



**DESENVOLVIMENTO: POWER BI PARA UMA
INSTITUIÇÃO DE ENSINO SUPERIOR FICTÍCIA**

Autor: Willian Ferreira Gonçalves

Curitiba, 02 de junho de 2025.

1. GERAÇÃO DOS DADOS

Como trata-se de um projeto fictício, para alimentar a base de dados, foi pedido a uma ferramenta de inteligência artificial a geração de uma planilha excel com os dados para então alimentar o Banco de Dados a ser criado. A ferramenta de IA utilizada foi o ChatGPT. Abaixo, segue a pergunta e contextualização feita a ele, para a geração da planilha:

PROJETO POWER BI:

Para validar e aprofundar meus conhecimentos, trazendo a teoria para a prática, quero desenvolver um projeto de Power BI. Primeiramente, vamos seguir com a base de dados, onde ressalto que tudo será inserido de forma fictícia:

CONTEXTUALIZAÇÃO DE CENÁRIO:

- Consideremos uma instituição fictícia de ensino superior que atende os estados: Paraná, Rio Grande do Sul, Santa Catarina, São Paulo, Rio de Janeiro, Minas Gerais, Mato Grosso e Mato Grosso do Sul, e Goiás;

- Em cada um dos estados, há polos em 10 cidades, incluindo a capital (pegar cidades aleatórias);

Assim sendo, construa uma planilha em Excel com os seguintes dados:

- Cidade, Estado, UFs e Nomes de Polos (seguindo os critérios acima);
- Data de recolhimento do dado (Data Considerada);
- Total de estudantes ativos (cursando), formados, desistentes e que trancaram seus cursos;
- Descrição dos cursos e total de vagas disponíveis para cada;
- Preço dos cursos;
- Satisfação dos estudantes com os cursos, de 1 a 5, onde: 1-2 = não superou minhas expectativas; 2-3 = superou pouco minhas expectativas; 3-3.9 = superou parcialmente minhas expectativas, ainda há pontos a melhorar; 4-5 = superou totalmente minhas expectativas;
- Status de pagamento: Adimplente ou Inadimplente.

O ChatGPT gerou uma planilha Excel com 451 linhas de dados, com as colunas: **Cidade, Estado, UF, Nome do Polo, Mês, Total Estudantes Ativos, Total Estudantes Formados, Total Estudantes Desistentes, Total Trancamentos, Descrição Curso, Total Vagas Disponíveis, Preços Cursos, Satisfação com o Curso e Status de Pagamento.**

2. MODELAGEM DA BASE DE DADOS

Por mais que seja um projeto fictício, se estabeleceu a ideia de criar a base de dados do Power BI em um Banco SQL, via SGBD Microsoft SQL Server Management Studio 2022, para uma tratativa de dados tal qual é, comumente, em cenários reais. Primeiramente, segue abaixo a modelagem dos dados:

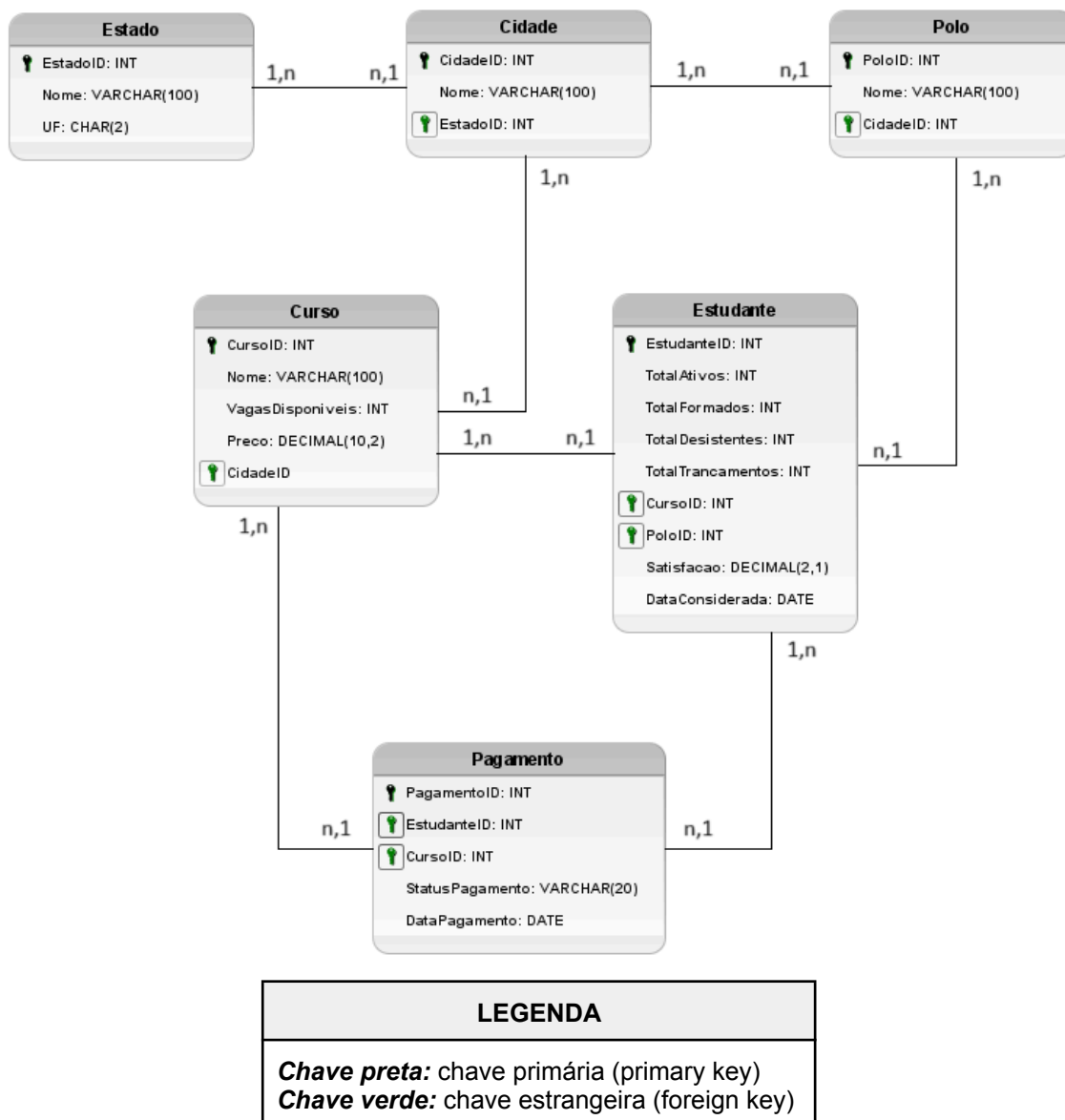


Figura 01 — Modelo Entidade-Relacionamento (MER) do projeto.

2.1. Relacionamento Estado Cidade

Direção: Estado → Cidade (1:N)

- **Significado:** Um Estado pode ter **muitas Cidades**. Ou seja, cada registro na tabela Estado está relacionado a vários registros na tabela Cidade.
- **Implementação:** A tabela Cidade possui um campo EstadoID como chave estrangeira (FOREIGN KEY) apontando para Estado(EstadoID).
- **Exemplo:** O Estado “São Paulo” pode ter cidades como “Campinas”, “Santos”, “Ribeirão Preto” etc.

Direção contrária: Cidade → Estado (N:1)

- **Significado:** Cada Cidade pertence a um único Estado.
- **Consequência lógica:** Não pode existir uma cidade que esteja vinculada a dois ou mais estados simultaneamente.
- **Exemplo:** A cidade de “Campinas” pertence **exatamente** ao Estado “São Paulo”.

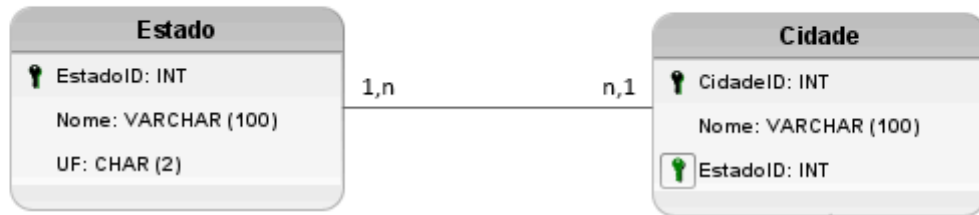


Figura 02 — Modelo Entidade-Relacionamento (MER) das tabelas Estado e Cidade.

2.2. Relacionamento Cidade Polo

Direção: Cidade → Polo (1:N)

- **Significado:** Cada Cidade pode ter muitos Polos. Ou seja, um registro na tabela Cidade pode estar relacionado a vários registros na tabela Polo.
- **Implementação:** A tabela Polo possui um campo CidadeID como chave estrangeira (FOREIGN KEY) apontando para Cidade(CidadeID).
- **Exemplo:** A cidade de "São Paulo" pode ter polos como "Polo Central", "Polo Zona Sul", "Polo Zona Norte", etc.

Direção contrária: Polo → Cidade (N:1)

- **Significado:** Cada Polo pertence a uma única Cidade.
- **Consequência lógica:** Não pode existir um polo que pertença a duas ou mais cidades ao mesmo tempo.
- **Exemplo:** O "Polo Central" pertence exclusivamente à cidade de "São Paulo".

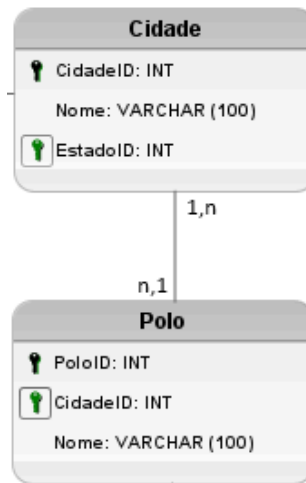


Figura 03 — Modelo Entidade-Relacionamento (MER) das tabelas Cidade e Polo.

2.3. Relacionamento Polo Estudante

Direção: Polo → Estudante (1:N)

- **Significado:** Cada Polo pode ter muitos Estudantes. Ou seja, um registro na tabela Polo pode estar relacionado a vários registros na tabela Estudante.
- **Implementação:** A tabela Estudante possui um campo PoloID como chave estrangeira (FOREIGN KEY) apontando para Polo(PoloID).
- **Exemplo:** O "Polo Central" pode ter vários Estudantes como "João", "Maria", "Carlos", etc.

Direção contrária: Estudante → Polo (N:1)

- **Significado:** Cada Estudante pertence a um único Polo.
- **Consequência lógica:** Não pode existir um Estudante que pertença a dois ou mais Polos simultaneamente.
- **Exemplo:** O Estudante "João" pertence ao "Polo Central".

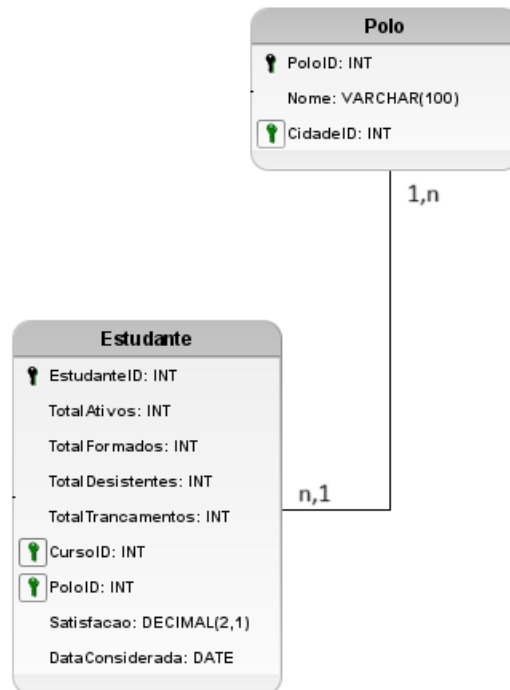


Figura 04 — Modelo Entidade-Relacionamento (MER) das tabelas Polo e Estudante.

2.4. Relacionamento Curso Estudante

Direção: Curso → Estudante (1:N)

- **Significado:** Um Curso pode ter vários Estudantes matriculados.
- **Implementação:** A tabela Estudante possui o campo CursoID como chave estrangeira referenciando Curso(CursoID).
- **Exemplo:** O curso “Engenharia de Software” pode ter 200 alunos ativos.

Direção contrária: Estudante → Curso (N:1)

- **Significado:** Cada Estudante está matriculado em um único Curso.
- **Exemplo:** A estudante Maria está matriculada somente no curso de “Engenharia de Software”.

