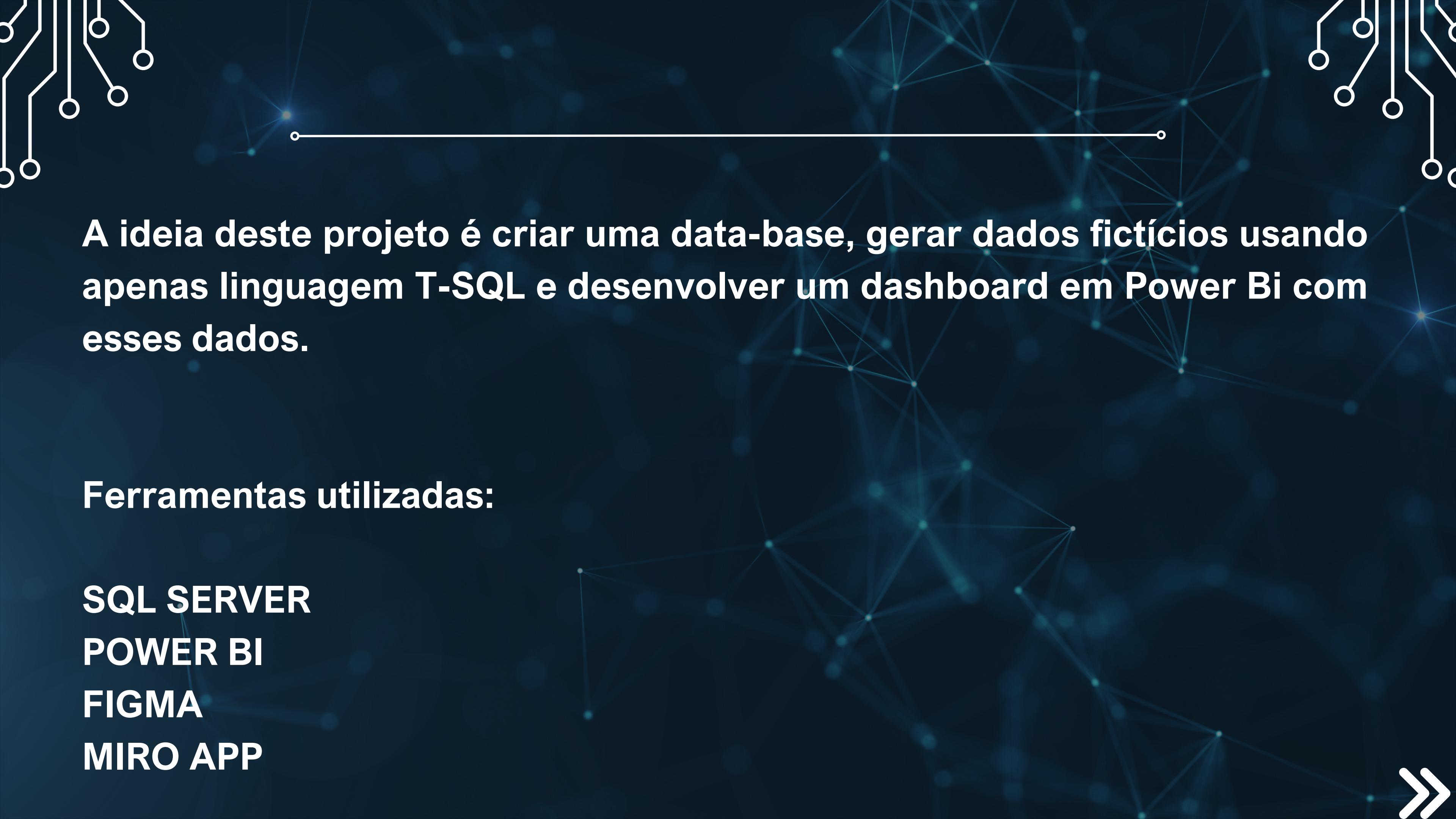


# **PROJETO - DA GERAÇÃO DE DADOS FICTIONÍCIOS AO DASHBOARD**



A ideia deste projeto é criar uma data-base, gerar dados fictícios usando apenas linguagem T-SQL e desenvolver um dashboard em Power Bi com esses dados.

