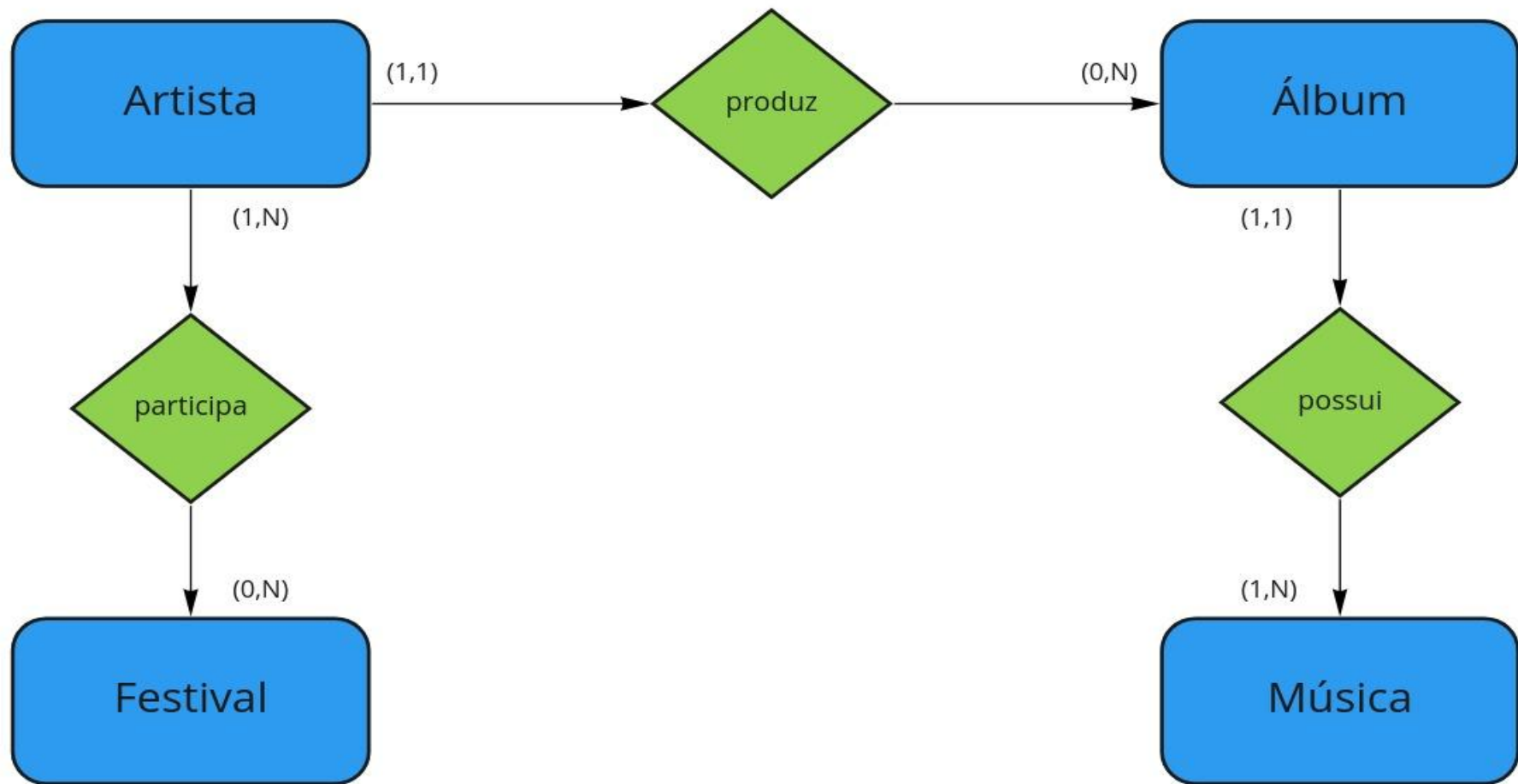
Suponha que você foi contratado para criar um banco de dados de um sistema para controlar registros de artistas, assim como seus álbuns e músicas e os festivais que os artistas participaram.

Quais entidades, relacionamentos e regras você consegue pensar para resolver este problema?