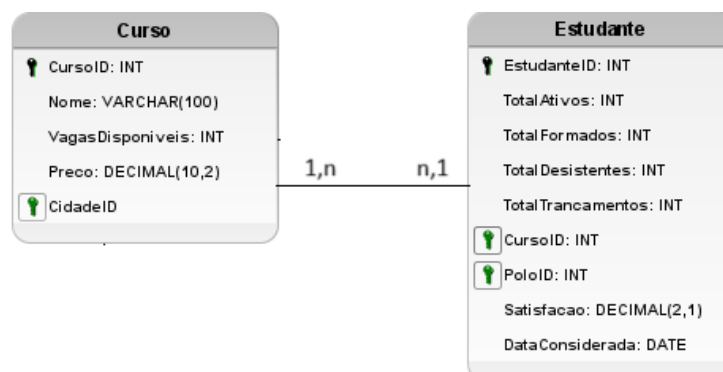


Figura 05 — Modelo Entidade-Relacionamento (MER) das tabelas Curso e Estudante.

2.5. Relacionamento Curso Cidade

Direção: Cidade → Curso (1:N)

- **Significado:** Cada Cidade pode ter muitos Cursos. Ou seja, um registro na tabela Cidade pode estar relacionado a vários registros na tabela Curso.
- **Implementação:** A tabela Curso possui um campo CidadeID como chave estrangeira (FOREIGN KEY) apontando para Cidade(CidadeID).
- **Exemplo:** A cidade de "São Paulo" pode ter cursos como "Curso de Engenharia", "Curso de Medicina", "Curso de Administração", etc.

Direção contrária: Curso → Cidade (N:1)

- **Significado:** Cada Curso é oferecido em uma única Cidade.
- **Consequência lógica:** Não pode existir um curso que seja oferecido em duas ou mais cidades simultaneamente.
- **Exemplo:** O "Curso de Engenharia" é oferecido exclusivamente na cidade de "São Paulo".

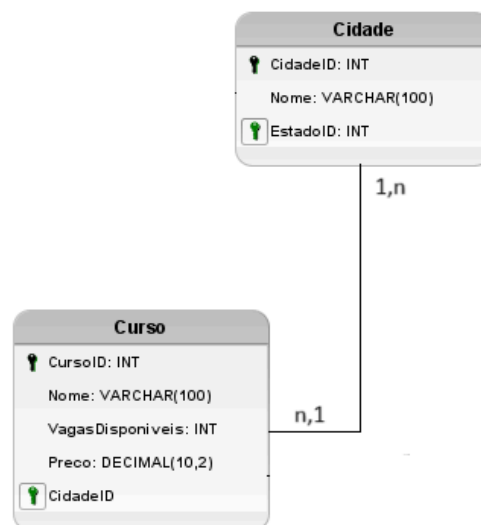


Figura 06 — Modelo Entidade-Relacionamento (MER) das tabelas Curso e Cidade.

2.6. Relacionamento Estudante Pagamento

Direção: Estudante → Pagamento (1:N)

- **Significado:** Cada Estudante pode ter muitos Pagamentos. Ou seja, um registro na tabela Estudante pode estar relacionado a vários registros na tabela Pagamento.

- **Implementação:** A tabela Pagamento possui um campo EstudanteID como chave estrangeira (FOREIGN KEY) apontando para Estudante(EstudanteID).
- **Exemplo:** O Estudante "João" pode ter vários pagamentos, como o "Pagamento do mês de Janeiro" ou "Pagamento do mês de Fevereiro".

Direção contrária: Pagamento → Estudante (N:1)

- **Significado:** Cada Pagamento é associado a um único Estudante.
- **Consequência lógica:** Não pode existir um pagamento que esteja vinculado a dois ou mais Estudantes simultaneamente.
- **Exemplo:** O "Pagamento do mês de Janeiro" está associado ao Estudante "João".

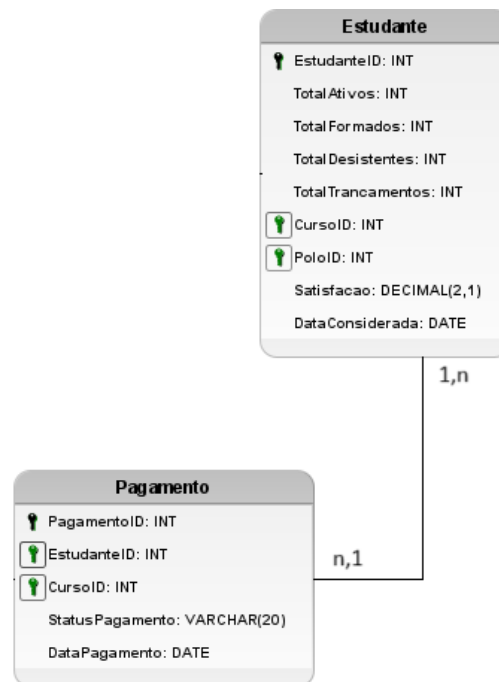


Figura 07 — Modelo Entidade-Relacionamento (MER) das tabelas Estudante e Pagamento.

2.7. Relacionamento Curso Pagamento

Direção: Curso → Pagamento (1:N)

- **Significado:** Cada Curso pode ter muitos Pagamentos. Ou seja, um registro na tabela Curso pode estar relacionado a vários registros na tabela Pagamento.
- **Implementação:** A tabela Pagamento possui um campo CursoID como chave estrangeira (FOREIGN KEY) apontando para Curso(CursoID).

- **Exemplo:** O "Curso de Engenharia" pode ter vários pagamentos, como "Pagamento de janeiro", "Pagamento de fevereiro", etc.

Direção contrária: Pagamento → Curso (N:1)

- **Significado:** Cada Pagamento está associado a um único Curso.
- **Consequência lógica:** Não pode existir um pagamento vinculado a dois ou mais cursos simultaneamente.
- **Exemplo:** O "Pagamento de Janeiro" está associado ao "Curso de Engenharia".

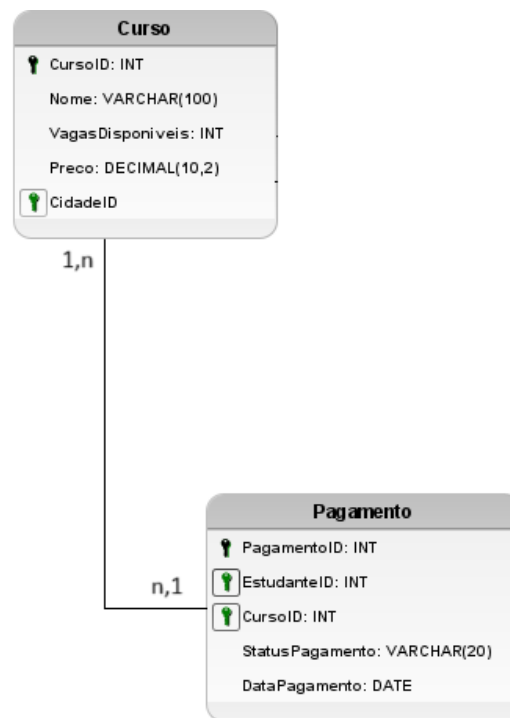


Figura 08 — Modelo Entidade-Relacionamento (MER) das tabelas Curso e Pagamento.

3. CRIAÇÃO DA BASE DE DADOS

1º) Primeiramente, na versão 2022 do SGBD Microsoft SQL Server Management Studio, em um servidor local mesmo (numa máquina), fez-se a criação do Banco de Dados por nome "Instituicao_Educacional_BI":

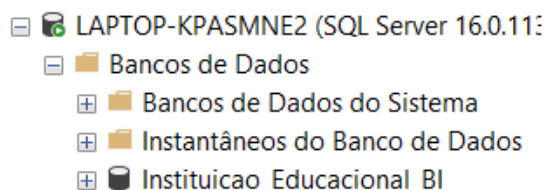


Figura 09 — Nome do banco de dados criado, no SQL Server.

2º) Em seguida, executou-se a Query para a criação das tabelas e seus relacionamentos, determinados anteriormente na Modelagem:

```
-- Tabela de Estados
CREATE TABLE Estado (
    EstadoID INT IDENTITY(1,1) PRIMARY KEY,
    Nome VARCHAR(100) NOT NULL,
    UF CHAR(2) NOT NULL
);

-- Tabela de Cidades
CREATE TABLE Cidade (
    CidadeID INT IDENTITY(1,1) PRIMARY KEY,
    Nome VARCHAR(100) NOT NULL,
    EstadoID INT NOT NULL,
    FOREIGN KEY (EstadoID) REFERENCES Estado(EstadoID)
);

-- Tabela de Polos
CREATE TABLE Polo (
    PoloID INT IDENTITY(1,1) PRIMARY KEY,
    Nome VARCHAR(100) NOT NULL,
    CidadeID INT NOT NULL,
    FOREIGN KEY (CidadeID) REFERENCES Cidade(CidadeID)
);

-- Tabela de Cursos
CREATE TABLE Curso (
    CursoID INT IDENTITY(1,1) PRIMARY KEY,
    Nome VARCHAR(100) NOT NULL,
    VagasDisponiveis INT NOT NULL,
    Preco DECIMAL(10,2) NOT NULL,
    CidadeID INT NOT NULL,
    FOREIGN KEY (CidadeID) REFERENCES Cidade(CidadeID)
);

-- Tabela de Estudantes
CREATE TABLE Estudante (
    EstudanteID INT IDENTITY(1,1) PRIMARY KEY,
    Total_Ativos INT,
    Total_Formados INT,
    Total_Desistentes INT,
    Total_Trancamentos INT,
    CursoID INT NOT NULL,
    PoloID INT NOT NULL,
    Satisfacao DECIMAL(2,1) CHECK (Satisfacao BETWEEN 1.0 AND 5.0),
    DataConsiderada DATE,
```

```

FOREIGN KEY (CursoID) REFERENCES Curso(CursoID),
FOREIGN KEY (PoloID) REFERENCES Polo(PoloID)
);

-- Tabela de Pagamentos
CREATE TABLE Pagamento (
  PagamentoID INT IDENTITY(1,1) PRIMARY KEY,
  EstudanteID INT NOT NULL,
  CursoID INT NOT NULL,
  StatusPagamento VARCHAR(20) CHECK (StatusPagamento IN
('Adimplente', 'Inadimplente')) NOT NULL,
  FOREIGN KEY (EstudanteID) REFERENCES
Estudante(EstudanteID),
  FOREIGN KEY (CursoID) REFERENCES Curso(CursoID)
);
  
```

3.1. Explicação linha a linha — Tabela de Estados

CÓDIGO SQL	DESCRIÇÃO
<code>CREATE TABLE Estado</code>	Cria uma nova tabela chamada Estado .
<code>EstadoID INT IDENTITY(1,1) PRIMARY KEY,</code>	Cria a coluna EstadoID : <ul style="list-style-type: none"> Tipo INT (número inteiro). IDENTITY(1,1) = autoincremento, começando em 1 e aumentando de 1 em 1. PRIMARY KEY = chave primária, valor único que identifica cada estado.
<code>Nome VARCHAR(100) NOT NULL</code>	Cria a coluna Nome : <ul style="list-style-type: none"> VARCHAR(100) = aceita texto de até 100 caracteres. NOT NULL = não pode ficar vazia.
<code>UF CHAR(2) NOT NULL</code>	Cria a coluna UF : <ul style="list-style-type: none"> CHAR(2) = exatamente 2 caracteres, armazena a sigla do estado, ex: 'SP', 'RJ'. NOT NULL = obrigatório.

3.2. Explicação linha a linha — Tabela de Cidades

CÓDIGO SQL	DESCRIÇÃO
<code>CREATE TABLE Cidade</code>	Cria uma nova tabela chamada Cidade .
<code>CidadeID INT IDENTITY(1,1) PRIMARY KEY</code>	Cria a coluna CidadeID : <ul style="list-style-type: none"> • INT = os dados são números inteiros. • IDENTITY(1,1) = autoincremento, começando em 1 e aumentando de 1 em 1. • PRIMARY KEY = chave primária, valor único que identifica cada cidade.
<code>Nome VARCHAR(100) NOT NULL,</code>	Cria a coluna Nome : <ul style="list-style-type: none"> • VARCHAR(100) = aceita texto de até 100 caracteres. • NOT NULL = não pode ficar vazia.
<code>EstadoID INT NOT NULL</code>	Cria a coluna EstadoID : <ul style="list-style-type: none"> • INT = os dados são números inteiros. • NOT NULL = não pode ficar vazia.
<code>FOREIGN KEY (EstadoID) REFERENCES Estado(EstadoID)</code>	<ul style="list-style-type: none"> • Define que a coluna EstadoID é uma chave estrangeira. • Ela aponta para a coluna EstadoID da tabela Estado. • Isso cria uma ligação entre Cidade e Estado.

3.3. Explicação linha a linha — Tabela de Polos

CÓDIGO SQL	DESCRIÇÃO
<code>CREATE TABLE Polo</code>	Cria uma nova tabela chamada Polo .
<code>PoloID INT IDENTITY(1,1) PRIMARY KEY</code>	Cria a coluna PoloID : <ul style="list-style-type: none"> • INT = os dados são números inteiros. • IDENTITY(1,1) = autoincremento, começando em 1 e aumentando de 1 em 1.