Ferramentas utilizadas:

**SQL SERVER**

**POWER BI**

**FIGMA**

**MIRO APP**



# Primeira etapa

---

Entender a regra de negócio e definir a modelagem, aqui utilizaremos o app Miro.

Precisamos de um banco de dados para armazenar informações referentes aos alunos, campus e cursos.

Cada aluno só pode estar matriculado em um único curso.

Baseado nisso decidi utilizar o Star schema para definir a modelagem e criar as seguintes tabelas:

Tabela dimensão para armazenar informações dos alunos: nome, idade e gênero, id\_aluno.



**Tabela dimensão para armazenar informações dos campus: nome, cidade e estado, id\_campus.**

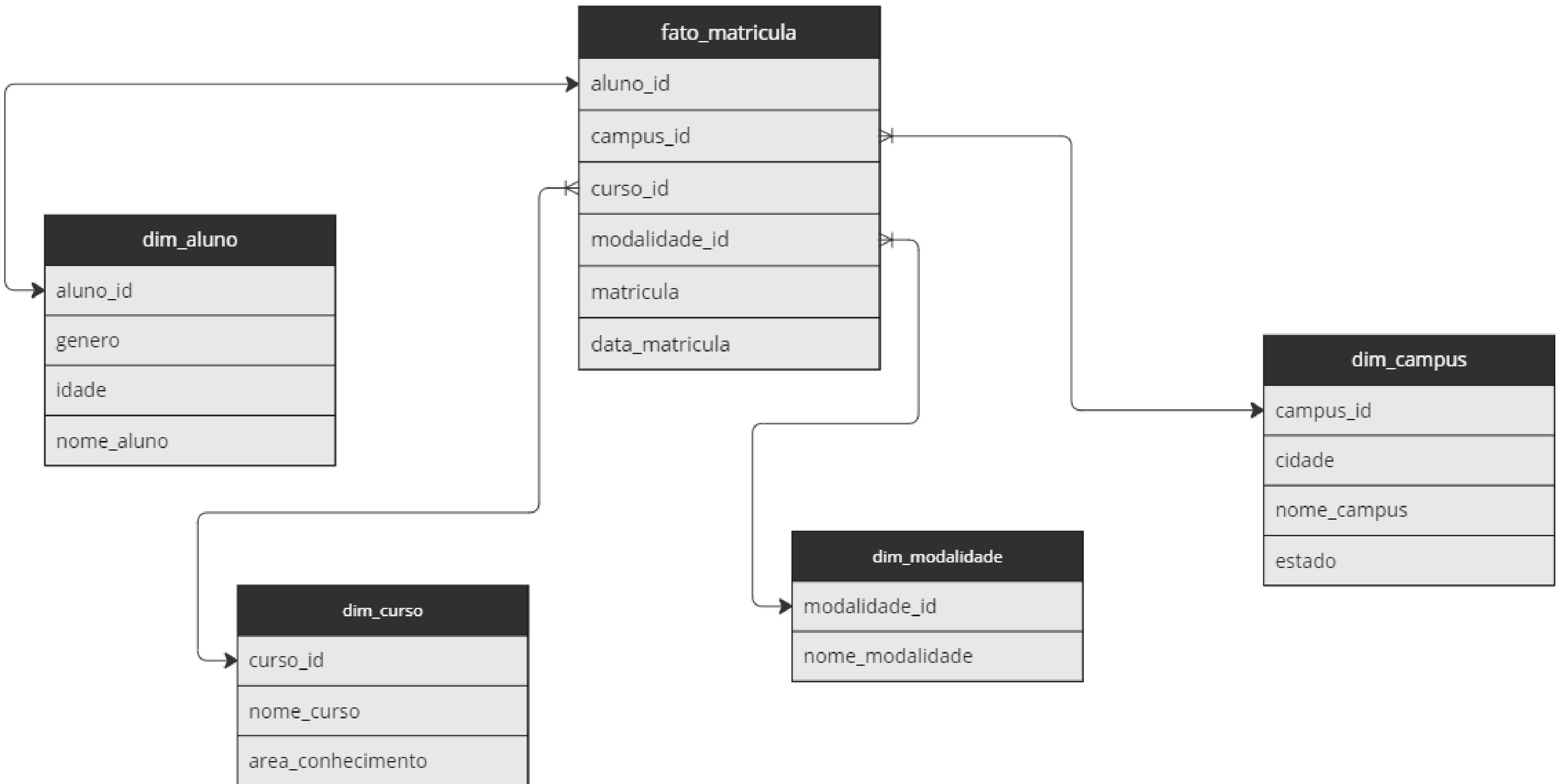
**Tabela dimensão para armazenar informações dos cursos: nome, área de conhecimento, id\_curso.**