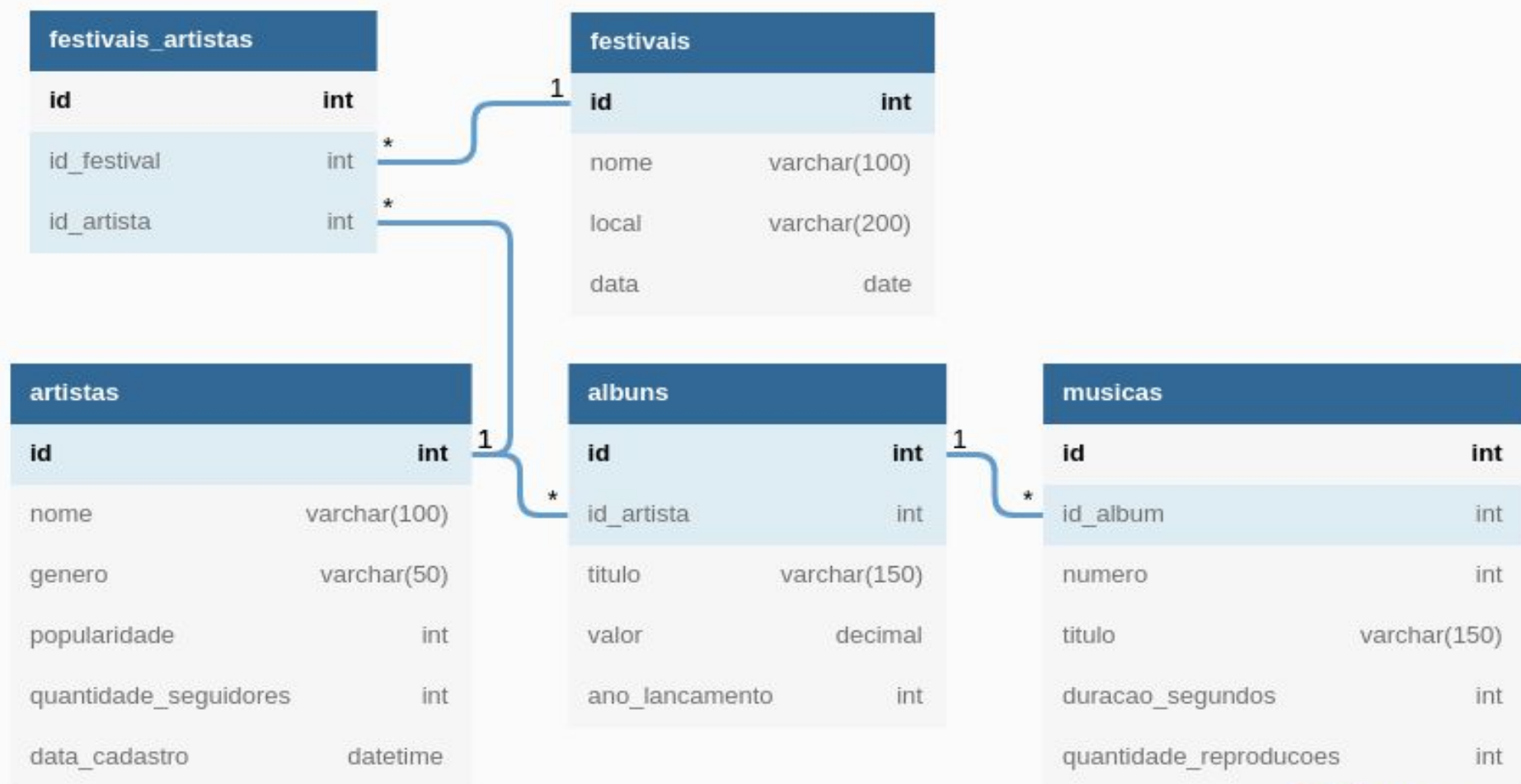


## Regras do BD:

- Um artista pode possuir nenhum ou vários álbuns;
- Um álbum deve pertencer a somente um artista;
- Um álbum deve conter uma ou várias músicas;
- Uma música deve pertencer a somente um álbum;
- Um artista pode participar de nenhum ou vários festivais;
- Um festival deve conter um ou vários artistas;







Dúvidas?



[app.edupulses.io](https://app.edupulses.io)



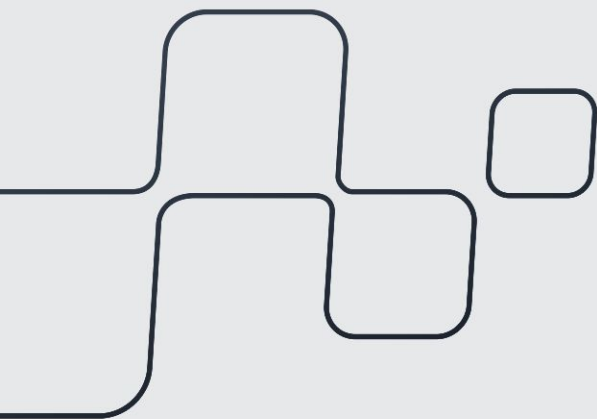
# Contato

[bruno.silva@bsoft.email](mailto:bruno.silva@bsoft.email)  
[linkedin.com/in/BrunoLuizSilva](https://linkedin.com/in/BrunoLuizSilva)  
[github.com/Margera](https://github.com/Margera)

[marcelo.chociay@bsoft.com.br](mailto:marcelo.chociay@bsoft.com.br)  
[linkedin.com/in/marcelo-luis-chociay](https://linkedin.com/in/marcelo-luis-chociay)  
[github.com/MarceloLuis1997](https://github.com/MarceloLuis1997)







Onde tem transportadora,  
**TEM BSOFT**