	<ul style="list-style-type: none"> PRIMARY KEY = chave primária, valor único que identifica cada polo.
Nome VARCHAR(100) NOT NULL	Cria a coluna Nome : <ul style="list-style-type: none"> VARCHAR(100) = aceita texto de até 100 caracteres. NOT NULL = não pode ficar vazia.
CidadeID INT NOT NULL	Cria a coluna CidadeID : <ul style="list-style-type: none"> INT = os dados são números inteiros. NOT NULL = não pode ficar vazia.
FOREIGN KEY (CidadeID) REFERENCES Cidade(CidadeID)	<ul style="list-style-type: none"> Define que a coluna CidadeID é uma chave estrangeira. Ela aponta para a coluna CidadeID da tabela Cidade. Isso cria uma ligação entre Polo e Cidade.

3.4. Explicação linha a linha — Tabela de Cursos

CÓDIGO SQL	DESCRIÇÃO
CREATE TABLE Curso	Cria uma nova tabela chamada Curso .
CursoID INT IDENTITY(1,1) PRIMARY KEY	Cria a coluna CursoID : <ul style="list-style-type: none"> INT = os dados são números inteiros. IDENTITY(1,1) = autoincremento, começando em 1 e aumentando de 1 em 1. PRIMARY KEY = chave primária, valor único que identifica cada curso.
Nome VARCHAR(100) NOT NULL	Cria a coluna Nome : <ul style="list-style-type: none"> VARCHAR(100) = aceita texto de até 100 caracteres. NOT NULL = não pode ficar vazia.
VagasDisponiveis INT NOT NULL	Cria a coluna VagasDisponiveis : <ul style="list-style-type: none"> INT = os dados são números inteiros.

	<ul style="list-style-type: none"> • NOT NULL = não pode ficar vazia.
<pre>Preco DECIMAL(10,2) NOT NULL</pre>	Cria a coluna Preco : <ul style="list-style-type: none"> • DECIMAL(10,2) = número decimal com 10 dígitos no total, com duas casas após o ponto. • NOT NULL = não pode ficar vazia
<pre>CidadeID INT NOT NULL</pre>	Cria a coluna CidadeID : <ul style="list-style-type: none"> • INT = os dados são números inteiros. • NOT NULL = não pode ficar vazia
<pre>FOREIGN KEY (CidadeID) REFERENCES Cidade(CidadeID)</pre>	<ul style="list-style-type: none"> • Define que a coluna CidadeID é uma chave estrangeira. • Ela aponta para a coluna CidadeID da tabela Cidade. • Isso cria uma ligação entre Curso e Cidade.

3.5. Explicação linha a linha — Tabela de Estudantes

CÓDIGO SQL	DESCRIÇÃO
<pre>CREATE TABLE Estudante</pre>	Cria uma nova tabela chamada Estudante .
<pre>EstudanteID INT IDENTITY(1,1) PRIMARY KEY</pre>	Cria a coluna EstudanteID : <ul style="list-style-type: none"> • INT = os dados são números inteiros. • IDENTITY(1,1) = autoincremento, começando em 1 e aumentando de 1 em 1. • PRIMARY KEY = chave primária, valor único que identifica cada curso.
<pre>Total_Ativos INT, Total_Formados INT, Total_Desistentes INT, Total_Trancamentos INT</pre>	Cria as colunas Total Ativos , Total Formados , Total Desistentes e Total Trancamentos : <ul style="list-style-type: none"> • INT = os dados são números inteiros.
<pre>CursoID INT NOT NULL, PoloID INT NOT NULL,</pre>	Cria as colunas CursoID e PoloID : <ul style="list-style-type: none"> • INT = os dados são números inteiros.

	<ul style="list-style-type: none"> • NOT NULL = não podem ficar vazias.
Satisfacao <code>DECIMAL(2,1) CHECK (Satisfacao BETWEEN 1.0 AND 5.0)</code>	Cria a coluna Satisfacao : <ul style="list-style-type: none"> • DECIMAL(2,1) = número decimal com 2 dígitos no total, com uma casa após o ponto. • CHECK (Satisfacao BETWEEN 1.0 AND 5.0) = é uma restrição de integridade. Garante que os valores estejam entre 1.0 e 5.0, não sendo aceitos outros que estejam fora desse intervalo.
DataConsiderada <code>DATE</code>	Cria a coluna DataConsiderada : <ul style="list-style-type: none"> • DATE = os dados são datas, no formato norte-americano YYYY-MM-DD.
<code>FOREIGN KEY (CursoID) REFERENCES Curso(CursoID),</code> <code>FOREIGN KEY (PoloID) REFERENCES Polo(PoloID)</code>	<ul style="list-style-type: none"> • Define que as colunas CursoID e PoloID são chaves estrangeiras. • Elas apontam, respectivamente, para a coluna CursoID da tabela Curso e para a coluna PoloID da tabela Polo. • Isso cria ligações entre Estudante e Curso e Estudante e Polo, respectivamente.

3.6. Explicação linha a linha — Tabela de Pagamentos

CÓDIGO SQL	DESCRIÇÃO
<code>CREATE TABLE Pagamento</code>	Cria uma nova tabela chamada Pagamento .
PagamentoID <code>INT IDENTITY(1,1) PRIMARY KEY</code>	Cria a coluna PagamentoID : <ul style="list-style-type: none"> • INT = os dados são números inteiros. • IDENTITY(1,1) = autoincremento, começando em 1 e aumentando de 1 em 1. • PRIMARY KEY = chave primária, valor único que identifica cada curso.

EstudanteID INT NOT NULL, CursoID INT NOT NULL,	Criam as colunas EstudanteID e CursoID : <ul style="list-style-type: none"> • INT = os dados são números inteiros. • NOT NULL = não podem ficar vazias
StatusPagamento VARCHAR(20) CHECK (StatusPagamento IN ('Adimplente', 'Inadimplente')) NOT NULL,	Cria a coluna StatusPagamento : <ul style="list-style-type: none"> • VARCHAR(20) = aceita texto de até 20 caracteres. • CHECK = faz a validação se o texto inserido é, obrigatoriamente, “Adimplente” ou “Inadimplente”. • NOT NULL = não pode ficar vazia.
FOREIGN KEY (EstudanteID) REFERENCES Estudante(EstudanteID), FOREIGN KEY (CursoID) REFERENCES Curso(CursoID)	<ul style="list-style-type: none"> • Define que as colunas CursoID e EstudanteID são chaves estrangeiras. • Elas apontam, respectivamente, para a coluna CursoID da tabela Curso e para a coluna EstudanteID da tabela Estudante. • Isso cria ligações entre Estudante e Pagamento e Pagamento e Curso, respectivamente.

4. IMPORTAÇÃO DOS DADOS PARA AS TABELAS DA BASE DE DADOS

Como os dados que irão compor e alimentar a base de dados foram gerados pela IA e os mesmos foram inseridos em uma planilha Excel (xlsx) única, primeiramente, será necessário um tratamento (anexo a este documento, essa planilha gerada pode ser consultada, o nome é “Dados_Projeto_Educacao_BI”).

Neste momento, cabe retomar e contextualizar o que foi pedido à IA, mais especificamente ChatGPT, gerar de dados:

PROJETO POWER BI:

Para validar e aprofundar meus conhecimentos, trazendo a teoria para a prática, quero desenvolver um projeto de Power BI. Primeiramente, vamos seguir com a base de dados, onde ressalto que tudo será inserido de forma fictícia:

CONTEXTUALIZAÇÃO DE CENÁRIO:

- Consideremos uma instituição fictícia de ensino superior que atende os estados: Paraná, Rio Grande do Sul, Santa Catarina, São Paulo, Rio de Janeiro, Minas Gerais, Mato Grosso e Mato Grosso do Sul, e Goiás;
- Em cada um dos estados, há polos em 10 cidades, incluindo a capital (pegar cidades aleatórias);

Assim sendo, construa uma planilha em Excel com os seguintes dados:

- Cidade e Estado (seguindo os critérios acima);
- Data de recolhimento do dado;
- Total de estudantes ativos (cursando), formados, desistentes e que trancaram seus cursos;
- Descrição dos cursos e total de vagas disponíveis para cada;
- Preço dos cursos;
- Satisfação dos estudantes com os cursos, de 1 a 5, onde: 1-2 = não minhas expectativas; 2-3 = superou pouco minhas expectativas; 3-4 = superou parcialmente minhas expectativas, ainda há pontos a melhorar; 4-5 = superou totalmente minhas expectativas;
- Status do pagamento (Adimplente ou Inadimplente);

ATENÇÃO:

Neste momento, de modo a ser didático e evitar interpretações erradas:

- O termo “**Campo**” será empregado quando se estiver tratando sobre a planilha de dados gerada pela IA (“Dados_Projeto_Educacao_BI.xlsx”) e que irá alimentar a base de dados;
- O termo “**Coluna**” será empregado quando se estiver tratando sobre as tabelas da base de dados (“Instituicao_Educacional_BI”);

4.1. Dados das tabelas de Estados, Cidades e Polos

Na planilha gerada, foi possível identificar que existem os campos “Estado_Polo” e “UF”, cujos os dados neles contidos serão importados para as colunas “Nome” e “UF” da tabela de Estados. Averiguou-se que, de fato, existem os 9 estados que a instituição fictícia abrange, e que estão no escopo do projeto (Paraná, Rio Grande do Sul, Santa Catarina, São Paulo, Rio de Janeiro, Minas Gerais, Mato Grosso e Mato Grosso do Sul, e Goiás), contudo, estão inseridos várias vezes, pois foi requisitado à IA a inserção de 10 cidades aleatórias de cada um destes estados e considerá-las como polos da tal instituição. Assim sendo, nesse contexto, acabou que uma só cidade possui uma quantidade X de polos e seu nome acabou sendo repetido essa tal quantidade X na planilha, juntamente com o estado e a UF. Segue uma breve exemplificação abaixo:

Cidade_Polo	Estado_Polo	UF	Nome_Polo
Curitiba	Paraná	PR	POLO DE CURITIBA 1
Curitiba	Paraná	PR	POLO DE CURITIBA 2
Curitiba	Paraná	PR	POLO DE CURITIBA 3
Curitiba	Paraná	PR	POLO DE CURITIBA 4
Curitiba	Paraná	PR	POLO DE CURITIBA 5
Londrina	Paraná	PR	POLO DE LONDRINA 1
Londrina	Paraná	PR	POLO DE LONDRINA 2
Londrina	Paraná	PR	POLO DE LONDRINA 3
Londrina	Paraná	PR	POLO DE LONDRINA 4
Londrina	Paraná	PR	POLO DE LONDRINA 5

Figura 07 — Exemplificação dos dados repetidos que a IA gerou para Cidade, Estado, UF e Nome do Polo.

Por exemplo, a cidade de Curitiba, no estado do Paraná, cuja UF é PR, possui 5 polos no total. Assim sendo, os campos se repetiram 5 vezes. O mesmo ocorre com Londrina e os dados subsequentes da planilha gerada.

Todavia, para as tabelas de Estado, Cidade e Polo, não há a necessidade de haver todas essas repetições, haja vista que as devidas quantificações serão proporcionadas pela existência das chaves estrangeiras, interligando as tabelas.

Então, através dos comandos INSERT INTO e VALUES, do SQL, procedeu-se com as inserções dos dados dos campos “Estado_Polo”, “Cidade_Polo”, “UF” e “Nome_Polo” para as tabelas “Estado”, “Cidade” e “Polo”, respectivamente:

1º) DADOS DA TABELA ESTADO:

```

-- IMPORTAÇÃO DOS DADOS DA TABELA ESTADO:
SELECT * FROM Estado

INSERT INTO Estado (Nome, UF)
VALUES
  ('Paraná', 'PR'),
  ('São Paulo', 'SP'),
  ('Rio de Janeiro', 'RJ'),
  ('Rio Grande do Sul', 'RS'),
  ('Santa Catarina', 'SC'),
  ('Minas Gerais', 'MG'),
  ('Mato Grosso', 'MT'),
  ('Mato Grosso do Sul', 'MS'),
  ('Goiás', 'GO');
  
```

EstadoID	Nome	UF
1	Paraná	PR
2	São Paulo	SP
3	Rio de Janeiro	RJ
4	Rio Grande do Sul	RS
5	Santa Catarina	SC
6	Minas Gerais	MG
7	Mato Grosso	MT
8	Mato Grosso do Sul	MS
9	Goiás	GO

Figura 08 — Dados inseridos na tabela Estado, na base de dados.

Como “EstadoID” foi definido como auto incremento, o seu preenchimento foi automático por parte do SQL, não houve a necessidade nem de referenciá-lo na inserção dos dados nas linhas das demais colunas, “Nome” e “UF”.