**Tabela dimensão para armazenar informações das modalidades: nome, modalidade\_id.**

**Tabela fato contendo a data de matrícula do aluno, id de matricula, e as respectivas chaves para se conectar as tabelas dimensão.**

**Segue a imagem da modelagem feita, no app miro, no próximo slide.**





# Segunda etapa

**Criar a data-base, tabelas dimensão e tabela fato.**

**Nessa etapa vamos utilizar o SQL SERVER para criar a data-base contendo as tabelas definidas na modelagem.**

Segue a imagem do script .sql no próximo slide.



```
CREATE DATABASE X;

-- Uso do banco de dados
USE X;

-- Criação de Tabelas Dimensão

-- Criação da dimensão "curso"
CREATE TABLE dim_curso (
    curso_id INT PRIMARY KEY,
    nome_curso VARCHAR(100),
    area_conhecimento VARCHAR(50)
);

-- Criação da dimensão "aluno"
CREATE TABLE dim_aluno (
    aluno_id INT PRIMARY KEY,
    nome_aluno VARCHAR(100),
    idade INT,
    genero VARCHAR(10)
);

-- Criação da dimensão "campus"
CREATE TABLE dim_campus (
    campus_id INT PRIMARY KEY,
    nome_campus VARCHAR(100),
    cidade VARCHAR(50),
    estado VARCHAR(50)
);

-- Criação da dimensão "modalidade"
CREATE TABLE dim_modalidade (
    modalidade_id INT PRIMARY KEY,
    nome_modalidade VARCHAR(50)
);

CREATE TABLE fato_matricula (
    matricula_id INT PRIMARY KEY,
    aluno_id INT,
    curso_id INT,
    campus_id INT,
    modalidade_id INT,
    data_matricula DATE,
    FOREIGN KEY (aluno_id) REFERENCES dim_aluno(aluno_id),
    FOREIGN KEY (curso_id) REFERENCES dim_curso(curso_id),
    FOREIGN KEY (campus_id) REFERENCES dim_campus(campus_id),
    FOREIGN KEY (modalidade_id) REFERENCES dim_modalidade(modalidade_id)
);
```



# Terceira etapa

Inserir dados manualmente nas tabelas dimensão campus, curso e modalidade.

Nessa etapa vamos inserir os dados de forma manual usando apenas o comando **INSERT INTO VALUES**.