2º) DADOS DA TABELA CIDADES:

```
-- IMPORTAÇÃO DOS DADOS DA TABELA CIDADE:
```

```
SELECT * FROM Cidade
```

```
INSERT INTO Cidade (Nome, EstadoID)
```

```
VALUES
```

```
    ('Curitiba', 1),  
    ('Londrina', 1),  
    ('Maringá', 1),  
    ('Cascavel', 1),  
    ('Ponta Grossa', 1),  
    ('Foz do Iguaçu', 1),  
    ('Guarapuava', 1),  
    ('Toledo', 1),  
    ('Arapongas', 1),  
    ('São José dos Pinhais', 1),  
    ('Porto Alegre', 4),  
    ('Caxias do Sul', 4),  
    ('Pelotas', 4),  
    ('Santa Maria', 4),  
    ('Gravataí', 4),  
    ('Canoas', 4),  
    ('Novo Hamburgo', 4),  
    ('Viamão', 4),  
    ('São Leopoldo', 4),  
    ('Rio Grande', 4),  
    ('Florianópolis', 5),
```

('Joinville', 5),
('Blumenau', 5),
('São José', 5),
('Chapecó', 5),
('Itajaí', 5),
('Criciúma', 5),
('Lages', 5),
('Jaraguá do Sul', 5),
('Palhoça', 5),
('São Paulo', 2),
('Campinas', 2),
('Santos', 2),
('Ribeirão Preto', 2),
('São José dos Campos', 2),
('Sorocaba', 2),
('Bauru', 2),
('Jundiaí', 2),
('Piracicaba', 2),
('Guarulhos', 2),
('Rio de Janeiro', 3),
('Niterói', 3),
('Nova Iguaçu', 3),
('Duque de Caxias', 3),
('São Gonçalo', 3),
('Petrópolis', 3),
('Volta Redonda', 3),
('Macaé', 3),
('Campos', 3),
('Belford Roxo', 3),
('Belo Horizonte', 6),
('Uberlândia', 6),
('Contagem', 6),
('Juiz de Fora', 6),
('Betim', 6),
('Montes Claros', 6),
('Ribeirão das Neves', 6),
('Uberaba', 6),
('Governador Valadares', 6),
('Ipatinga', 6),
('Cuiabá', 7),
('Várzea Grande', 7),
('Rondonópolis', 7),
('Sinop', 7),
('Tangará da Serra', 7),
('Cáceres', 7),
('Sorriso', 7),
('Barra do Garças', 7),
('Lucas do Rio Verde', 7),
('Alta Floresta', 7),

```

('Campo Grande', 8),
('Dourados', 8),
('Três Lagoas', 8),
('Corumbá', 8),
('Ponta Porã', 8),
('Naviraí', 8),
('Nova Andradina', 8),
('Paranaíba', 8),
('Aquidauana', 8),
('Sidrolândia', 8),
('Goiânia', 9),
('Aparecida de Goiânia', 9),
('Anápolis', 9),
('Rio Verde', 9),
('Luziânia', 9),
('Águas Lindas de Goiás', 9),
('Valparaíso de Goiás', 9),
('Trindade', 9),
('Formosa', 9),
('Itumbiara', 9);
  
```

CidadeID	Nome	EstadoID
1	Curitiba	1
2	Londrina	1
3	Maringá	1
4	Cascavel	1
5	Ponta Grossa	1
6	Foz do Iguaçu	1
7	Guarapuava	1
8	Toledo	1
9	Arapongas	1
10	São José dos Pinhais	1
11	Porto Alegre	4
12	Caxias do Sul	4

Figura 09 — Dados inseridos na tabela Cidade, na base de dados.

Como “CidadeID” foi definido como auto incremento, o seu preenchimento foi automático por parte do SQL, não houve a necessidade nem de referenciá-lo na inserção dos dados nas linhas das demais colunas, “Nome” e “EstadoID”. “EstadoID”, por sua vez, é a chave estrangeira, que interconecta e referencia cada cidade ao seu estado pertencente.

3º) DADOS DA TABELA POLO:

```

-- IMPORTAÇÃO DOS DADOS DA TABELA POLO:
SELECT * FROM Polo
  
```

```
INSERT INTO Polo (Nome, CidadeID)
VALUES
('POLO DE CURITIBA 1', 1),
(...)
('POLO DE ITUMBIARA 5', 90)
```

ATENÇÃO: Esse INSERT completo pode ser consultado na query SQL por nome “dados_polos_formatados” localizada na pasta “queries”, dentro do diretório do projeto. Não foi adicionado o código inteiramente aqui a fim de evitar aumentos das páginas do documento, além de ser inviável a visualização, haja vista as repetições dos dados.

4.2. Dados das tabelas de Cursos e Estudantes

Na planilha gerada, evidenciou-se os campos "Total_Estudantes_Ativos", "Total_Estudantes_Formados", "Total_Estudantes_Desistentes", "Total_Estudantes Trancamentos", "Descrição_Curso", "Total_Vagas_Disponíveis", "Preços Cursos" e "Satisfação com o Curso" como sendo os que trazem informações acerca dos estudantes e cursos da instituição. Então, assim sendo, procedeu-se com a inserção dos dados das tabela Curso e Estudante conforme o padrão abaixo:

```
-- IMPORTAÇÃO DOS DADOS DA TABELA CURSO:
SELECT * FROM Curso

INSERT INTO Curso ([Nome], [VagasDisponiveis], [Preco],
[CidadeID])
VALUES
('Análise e Desenvolvimento de Sistemas', 149, 612, 1),
('Gestão Empresarial', 147, 525, 1),
('Pedagogia', 125, 903, 2),
('Direito', 167, 593, 1),
(...)

-- IMPORTAÇÃO DOS DADOS DA TABELA ESTUDANTE:
SELECT * FROM Estudante

INSERT INTO Estudante(Total_Ativos, Total_Formados,
Total_Desistentes, Total_Trancamentos, CursoID, PoloID,
Satisfacao, DataConsiderada)
VALUES
```

```
(52.0, 50.0, 6.0, 5.0, 1.0, 1.0, 1.1, '2024-01-01'),  
(73.0, 7.0, 1.0, 0.0, 2.0, 2.0, 2.3, '2024-06-21'),  
(27.0, 8.0, 8.0, 0.0, 3.0, 3.0, 2.7, '2024-02-12'),  
(...)
```

ATENÇÃO: Esses INSERTs completos podem ser consultados nas queries SQL por nomes “dados_cursos_formatados” e “dados_estudantes_formatados” localizadas na pasta “queries”, dentro do diretório do projeto. Não foi adicionado o código inteiramente aqui a fim de evitar aumentos das páginas do documento, além de ser inviável a visualização, haja vista as repetições dos dados.

4.3. Dados da tabela de Pagamentos

Na planilha gerada, evidenciou-se o campo “Status_Pagamento” como sendo o que traz informações acerca dos status dos pagamentos de seus cursos por parte dos estudantes da instituição, onde existem somente dois estados “Adimplente” ou “Inadimplente”. Então, assim sendo, procedeu-se com a inserção dos dados nas colunas da tabela Pagamento conforme o padrão abaixo:

```
-- IMPORTAÇÃO DOS DADOS DA TABELA PAGAMENTO:  
SELECT * FROM Pagamento  
  
INSERT INTO Pagamento ([EstudanteID], [CursoID],  
[StatusPagamento])  
VALUES  
(1, 1, 'Adimplente'),  
(2, 2, 'Inadimplente'),  
(3, 3, 'Inadimplente'),  
(4, 4, 'Inadimplente'),  
(...)
```

ATENÇÃO: Esse INSERT completo pode ser consultado na query SQL por nome “dados_pagamentos_formatados” localizada na pasta “queries”, dentro do diretório do projeto. Não foi adicionado o código inteiramente aqui a fim de evitar aumentos das páginas do documento, além de ser inviável a visualização, haja vista as repetições dos dados.

E assim, com as importações feitas com estes códigos de INSERT, prossegue-se com a criação das Views, que será de onde o Power BI carregará as informações para a exibição e processamento dos dashboards.

5. ESBOÇO DOS DASHBOARDS DO POWER BI

Com as tabelas criadas e os dados devidamente inseridos nas etapas anteriores, o próximo passo é a conexão com o tal Banco de Dados criado e criar de fato os dashboards, no Power BI. Para a conexão com os dados do BD, comumente, cria-se as chamadas views, que centralizam a busca dos dados e facilitam o processamento dos mesmos no Power BI. Mas, primeiramente, **será apresentado um esboço das ideias de tudo que o Power BI do projeto irá apresentar**, pois é necessário saber as necessidades do mesmo para então proceder-se com a criação das tais views:

DASHBOARD 01: VISÃO GERAL DA INSTITUIÇÃO

Objetivo: Este dashboard proporcionará uma visão consolidada sobre os estudantes, cursos e satisfação geral dentro da instituição. Ele permite monitorar o número de estudantes em diferentes estágios (ativos, formados, desistentes, trancados), a média dos preços dos cursos e o nível de satisfação dos estudantes com os cursos oferecidos.

Elementos do Dashboard:

1. Cards - Nº de Estudantes Ativos, Formados, Desistentes e

Trancamentos:

Estes cards são baseados nos dados da tabela **Estudante**, e calculam a quantidade de estudantes com base nas suas categorias.

- **Ativos:** Estudantes com status ativo (não formados, desistentes ou trancados).
- **Formados:** Estudantes que completaram o curso.
- **Desistentes:** Estudantes que desistiram do curso.

- **Trancamentos:** Estudantes que trancaram o curso.
Cálculo baseado nas colunas Total_Ativos, Total_Formados,
Total_Desistentes, Total_Trancamentos.



Figura 10 — Cards que estarão no dashboard “Visão Geral da Instituição”.

2. Gráfico de Colunas Empilhadas - Média de Preços dos Cursos:

Este gráfico empilhado compara a média dos preços de cursos entre diferentes cursos, representando as diferenças de preço entre eles.

- Eixo X: Nome do Curso.
- Eixo Y: Média do Preço (calculada a partir da coluna Preço na tabela **Curso**).

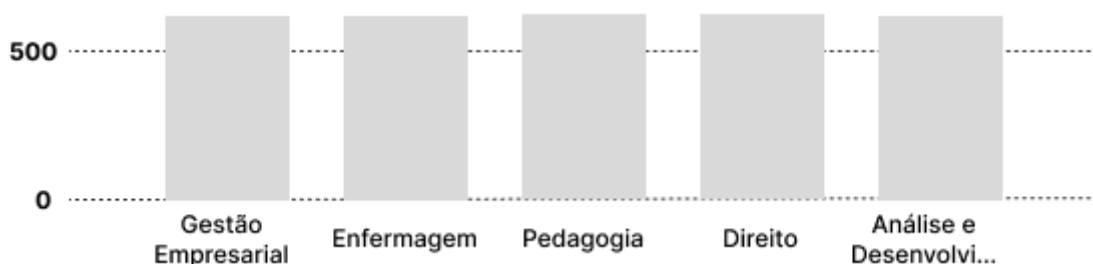


Figura 11 — Gráfico de colunas empilhadas que estará no dashboard “Visão Geral da Instituição”.

3. Tabela - Cursos Disponíveis e Qtd. de Vagas Disponíveis:

A tabela exibe todos os cursos disponíveis na instituição e o número de vagas restantes para cada um, que vem da tabela **Curso**.

- Colunas: Nome do Curso (Cursos Disponíveis), Qtd. de Vagas Disponíveis.

Cursos Disponíveis	Qtd. de Vagas Disponíveis
Análise e Desenvolvimento de Sistemas	
Direito	
Enfermagem	
Gestão Empresarial	
Pedagogia	
Total	

Figura 12 — Tabela que estará no dashboard “Visão Geral da Instituição”.

4. Gráfico de Pizza - Satisfação dos Estudantes com os Cursos:

Este gráfico de pizza exibe a soma das classificações de satisfação dos estudantes com cada curso.

- Legenda: Classificação de Satisfação (ex: 1 a 5).
- Valores: Soma de Satisfação com o Curso (baseado na coluna Satisfacao na tabela **Estudante**).

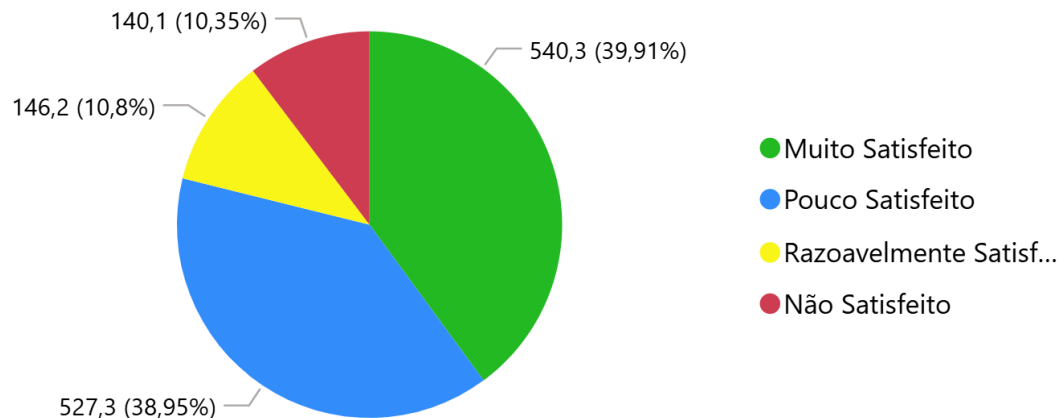


Figura 13 — Gráfico de pizza que estará no dashboard “Visão Geral da Instituição”.

As regras para as 4 classificações dos valores somas — processados da coluna “Satisfacao” da tabela Estudante — são:

- Se Satisfacao ≤ 1.9 → **Não Satisfeito** → Cor vermelha de destaque (#FF0013)
- Se Satisfacao ≤ 3.5 → **Pouco Satisfeito** → Cor azul de destaque (#118DFF)
- Se Satisfacao ≤ 3.9 → **Razoavelmente Satisfeito** → Cor amarela de destaque (#FFD200)
- Se Satisfacao ≤ 5 → **Muito Satisfeito** → Cor verde de destaque (#00FF4D)

5. Gráfico de Colunas Clusterizado - Adimplência e Inadimplência por Curso

Este gráfico de colunas clusterizado exibe a quantidade de adimplentes e inadimplentes por curso.

- Eixo X: Nome do Curso.
- Eixo Y: Soma de Adimplência e Soma de Inadimplência.

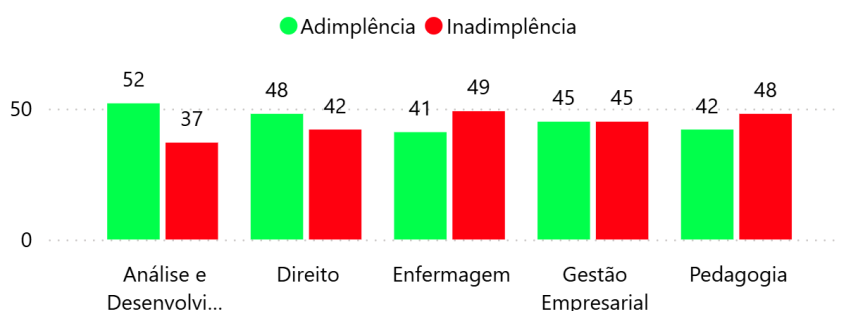


Figura 14 — Gráfico de colunas clusterizado que estará no dashboard “Visão Geral da Instituição”.

02. DASHBOARD VISÃO DOS ESTUDANTES

Objetivo: Este dashboard foca na análise detalhada do perfil dos estudantes, incluindo sua satisfação com os cursos e distribuição por estado (UF), permitindo insights sobre o comportamento dos estudantes.

Elementos do Dashboard:

1. Cards - Nº de Estudantes Ativos, Formados, Desistentes e Trancamentos:

Como no primeiro dashboard, mostra os mesmos indicadores de número de estudantes, calculados a partir das informações na tabela **Estudante**.



Figura 15 — Cards que estarão no dashboard “Visão dos Estudantes”.

2. Gráfico de Colunas Empilhadas - Média de Satisfação com o Curso por UF:

Este gráfico mostra como a média de satisfação varia entre as diferentes UFs.

- Eixo X: UF (proveniente da tabela **Estado**).
- Eixo Y: Média de Satisfação (calculada a partir das classificações de satisfação dos estudantes).

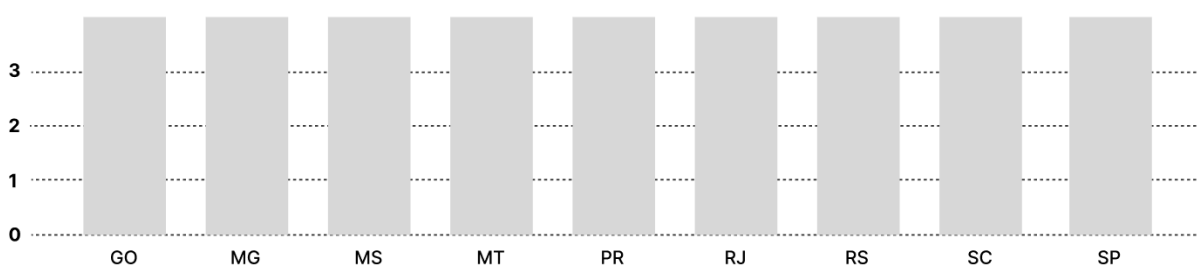


Figura 16 — Gráfico de colunas empilhadas que estará no dashboard “Visão dos Estudantes”.

3. Tabela - Cursos Disponíveis:

Exibe os cursos disponíveis, diretamente da tabela **Curso**.

Cursos Disponíveis
Análise e Desenvolvimento de Sistemas
Direito
Enfermagem
Gestão Empresarial
Pedagogia

Figura 17 — Tabela que estará no dashboard “Visão dos Estudantes”.

4. Gráfico de Pizza - Satisfação dos Estudantes com os Cursos:

O mesmo gráfico de pizza do primeiro dashboard, com foco na satisfação dos estudantes.

- Legenda: Classificação da Satisfação.
- Valores: Soma da Satisfação com o Curso.

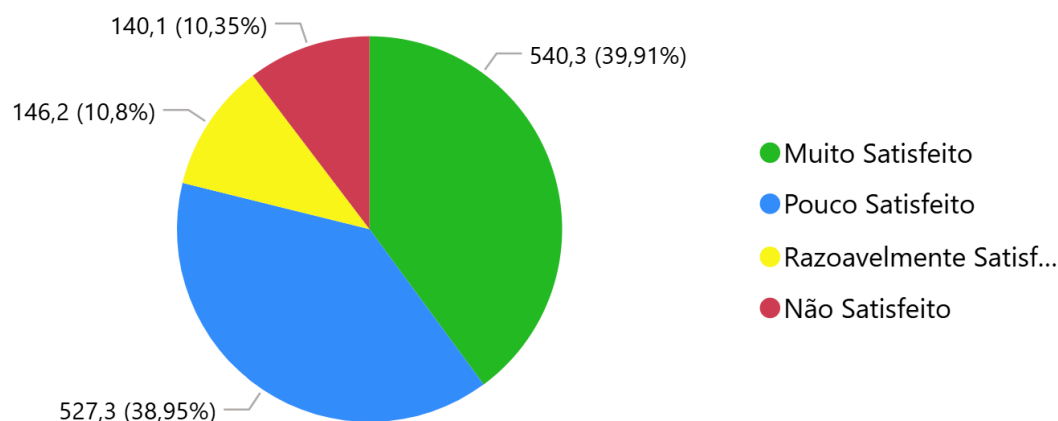


Figura 18 — Gráfico de pizza que estará no dashboard “Visão Geral da Instituição”.

As regras para as 4 classificações dos valores somas — processados da coluna “Satisfacao” da tabela Estudante — são:

- Se Satisfacao ≤ 1.9 → **Não Satisfeito** → Cor vermelha de destaque (#FF0013)
- Se Satisfacao ≤ 3.5 → **Pouco Satisfeito** → Cor azul de destaque (#118DFF)
- Se Satisfacao ≤ 3.9 → **Razoavelmente Satisfeito** → Cor amarela de destaque (#FFD200)
- Se Satisfacao ≤ 5 → **Muito Satisfeito** → Cor verde de destaque (#00FF4D)

5. Gráfico de Linhas - Média de Satisfação com o Curso por Mês:

Este gráfico acompanha a evolução da satisfação dos estudantes ao longo do tempo (por mês).

- Eixo X: Data (mês e ano).
- Eixo Y: Média de Satisfação (calculada a partir da tabela **Estudante**).

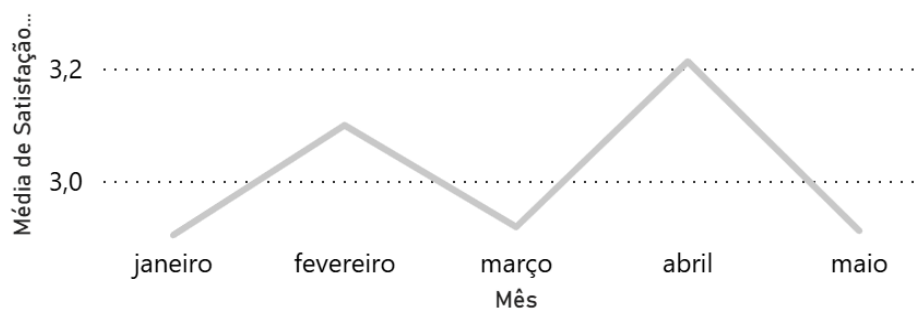


Figura 19 — Gráfico de linha que estará no dashboard “Visão dos Estudantes”.

6. Funil - Percentual de Formados, Desistentes e Trancamentos com relação à Ativação da Matrícula:

Este funil acompanha o percentual de formados, desistentes e trancamentos com relação ao momento em que os estudantes ativam a matrícula na instituição.

- Eixo X: Data (mês e ano).
- Eixo Y: Média de Satisfação (calculada a partir da tabela **Estudante**).

A ideia é somar o percentual de formados, desistentes e trancamentos e dividir pelo percentual de ativos.

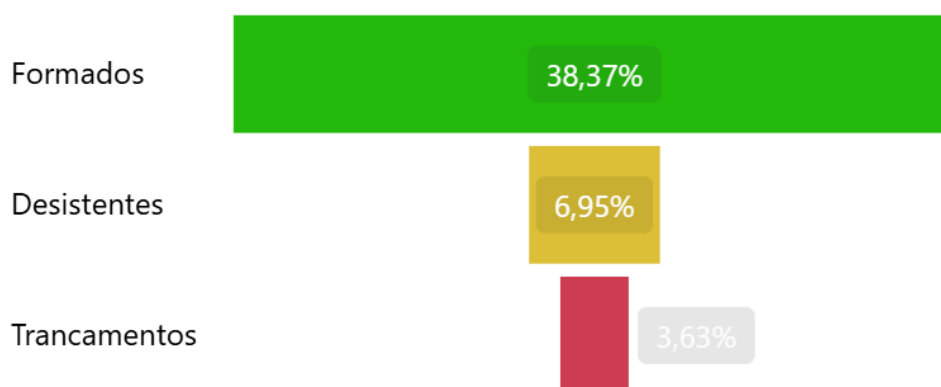


Figura 20 — Funil que estará no dashboard “Visão dos Estudantes”.

03. DASHBOARD VISÃO DOS CURSOS

Objetivo: Foco no desempenho dos cursos oferecidos pela instituição, considerando preço, vagas, e a satisfação dos estudantes, com o objetivo de otimizar a oferta de cursos com base nas variáveis analisadas.

Elementos do Dashboard:

1. **Cards - Total de Cursos, Total de Vagas Disponíveis, Curso Mais Caro, Curso Mais Barato, Curso com Mais Vagas, Curso com Menos Vagas:**
 - **Total de Cursos:** Número total de cursos disponíveis na tabela **Curso**.
 - **Total de Vagas Disponíveis:** Soma de todas as vagas de todos os cursos.
 - **Curso Mais Caro:** Identifica o curso com o maior preço (coluna Preço na tabela **Curso**).

- **Curso Mais Barato:** Identifica o curso com o menor preço (coluna Preço na tabela **Curso**).
- **Curso com Mais Vagas:** Identifica o curso com o maior número de vagas.
- **Curso com Menos Vagas:** Identifica o curso com o menor número de vagas.



Figura 21 — Cards que estarão no dashboard “Visão dos Cursos” (OBS.: No dashboard, ficarão um ao lado do outro, todavia, por questões de espaço da página documento, quebrou-se em duas linhas!).

2. Gráfico de Barras Clusterizado - Média de Vagas Disponíveis por Curso:

Este gráfico mostra a média de vagas disponíveis para cada curso.

- Eixo Y: Nome dos Cursos.
- Eixo X: Total de Vagas Disponíveis.