Segue a imagem do script .sql no próximo slide.



```
1 -- Inserção de dados na dimensão "curso"
2 -- Aqui inserimos dados fictícios de cursos com suas respectivas áreas de conhecimento
3 INSERT INTO dim_curso (curso_id, nome_curso, area_conhecimento) VALUES
4 (1, 'Engenharia Mecânica', 'Engenharia'),
5 (2, 'Ciência da Computação', 'Tecnologia'),
6 (3, 'Administração de Empresas', 'Administração'),
7 (4, 'Direito', 'Jurídico'),
8 (5, 'Medicina', 'Saúde'),
9 (6, 'Economia', 'Economia'),
10 (7, 'Psicologia', 'Ciências Humanas'),
11 (8, 'Arquitetura', 'Arquitetura'),
12 (9, 'Nutrição', 'Saúde'),
13 (10, 'Design Gráfico', 'Design'),
14 (11, 'Engenharia Elétrica', 'Engenharia'),
15 (12, 'Letras', 'Humanidades'),
16 (13, 'Biologia', 'Ciências Biológicas'),
17 (14, 'Marketing', 'Marketing'),
18 (15, 'Fisioterapia', 'Saúde'),
19 (16, 'História', 'Ciências Humanas');
20
21 -- Inserção de dados na dimensão "campus"
22 -- Aqui inserimos dados fictícios de campi com seus respectivos nomes, cidades e estados
23 INSERT INTO dim_campus (campus_id, nome_campus, cidade, estado) VALUES
24 (1, 'Campus São Paulo', 'São Paulo', 'São Paulo'), -- Sudeste
25 (2, 'Campus Rio de Janeiro', 'Rio de Janeiro', 'Rio de Janeiro'), -- Sudeste
26 (3, 'Campus Belo Horizonte', 'Belo Horizonte', 'Minas Gerais'), -- Sudeste
27 (4, 'Campus Porto Alegre', 'Porto Alegre', 'Rio Grande do Sul'), -- Sul
28 (5, 'Campus Recife', 'Recife', 'Pernambuco'), -- Nordeste
29 (6, 'Campus Salvador', 'Salvador', 'Bahia'), -- Nordeste
30 (7, 'Campus Curitiba', 'Curitiba', 'Paraná'), -- Sul
31 (8, 'Campus Fortaleza', 'Fortaleza', 'Ceará'), -- Nordeste
32 (9, 'Campus Goiânia', 'Goiânia', 'Goiás'), -- Centro-Oeste
33 (10, 'Campus Manaus', 'Manaus', 'Amazonas'); -- Norte
34
35 -- Inserção de dados na dimensão "modalidade"
36 -- Aqui inserimos dados fictícios de modalidades de ensino
37 INSERT INTO dim_modalidade (modalidade_id, nome_modalidade) VALUES
38 (1, 'Presencial'),
39 (2, 'EAD'),
40 (3, 'Semi_Presencial');
```



# Quarta etapa

Desenvolver um código em T-SQL para gerar dados fictícios e inserir esses dados na tabela dim\_aluno.

Aqui usei um loop WHILE para gerar dados fictícios de 4 mil alunos em T-SQL. Cada registro tem um aluno\_id único, nome\_aluno gerado automaticamente, idade entre 18 e 28 anos e gênero aleatório (Masculino ou Feminino). O loop incrementa o contador a cada iteração até atingir 4000 registros. Em seguida inseri os dados na tabela dim\_aluno usando o comando INSERT INTO.

Segue a imagem do script .sql no próximo slide.



```
1 -- Inserção de dados na dimensão "aluno"
2 -- Aqui inserimos dados fictícios de alunos com nomes, idades e gêneros aleatórios
3 DECLARE @i INT = 1;
4 WHILE @i <= 4000
5 BEGIN
6     INSERT INTO dim_aluno (aluno_id, nome_aluno, idade, genero) VALUES
7     (@i, 'Aluno ' + CAST(@i AS VARCHAR(10)), CAST(ROUND(RAND() * 10 + 18, 0) AS INT), CASE WHEN RAND() > 0.5 THEN 'Masculino' ELSE 'Feminino' END);
8     SET @i = @i + 1;
9 END;
10
```



# Quinta etapa

Desenvolver um código em SQL para gerar dados fictícios e inserir esses dados na tabela **fato\_matricula**.

Dessa vez utilizei o loop WHILE para gerar 4000 registros fictícios de matrículas, cada um com um aluno, curso, campus, modalidade e data de matrícula aleatórios em seguida usei o comando INSERT para inseri-los na tabela **fato\_matricula**.

Segue a imagem do script .sql no próximo slide.



```

1  -- Criação e distribuição de dados fictícios de forma aleatória.
2  DECLARE @matricula_id INT = 1;
3  DECLARE @aluno_id INT;
4  DECLARE @curso_id INT;
5  DECLARE @campus_id INT;
6  DECLARE @modalidade_id INT;
7  DECLARE @data_matricula DATE;
8
9  -- Criação de uma tabela temporária para controlar os alunos já matriculados
10 CREATE TABLE #alunos_matriculados (
11     aluno_id INT PRIMARY KEY
12 );
13
14 -- Loop para inserir dados fictícios de matrícula de alunos
15 WHILE @matricula_id <= 4000
16 BEGIN
17     -- Geração de um aluno não matriculado anteriormente
18     SET @aluno_id = (SELECT TOP 1 aluno_id FROM dim_aluno WHERE aluno_id NOT IN (SELECT aluno_id FROM #alunos_matriculados) ORDER BY NEWID());
19     -- Inserção do aluno na tabela temporária de controle
20     INSERT INTO #alunos_matriculados (aluno_id) VALUES (@aluno_id);
21
22     -- Geração de um curso aleatório ponderado para distribuir os alunos de forma mais equilibrada
23     SET @curso_id = (
24         CASE
25             WHEN RAND() < 0.19 THEN 1    -- Engenharia Mecânica
26             WHEN RAND() < 0.24 THEN 2    -- Ciência da Computação
27             WHEN RAND() < 0.15 THEN 3    -- Administração de Empresas
28             WHEN RAND() < 0.09 THEN 4    -- Direito
29             WHEN RAND() < 0.22 THEN 5    -- Medicina
30             WHEN RAND() < 0.25 THEN 6    -- Economia
31             WHEN RAND() < 0.1 THEN 7    -- Psicologia
32             WHEN RAND() < 0.025 THEN 8    -- Arquitetura
33             WHEN RAND() < 0.022 THEN 9    -- Nutrição
34             WHEN RAND() < 0.13 THEN 10   -- Design Gráfico
35             WHEN RAND() < 0.07 THEN 11   -- Engenharia Elétrica
36             WHEN RAND() < 0.04 THEN 12   -- Letras
37             WHEN RAND() < 0.063 THEN 13   -- Biologia
38             WHEN RAND() < 0.05 THEN 14   -- Marketing
39             WHEN RAND() < 0.11 THEN 15   -- Fisioterapia
40             ELSE 16                    -- História
41         END
42     );
43
44     -- Geração de um campus aleatório ponderado para distribuir os alunos de forma mais equilibrada
45     SET @campus_id = (
46         CASE
47             WHEN @curso_id IN (1, 5, 10, 11, 12, 13, 14, 15, 16) THEN ROUND(RAND(CHECKSUM(NEWID())) * 9, 0) + 1 -- Campus variando de 1 a 9
48             WHEN @curso_id IN (2, 3, 6, 7, 8, 9) THEN ROUND(RAND(CHECKSUM(NEWID())) * 4, 0) + 1 -- Campus variando de 1 a 4
49             WHEN @curso_id = 4 THEN ROUND(RAND(CHECKSUM(NEWID())) * 3, 0) + 1 -- Campus variando de 1 a 3
50             ELSE ROUND(RAND(CHECKSUM(NEWID())) * 2, 0) + 1 -- Campus variando de 1 a 2
51         END
52     );
53
54     -- Definindo a modalidade como 1 (presencial), 2 (EAD) ou 3 (Semi_Presencial)
55     SET @modalidade_id = (
56         CASE
57             WHEN RAND() < 0.47 THEN 1    -- presencial
58             WHEN RAND() < 0.38 THEN 2    -- EAD
59             ELSE 3                      -- Semi_Presencial
60         END
61     );
62
63     -- Geração de uma data de matrícula aleatória entre '2022-01-01' e '2023-01-01'
64     SET @data_matricula = DATEADD(day, (CAST(RAND() * 365 AS INT)), '2022-01-01');
65
66     -- Inserção dos dados de matrícula na tabela fato
67     INSERT INTO fato_matricula (matricula_id, aluno_id, curso_id, campus_id, modalidade_id, data_matricula)
68     VALUES (@matricula_id, @aluno_id, @curso_id, @campus_id, @modalidade_id, @data_matricula);
69
70     SET @matricula_id = @matricula_id + 1;
71 END;

```



**ANTES DE DESENVOLVER O DASHBOARD, VAMOS  
VER COMO FICOU NOSSO BANCO DE DADOS ?**

Segue a imagens das tabelas nos próximos slides.