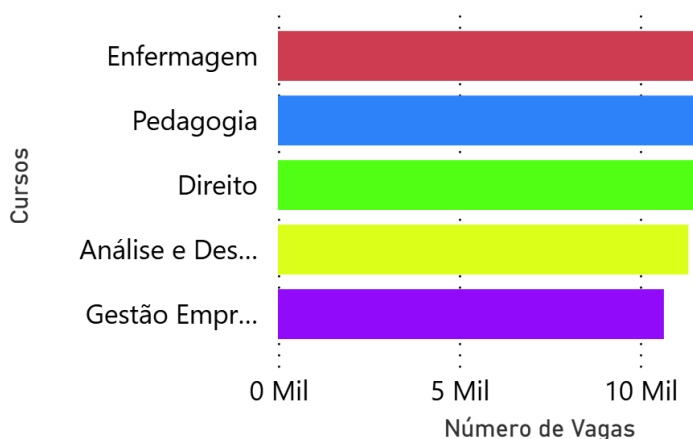


Figura 22 — Gráfico de barras clusterizado que estará no dashboard “Visão dos Cursos”.

3. Gráfico de Colunas Clusterizado - Média de Preços dos Cursos por UF:

Esse gráfico exibe a média de preços de cursos oferecidos por cada UF.

- Eixo X: UF.

- Eixo Y: Média dos Preços dos Cursos.

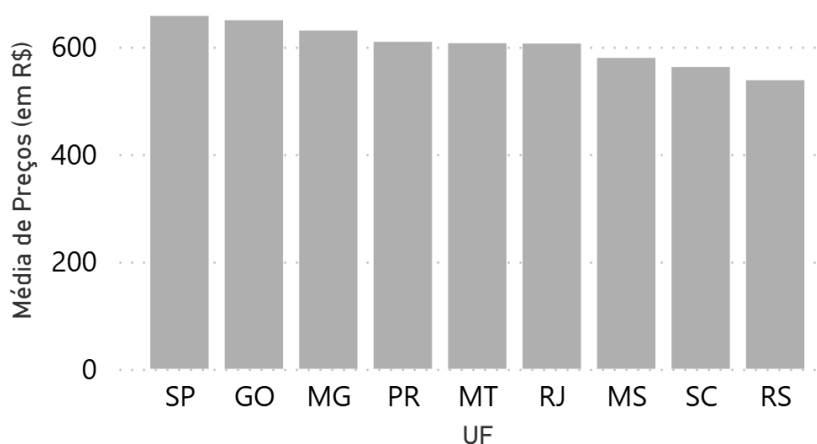


Figura 23 — Gráfico de barras clusterizado que estará no dashboard “Visão dos Cursos”.

4. Gráfico de Linhas - Média de Preços Totais dos Cursos por Mês:

Exibe a evolução do preço médio dos cursos ao longo do tempo.

- Eixo X: Data (mês/ano).
- Eixo Y: Média de Preços dos Cursos.

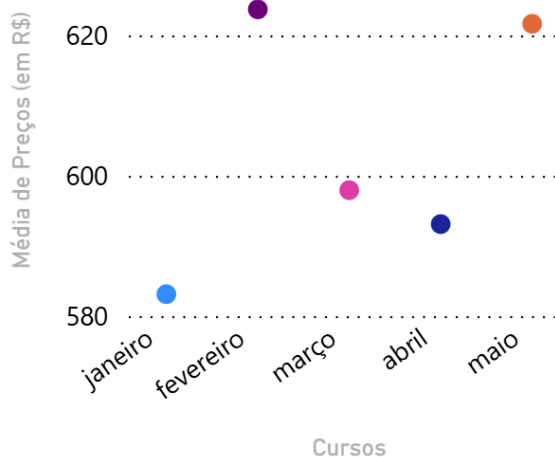


Figura 24 — Gráfico de barras clusterizado que estará no dashboard “Visão dos Cursos”.

5. Tabela - Quantidade de Vagas Disponíveis por Curso e por UF:

Este gráfico mostra a quantidade de vagas disponíveis para os cursos agrupados por UF.

- Linhas: Nome do Curso.
- Colunas: UF.

- Valores: Soma de Vagas Disponíveis

Nome do Curso	GO	MG	MS	MT	PR	RJ	RS	SC	SP	Total
Análise e Desenvolvimento de Sistemas	1034	1220	1268	1147	1368	1244	1219	1477	1329	11306
Direito	1212	1429	1164	1296	1198	1259	1413	1307	1286	11564
Enfermagem	1283	1358	1303	1250	1475	1457	1281	1177	1103	11687
Gestão Empresarial	1221	1317	1144	1059	1312	1296	1029	1153	1095	10626
Pedagogia	1539	1192	1483	1311	1036	1247	1034	1207	1453	11502
Total	6289	6516	6362	6063	6389	6503	5976	6321	6266	56685

Figura 25 — Gráfico de barras clusterizado que estará no dashboard “Visão dos Cursos”.

6. Tabela - Cursos Disponíveis:

Exibe os cursos disponíveis na instituição.

Cursos Disponíveis
Análise e Desenvolvimento de Sistemas
Direito
Enfermagem
Gestão Empresarial
Pedagogia

Figura 26 — Tabela que estará no dashboard “Visão dos Cursos”.

04. DASHBOARD VISÃO FINANCEIRA

Objetivo: Foca na análise da saúde financeira da instituição, mostrando as condições de pagamento dos estudantes, com destaque para a adimplência e inadimplência, e como esses dados variam ao longo do tempo.

Elementos do Dashboard:

1. Cards - Total Adimplentes, Total Inadimplentes, Curso com Maior Adimplência, Curso com Maior Inadimplência:

- **Total Adimplentes:** Número de estudantes com pagamento em dia (baseado na tabela **Pagamento**).
- **Total Inadimplentes:** Número de estudantes com pagamento em atraso.

- **Alta de Adimplência Em:** Identifica o curso com maior número de estudantes adimplentes.
- **Alta de Inadimplência Em:** Identifica o curso com maior número de estudantes inadimplentes.



Figura 27 — Cards que estarão no dashboard “Visão Financeira”.

2. Gráfico de Colunas Clusterizado - Adimplência e Inadimplência por Curso:

Exibe a quantidade de estudantes adimplentes e inadimplentes para cada curso.

- Eixo X: Nome dos Cursos.
- Eixo Y: Total de Adimplentes e Total de Inadimplentes.

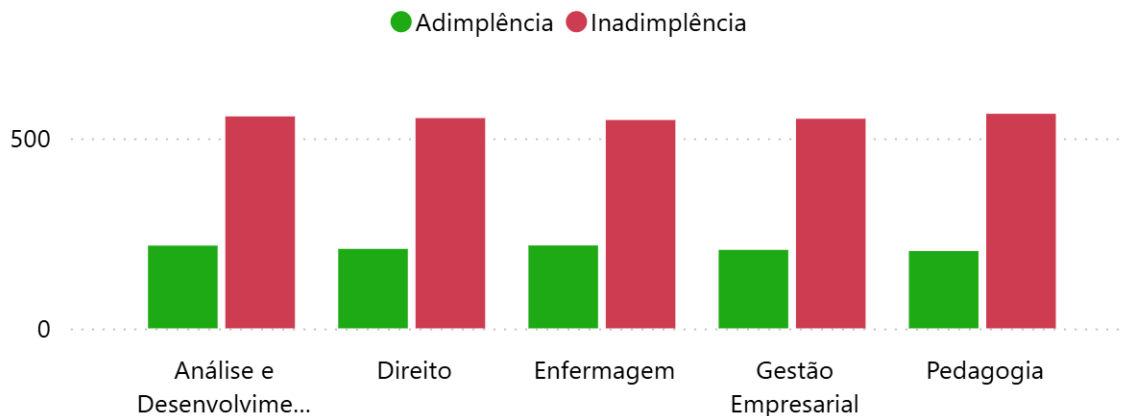


Figura 28 — Gráfico de colunas clusterizado que estará no dashboard “Visão Financeira”.

3. Gráfico de Linhas - Variação de Adimplência e Inadimplência por Curso:

Este gráfico acompanha a variação ao longo do tempo de adimplência e inadimplência.

- Eixo X: Nome dos Cursos.
- Eixo Y (Primário): Adimplência.
- Eixo Y (Secundário): Inadimplência.

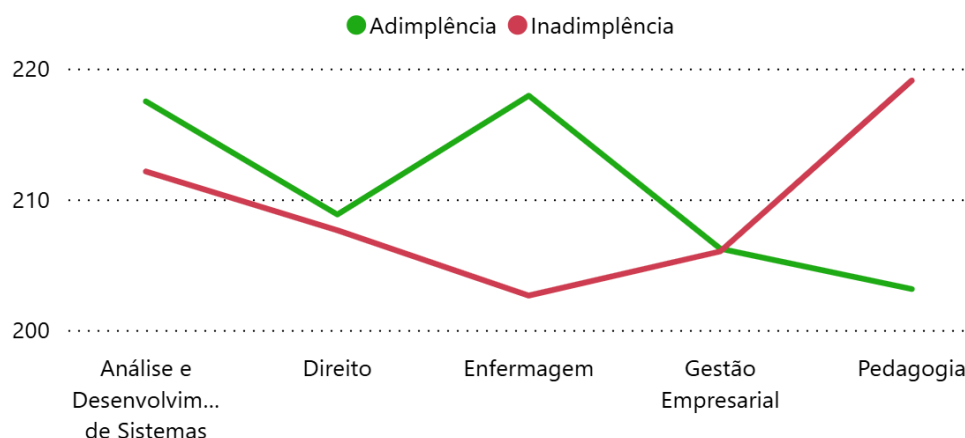


Figura 29 — Gráfico de linhas que estará no dashboard “Visão Financeira”.

4. Gráfico de Área - Variação da Adimplência por Mês:

Acompanha a variação de adimplência ao longo do tempo.

- Eixo X: Data (mês/ano).
- Eixo Y: Percentual de Adimplência.

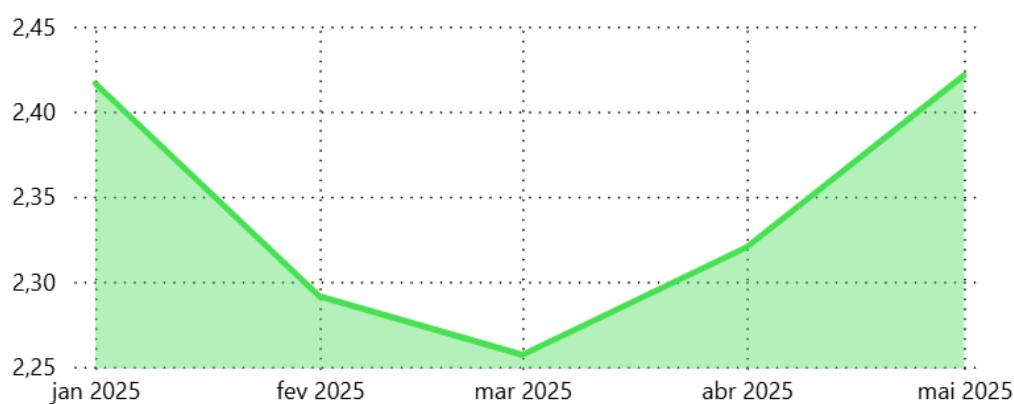


Figura 30 — Gráfico de área que estará no dashboard “Visão Financeira”.

5. Gráfico de Área - Variação da Inadimplência por Mês:

Similar ao gráfico anterior, mas mostrando a variação da inadimplência.

- Eixo X: Data (mês/ano).
- Eixo Y: Percentual de Inadimplência.

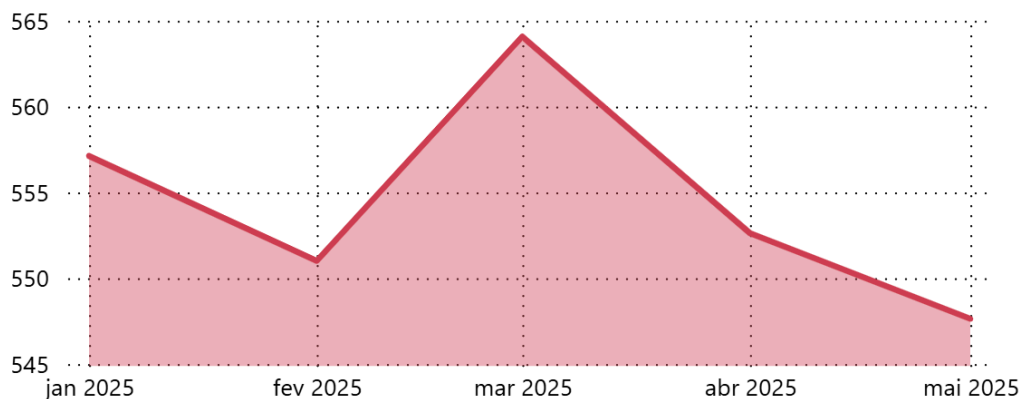


Figura 31 — Gráfico de área que estará no dashboard “Visão Financeira”.