## SQLQuery1.sql - DE...JB9U\henrique (51)\*

```
USE X;
SELECT TOP (1000) [aluno_id]
      ,[nome_aluno]
      ,[idade]
      ,[genero]
  FROM [X].[dbo].[dim_aluno]
```

100 %

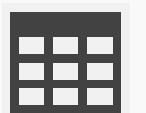
Resultados Mensagens

	aluno_id	nome_aluno	idade	genero
1	1	Aluno 1	19	Feminino
2	2	Aluno 2	26	Masculino
3	3	Aluno 3	19	Masculino
4	4	Aluno 4	24	Masculino
5	5	Aluno 5	20	Feminino
6	6	Aluno 6	26	Feminino
7	7	Aluno 7	24	Feminino
8	8	Aluno 8	20	Feminino
9	9	Aluno 9	22	Masculino
10	10	Aluno 10	25	Masculino

## SQLQuery4.sql - DE...JB9U\henrique (61)\* ✎ X

```
SELECT [modalidade_id],  
       [nome_modalidade]  
  FROM [X].[dbo].[dim_modalidade]
```

100 %



Resultados



Mensagens

	modalidade_id	nome_modalidade
1	1	Presencial
2	2	EAD
3	3	Semi_Presencial



```
SELECT [campus_id]
      ,[nome_campus]
      ,[cidade]
      ,[estado]
  FROM [X].[dbo].[dim_campus]
```

100 %

Resultados Mensagens

	campus_id	nome_campus	cidade	estado
1	1	Campus São Paulo	São Paulo	São Paulo
2	2	Campus Rio de Janeiro	Rio de Janeiro	Rio de Janeiro
3	3	Campus Belo Horizonte	Belo Horizonte	Minas Gerais
4	4	Campus Porto Alegre	Porto Alegre	Rio Grande do Sul
5	5	Campus Recife	Recife	Pernambuco
6	6	Campus Salvador	Salvador	Bahia
7	7	Campus Curitiba	Curitiba	Paraná
8	8	Campus Fortaleza	Fortaleza	Ceará
9	9	Campus Goiânia	Goiânia	Goiás
10	10	Campus Manaus	Manaus	Amazonas

## SQLQuery6.sql - DE...JB9U\henrique (62))\*

```
SELECT [curso_id]
      ,[nome_curso]
      ,[area_conhecimento]
   FROM [X].[dbo].[dim_curso]
```

100 %

Resultados Mensagens

	curso_id	nome_curso	area_conhecimento
1	1	Engenharia Mecânica	Engenharia
2	2	Ciência da Computação	Tecnologia
3	3	Administração de Empresas	Administração
4	4	Direito	Jurídico
5	5	Medicina	Saúde
6	6	Economia	Economia
7	7	Psicologia	Ciências Humanas
8	8	Arquitetura	Arquitetura
9	9	Nutrição	Saúde
10	10	Design Gráfico	Design
11	11	Engenharia Elétrica	Engenharia
12	12	Letras	Humanidades
13	13	Biologia	Ciências Biológicas
14	14	Marketing	Marketing
15	15	Fisioterapia	Saúde
16	16	História	Ciências Humanas



## SQLQuery7.sql - DE...JB9U\henrique (61))\*

```
SELECT TOP (1000) [matricula_id]
      ,[aluno_id]
      ,[curso_id]
      ,[campus_id]
      ,[modalidade_id]
      ,[data_matricula]
  FROM [X].[dbo].[fato_matricula]
```

100 %

Resultados Mensagens

	matricula_id	aluno_id	curso_id	campus_id	modalidade_id	data_matricula	
1	1	2210	12	4	2	2022-10-10	
2	2	1333	11	5	1	2022-12-13	
3	3	1162	16	10	3	2022-06-27	
4	4	1245	2	2	3	2022-03-03	
5	5	512	4	3	2	2022-12-10	
6	6	3039	1	2	1	2022-01-11	
7	7	2145	4	2	2	2022-11-15	
8	8	610	1	8	2	2022-09-28	
9	9	1535	6	2	3	2022-02-01	
10	10	186	16	9	3	2022-02-08	
11	11	2582	2	4	3	2022-03-20	
12	12	784	1	8	2	2022-09-13	

# Sexta etapa

Definir quais os KPI'S importantes e então elaborar o Storytelling, conectar o Power bi na base de dados e desenvolver o dashboard , aqui utilizei o Figma para fazer o background.

Principais KPI'S.

Quantidade de alunos em cada curso, campus e regiao;

Distribuição de alunos por modalidade;

Cursos mais populares e menos populares.

Segue a imagem do dashboard no próximo slide.



## HP DASHBOARD

### TOTAL DE ALUNOS

4000

### CURSO COM MAIS ALUNOS

Engenharia  
Mecânica

### CURSO COM MENOS ALUNOS

Arquitetura

### Estado

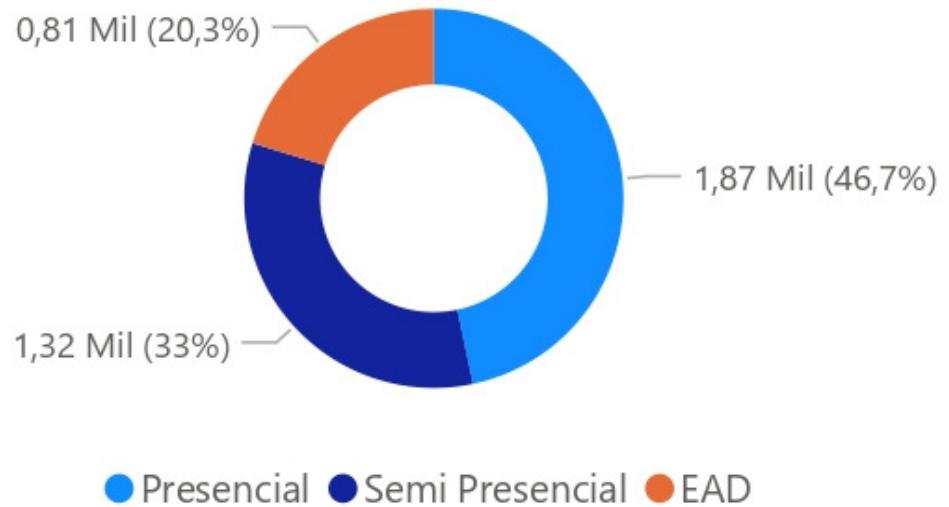
Todos

### Curso

Todos

**Limpar Filtros**

### Quantidade de Alunos por Modalidade



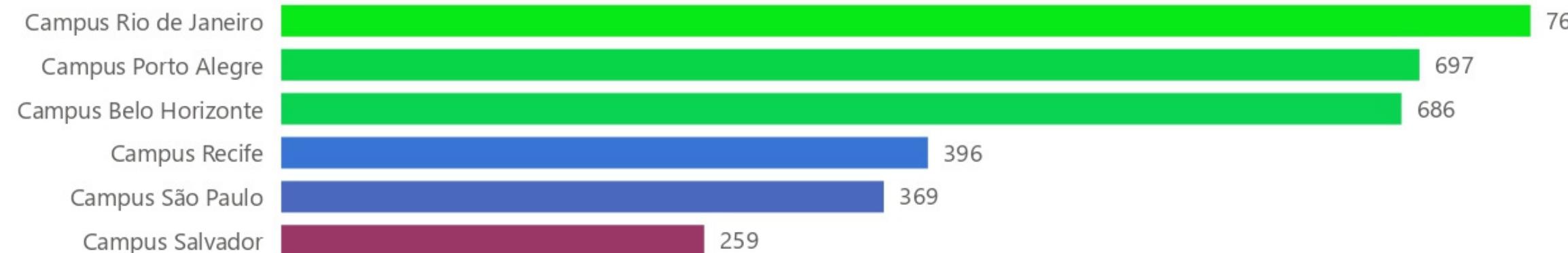
### Quantidade de Alunos por Estado



### Quantidade de Alunos por Curso



### Quantidade de Alunos por Campus



### Ranking Alunos por Campus



Campus Rio de Janeiro



Campus Porto Alegre



Campus Belo Horizonte



# OBRIGADO!



<https://www.linkedin.com/in/henriqueproscholdt/>



<https://github.com/proscholdt>



[proscholdt.h2014@gmail.com](mailto:proscholdt.h2014@gmail.com)



27995130691