6. CRIAÇÃO DAS VIEWS

Uma view, no contexto de bancos de dados SQL, pode ser definida como uma **tabela virtual**, ou seja, é algo que não existe no BD de fato. Mas como assim? Basicamente, **a ideia é fazer uma consulta personalizada a várias tabelas ao mesmo tempo e reunir as determinadas colunas de dados desejadas dessas tabelas consultadas em uma única tabela**. Por exemplo, aqui no contexto deste projeto: existem as tabelas *Estado*, *Cidade*, *Polo*, *Curso* e *Estudante*, “isoladas” umas das outras, com exceção das chaves estrangeiras (FKs) — o termo “isolado” foi empregado no sentido de dizer que as tabelas não conseguem enxergar as colunas umas das outras, somente partilham dados entre si quando existem FKs. Pode-se fazer uma view que engloba os seguintes dados em uma única tabela: “Data Considerada”, “Nome do Curso”, “Quantidade de Vagas Disponíveis”, “Polo”, “Cidade” e “UF”.

Nesse contexto, criou-se **4 views** que irão organizar o Banco de Dados de modo a facilitar a criação de 4 páginas de dashboards no Power BI.

6.1. View Resumo Quantidade Estudantes

6.1.1. Objetivo

Exibir o total de estudantes ativos, formados, desistentes e trancados por polo, cidade, estado e UF.

6.1.2. Organização

- Da tabela **Estudante**, extrair: **Total_Ativos**, **Total_Formados**, **Total_Desistentes**, **Total_Trancamentos**;
- Da tabela **Curso**, extrair: **Nome**;
- Da tabela **Polo**, extrair: **Nome**;
- Da tabela **Cidade**, extrair: **Nome**;
- Da tabela **Estado**, extrair: **Nome**, **UF**;

6.1.3. Rascunho do código

```
CREATE VIEW vw_ResumoQuantidadeEstudantes AS
SELECT
    Estudante.Total_Ativos, Estudante.Total_Formados, Estudante.Total_Desistentes,
    Estudante.Total_Trancamentos, Polo.Nome, Cidade.Nome, Estado.Nome, Estado.UF
FROM Estudante
    INNER JOIN Curso ON Curso.CursoID = Estudante.CursoID
    INNER JOIN Polo ON Polo.PoloID = Estudante.PoloID
    INNER JOIN Cidade ON Polo.CidadeID = Cidade.CidadeID
    INNER JOIN Estado ON Estado.EstadoID = Cidade.EstadoID;
```

6.1.4. Explicação narrativa do código

```
CRIE A VIEW vw_ResumoQuantidadeEstudantes COMO
SELECIONE
    Total_Ativos da tabela Estudante, Total_Formados da tabela Estudante,
    Total_Desistentes da tabela Estudante, Total_Trancamentos da tabela Estudante,
    Nome da tabela Polo,
    Nome da tabela Polo, Nome da tabela Cidade, Nome da tabela Estado, UF da tabela
    Estado
DA TABELA Estudante
    FAÇA JUNÇÃO COM Curso ONDE CursoID da tabela Curso SEJA IGUAL a CursoID da
    tabela Estudante
    FAÇA JUNÇÃO COM Polo ONDE PoloID da tabela Polo SEJA IGUAL a PoloID da tabela
    Estudante
    FAÇA JUNÇÃO COM Cidade ONDE CidadeID da tabela Polo SEJA IGUAL a CidadeID
    da tabela Cidade
    FAÇA JUNÇÃO COM Estado ONDE EstadoID da tabela Estado SEJA IGUAL a EstadoID
    da tabela Cidade
```

6.1.5. Código definitivo

```
CREATE VIEW vw_ResumoQuantidadeEstudantes AS
SELECT
    E.Total_Ativos, E.Total_Formados, E.Total_Desistentes, E.Total_Trancamentos,
    P.Nome AS 'Polo Considerado',
    Ci.Nome AS 'Cidade Considerada',
    Es.Nome AS 'Estado Considerado',
    Es.UF AS 'UF Respectiva'
FROM Estudante E
    INNER JOIN Curso C ON C.CursoID = E.CursoID
    INNER JOIN Polo P ON P.PoloID = E.PoloID
    INNER JOIN Cidade Ci ON P.CidadeID = Ci.CidadeID
    INNER JOIN Estado Es ON Es.EstadoID = Ci.EstadoID;
```

6.2. View Satisfação dos Estudantes

6.2.1. Objetivo

Exibir a satisfação dos estudantes por polo, cidade, estado e UF.

6.2.2. Organização

- Da tabela **Estudante**, extrair: **DataConsiderada**, **Satisfação**;
- Da tabela **Curso**, extrair: **Nome**;
- Da tabela **Polo**, extrair: **Nome**;
- Da tabela **Cidade**, extrair: **Nome**;
- Da tabela **Estado**, extrair: **Nome**, **UF**;

6.2.3. Rascunho do código

```
CREATE VIEW vw_SatisfacaoDosEstudantes AS
SELECT
    Estudante.DataConsiderada, Estudante.Satisfacao,
    Curso.Nome, Polo.Nome, Cidade.Nome, Estado.Nome, Estado.UF
FROM Estudante
    INNER JOIN Curso ON Curso.CursoID = Estudante.CursoID
    INNER JOIN Polo ON Polo.PoloID = Estudante.PoloID
    INNER JOIN Cidade ON Polo.CidadeID = Cidade.CidadeID
    INNER JOIN Estado ON Estado.EstadoID = Cidade.EstadoID;
```

6.2.4. Explicação narrativa do código

CRIE A VIEW vw_SatisfacaoEstudantes COMO SELECCIONE

DataConsiderada da tabela Estudante, Satisfacao da tabela Estudante, Nome da tabela Curso, Nome da tabela Polo, Nome da tabela Cidade, Nome da tabela Estado, UF da tabela Estado

DA TABELA Estudante

FAÇA JUNÇÃO COM Curso ONDE CursoID da tabela Curso SEJA IGUAL a CursoID da tabela Estudante

FAÇA JUNÇÃO COM Polo ONDE PoloID da tabela Polo SEJA IGUAL a PoloID da tabela Estudante

FAÇA JUNÇÃO COM Cidade ONDE CidadeID da tabela Polo SEJA IGUAL a CidadeID da tabela Cidade

FAÇA JUNÇÃO COM Estado ONDE EstadoID da tabela Estado SEJA IGUAL a EstadoID da tabela Cidade

6.2.5. Código definitivo

CREATE VIEW vw_SatisfacaoDosEstudantes AS SELECT

E.DataConsiderada AS 'Data Considerada',
E.Satisfacao AS 'Satisfação',
C.Nome AS 'Nome do Curso',
P.Nome AS 'Polo Considerado',
Ci.Nome AS 'Cidade Considerada',
Es.Nome AS 'Estado Considerado',
Es.UF AS 'UF Respectiva'

FROM Estudante E

INNER JOIN Curso C ON C.CursoID = E.CursoID
INNER JOIN Polo P ON P.PoloID = E.PoloID
INNER JOIN Cidade Ci ON P.CidadeID = Ci.CidadeID
INNER JOIN Estado Es ON Es.EstadoID = Ci.EstadoID;

6.3. View Resumo dos Cursos

6.3.1. Objetivo

Exibir a média de preço dos cursos por polo, cidade, estado e UF.

6.3.2. Organização

- Da tabela **Estudante**, extrair: **DataConsiderada**;
- Da tabela **Curso**, extrair: **Nome**, **Preço**, **VagasDisponiveis**;

- Da tabela **Polo**, extrair: **Nome**;
- Da tabela **Cidade**, extrair: **Nome**;
- Da tabela **Estado**, extrair: **Nome**, **UF**;

6.3.3. Rascunho do código

```
CREATE VIEW vw_ResumoDosCursos AS
SELECT
    Estudante.DataConsiderada, Curso.Nome, Curso.Preco, Curso.VagasDisponiveis,
    Polo.Nome, Cidade.Nome, Estado.Nome, Estado.UF
FROM Estudante
    INNER JOIN Curso ON Curso.CursoID = Estudante.CursoID
    INNER JOIN Polo ON Polo.PoloID = Estudante.PoloID
    INNER JOIN Cidade ON Polo.CidadeID = Cidade.CidadeID
    INNER JOIN Estado ON Estado.EstadoID = Cidade.EstadoID;
```

6.3.4. Explicação narrativa do código

```
CRIE A VIEW vw_ResumoDosCursos COMO
SELECIONE
    DataConsiderada da tabela Estudante, Nome da tabela Curso, Preco da tabela
    Curso, VagasDisponiveis da tabela Curso, Nome da tabela Polo, Nome da tabela
    Cidade, Nome da tabela Estado, UF da tabela Estado
DA TABELA Estudante
    FAÇA JUNÇÃO COM Curso ONDE CursoID da tabela Curso SEJA IGUAL a CursoID da
    tabela Estudante
    FAÇA JUNÇÃO COM Polo ONDE PoloID da tabela Polo SEJA IGUAL a PoloID da tabela
    Estudante
    FAÇA JUNÇÃO COM Cidade ONDE CidadeID da tabela Polo SEJA IGUAL a CidadeID
    da tabela Cidade
    FAÇA JUNÇÃO COM Estado ONDE EstadoID da tabela Estado SEJA IGUAL a EstadoID
    da tabela Cidade
```

6.3.5. Código definitivo

```
CREATE VIEW vw_ResumoDosCursos AS
SELECT
    E.DataConsiderada AS 'Data Considerada',
    C.Nome AS 'Nome do Curso', C.Preco 'Preço', C.VagasDisponiveis AS 'Vagas
    Disponíveis',
```



```
P.Nome AS 'Polo Considerado',  
Ci.Nome AS 'Cidade Considerada',  
Es.Nome AS 'Estado Considerado',  
Es.UF AS 'UF Respectiva'
```

```
FROM Estudante E  
INNER JOIN Curso C ON C.CursoID = E.CursoID  
INNER JOIN Polo P ON P.PoloID = E.PoloID  
INNER JOIN Cidade Ci ON P.CidadeID = Ci.CidadeID  
INNER JOIN Estado Es ON Es.EstadoID = Ci.EstadoID;
```

6.4. View Resumo Financeiro

6.4.1. Objetivo

Exibir o status de pagamento, considerando data, nome do curso, polo, cidade, estado e UF.

6.4.2. Organização

- Da tabela **Estudante**, extrair: **DataConsiderada**;
- Da tabela **Curso**, extrair: **Nome**;
- Da tabela **Polo**, extrair: **Nome**;
- Da tabela **Cidade**, extrair: **Nome**;
- Da tabela **Estado**, extrair: **Nome**, **UF**;

6.4.3. Rascunho do código

```
CREATE VIEW vw_ResumoFinanceiro AS  
SELECT  
    Estudante.DataConsiderada, Curso.Nome, Polo.Nome, Cidade.Nome,  
    Estado.Nome, Estado.UF, SUM(CASE WHEN Pagamento.StatusPagamento =  
    'Adimplente' THEN 1 ELSE 0 END) AS 'Adimplente', SUM(CASE WHEN  
    Pagamento.StatusPagamento = 'Inadimplente' THEN 1 ELSE 0 END) AS  
    'Inadimplente'  
FROM Estudante  
    INNER JOIN Pagamento ON Estudante.EstudanteID = Pagamento.EstudanteID  
    INNER JOIN Curso ON Curso.CursoID = Estudante.CursoID  
    INNER JOIN Polo ON Polo.PoloID = Estudante.PoloID  
    INNER JOIN Cidade ON Polo.CidadeID = Cidade.CidadeID  
    INNER JOIN Estado ON Estado.EstadoID = Cidade.EstadoID;  
GROUP BY
```

Estudante.DataConsiderada, Curso.Nome, Polo.Nome, Cidade.Nome,
Estado.Nome, Estado.UF

6.4.4. Explicação narrativa do código

CRIE A VIEW vw_ResumoFinanceiro COMO

SELECIONE

DataConsiderada da tabela Estudante, Nome da tabela Curso, Nome da tabela Polo, Nome da tabela Cidade, Nome da tabela Estado, UF da tabela Estado, SOME(Caso StatusPagamento da tabela Pagamento seja 'Adimplente' ENTÃO coloque 1 CASO CONTRÁRIO coloque 0) é 'Adimplente', SOME(Caso StatusPagamento da tabela Pagamento seja 'Inadimplente' ENTÃO coloque 1 CASO CONTRÁRIO coloque 0) é 'Inadimplente'

DA TABELA Estudante

FAÇA JUNÇÃO COM Pagamento ONDE EstudanteID da tabela Estudante SEJA IGUAL a EstudanteID da tabela Pagamento

FAÇA JUNÇÃO COM Curso ONDE CursoID da tabela Curso SEJA IGUAL a CursoID da tabela Estudante

FAÇA JUNÇÃO COM Polo ONDE PoloID da tabela Polo SEJA IGUAL a PoloID da tabela Estudante

FAÇA JUNÇÃO COM Cidade ONDE CidadeID da tabela Polo SEJA IGUAL a CidadeID da tabela Cidade

FAÇA JUNÇÃO COM Estado ONDE EstadoID da tabela Estado SEJA IGUAL a EstadoID da tabela Cidade

AGRUPE POR

DataConsiderada da tabela Estudante, Nome da tabela Curso, Nome da tabela Polo, Nome da tabela Cidade, Nome da tabela Estado, Nome da tabela UF

6.4.5. Código definitivo

CREATE VIEW vw_ResumoFinanceiro AS

SELECT

E.DataConsiderada AS 'Data Considerada',
C.Nome AS 'Nome do Curso', C.Preco 'Preço', C.VagasDisponiveis AS 'Vagas Disponíveis',
P.Nome AS 'Polo Considerado',
Ci.Nome AS 'Cidade Considerada',
Es.Nome AS 'Estado Considerado',
Es.UF AS 'UF Respectiva'

FROM Estudante E

INNER JOIN Curso C ON C.CursoID = E.CursoID

INNER JOIN Polo P ON P.PoloID = E.PoloID

INNER JOIN Cidade Ci ON P.CidadeID = Ci.CidadeID

INNER JOIN Estado Es ON Es.EstadoID = Ci.EstadoID

GROUP BY

E.DataConsiderada, C.Nome, P.Nome, Ci.Nome, Es.Nome, Es.UF;

ATENÇÃO: A query com os códigos executados no banco referente à criação de cada um das views encontra-se na pasta do projeto, em “Queries”.

7. CRIAÇÃO DO DESIGN DO POWER BI

No quesito estilização, o Power BI deste projeto terá um design simples mas bastante moderno. Primeiramente, ressalta-se o padrão que os 4 dashboards seguirão:

- 1 menu lateral (navbar) com botões de navegação entre os dashboards, onde tais botões terão ícones remetentes ao conteúdo do dashboard que irão carregar, e, na parte superior, uma logo da instituição, que será simplesmente “IF”, acrônimo de “Instituição Fictícia”;
- Ainda no menu lateral, em sua parte inferior, haverá um botão especial que, quando clicado, carregará uma janela de filtros que poderão ser aplicados nos dashboards. Nessa tal janela de filtros, haverá um botão de fechar (ícone de “X”) que permitirá esconder a janela e mostrar os dashboards inteiros novamente;
- 1 fundo escurecido;
- Na parte superior, haverá, em destaque, o título do conteúdo do dashboard, bem como em cada elemento haverá um título indicativo que remeterá aquilo que exibe;



Figura 32 — Design padrão de todos os dashboards do projeto.

No quesito ferramenta escolhida, utilizou-se o **Figma**, que é uma ferramenta própria para o desenho e a construção de designs UX/UI de softwares profissionais.

A seguir, será discorrido detalhadamente a respeito da criação do design dos dashboards.

7.1. Menu lateral

Em uma camada acima do background preto aplicado, adicionou-se um retângulo com os cantos arredondados, cuja altura é 684 px e largura 76 px. Esse retângulo será o menu lateral (navbar) que permitirá a navegação e alternância rápida e intuitiva por entre os 4 dashboards e, também, abrir a janela de aplicação de filtros.

Sobre o retângulo com os cantos arredondados criado, adicionou-se:

- Na parte superior, a logo “IF”, acrônimo de “Instituição Fictícia”;
- Abaixo da logo, adicionou-se cinco retângulos de altura 39 px por largura de 62 px, onde um deles ficou na parte inferior do retângulo arredondado maior que delimita o tamanho do menu;
- Dentro desses retângulos arredondados pequenos, adicionou-se: **um ícone de casa**, que, quando clicado no BI, abrirá o dashboard Visão Geral da

Instituição; **um ícone de grupo de pessoas**, que, quando clicado no BI, abrirá o dashboard Visão dos Estudantes; **um ícone de livro**, que, quando clicado, abrirá o dashboard Visão dos Cursos; **um ícone de cifrão**, que, quando clicado, abrirá o dashboard Visão Financeira; e, por fim, no último retângulo arredondado que fica na parte inferior, **um ícone de filtro**, que, quando clicado no BI, abrirá a janela de filtros;

- Por fim, para uma estilização moderna e tornar o menu mais visível, adicionou-se, no retângulo de cantos arredondados maior, uma sombra projetada, com estas definições abaixo:

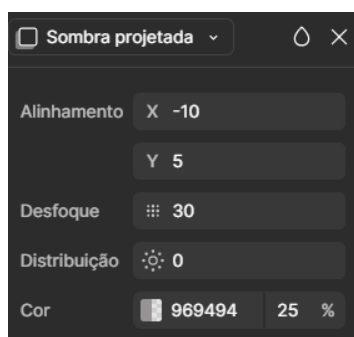


Figura 33 — Estilização de sombra do menu lateral.

E assim, o resultado do menu lateral foi este abaixo:



Figura 34 — Menu lateral de navegação entre os dashboards.

7.2. Dashboard Visão Geral da Instituição

A inserção dos elementos em cada dashboard é feita no Power BI, todavia o local onde tais elementos ficarão podem ser estilizados e delimitados no momento em que se está criando o design dos dashboards. Sendo assim, procedeu-se com a criação do dashboard Visão Geral da Instituição da seguinte forma:

- Na parte superior, adicionou-se o título “Visão Geral da Instituição”, em cor branca e fonte Inter de tamanho 20 com efeito Bold (Negrito);
- Logo abaixo, para fazer uma divisão e organizar o visual do dashboard, inseriu-se uma linha de cor cinza (#707070);
- Adicionou-se 3 retângulos de cantos arredondados de altura 61 px por largura 118 px para os cards de “Ativos”, “Formados”, “Desistentes” e 1 retângulo de altura 61 px por largura de 137 px para “Trancamentos”. Aplicou-se a esses quatro retângulos pequenos a seguinte estilização de sombra projetada:

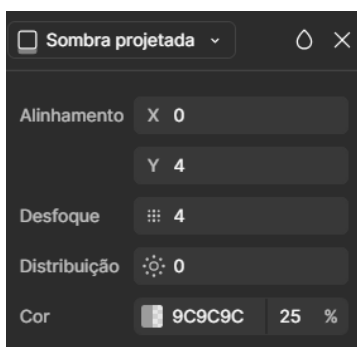


Figura 35 — Estilização de sombra dos retângulos pequenos que terão os 4 cards no BI.

- Na parte interna dos cards, adicionou-se um retângulo de cantos arredondados sem preenchimento, colocando somente um traçado de cor cinza (#6C6C6C), de tamanho 0.5 px, para um efeito moderno;
- Por fim, adicionou-se mais quatro retângulos de cantos arredondados que irão dispor os conteúdos: “Média de Preço dos Cursos”, “Cursos e Quantidade de Vagas Disponíveis”, “Nível de Satisfação por Curso” e “Status de Pagamentos por Estudante”;
- Também foi incluída mais uma linha em cor cinza que faz uma divisão e possibilita uma organização visual do dashboard;
- Na parte superior do background, de modo a ter-se um pequeno tom artístico, adicionou-se três linhas com cinzas #353535, #262626, e #313131;

- No menu lateral, no botão com ícone de casa, adicionou-se um traçado de 1 px de cor cinza (#B0B0B0). Com isso, a ideia é enfatizar que o dashboard “Visão Geral da Instituição” é que está aberto;

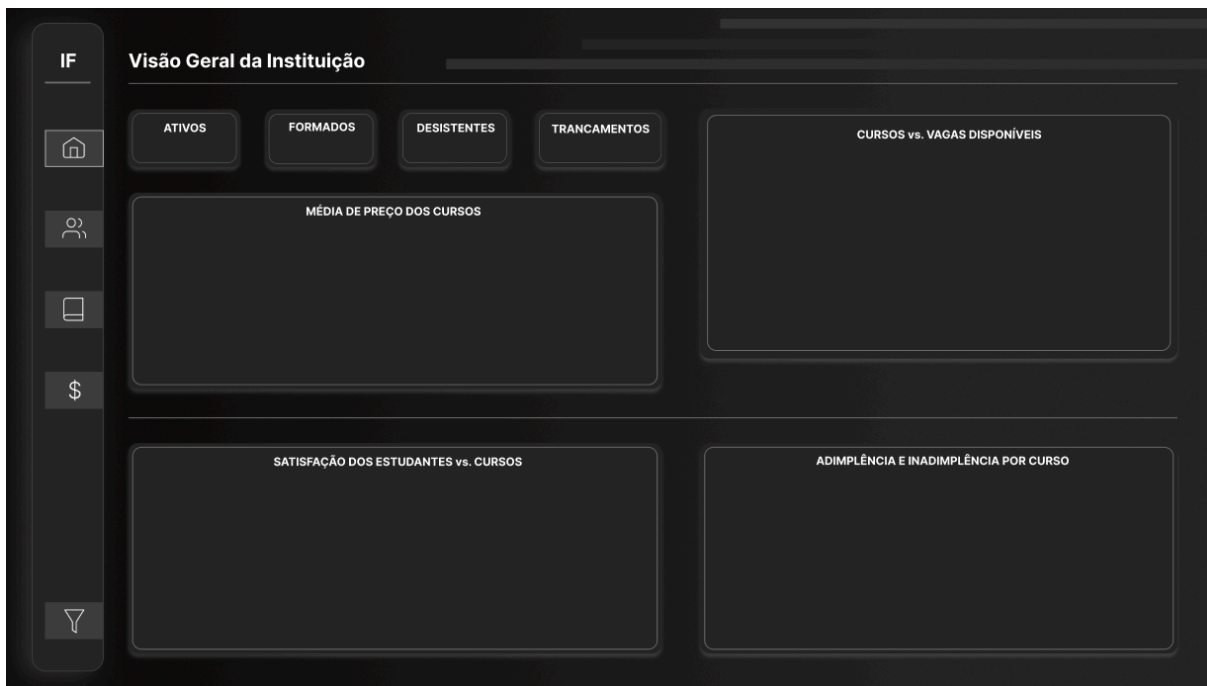


Figura 36 — Design do dashboard Visão Geral da Instituição.

Salvou-se o ficheiro do design com o nome “01. Visão Geral da Instituição” e com a extensão SVG, na pasta “Backgrounds”, nos diretórios do projeto.

7.3. Dashboard Visão dos Estudantes

- Na parte superior, adicionou-se o título “Visão dos Estudantes”, em cor branca e fonte Inter de tamanho 20 com efeito Bold (Negrito);
- Logo abaixo, para fazer uma divisão e organizar o visual do dashboard, inseriu-se uma linha de cor cinza (#707070);
- Tal como no dashboard Visão Geral da Instituição, adicionou-se 3 retângulos de cantos arredondados de altura 61 px por largura 118 px para os cards de “Ativos”, “Formados”, “Desistentes” e 1 retângulo de altura 61 px por largura de 137 px para “Trancamentos”. Aplicou-se a esses quatro retângulos pequenos a seguinte estilização de sombra projetada:

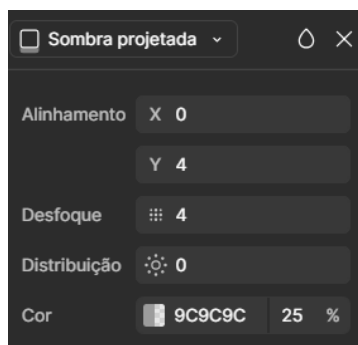


Figura 37 — Estilização de sombra dos retângulos pequenos que terão os 4 cards no BI.

- Adicionou-se 6 retângulos de cantos arredondados, com a mesma estilização dos cards: 1 para Média de Satisfação com o Curso por UF (Largura 562 px / Altura = 222 px), 1 para Média de Satisfação com o Curso por Mês (Largura 562 px / Altura = 232 px), 1 para Satisfação dos Estudantes vs. Cursos (Largura 511 px / Altura = 188 px), 1 para Percentual de Formados, Desistentes e Trancamentos com Relação à Ativação da Matrícula (Largura 511 px / Altura = 186 px), 1 para Cursos Disponíveis (Largura 251 px / Altura = 147 px) e 1 para Média Geral de Satisfação (Largura 135 px / Altura = 76 px);
- Uma consideração: no retângulo de Cursos Disponíveis não adicionou-se um título, pois no Power BI, adicionando a tabela, obrigatoriamente, terá o cabeçalho (não é possível remover) e, assim, o título será aproveitado diretamente lá;
- Na parte interna dos cards, adicionou-se um retângulo de cantos arredondados sem preenchimento, colocando somente um traçado de cor cinza (#6C6C6C), de tamanho 0.5 px, para um efeito moderno;
- Abaixo do retângulo de Média Geral de Satisfação, adicionou-se um pequeno box com as REGRAS DA SATISFAÇÃO. A ideia é que conforme seja o valor calculado de satisfação geral da instituição, o mesmo tenha suas cores alteradas, para condizer com os 4 status “Não Satisfeito”, “Pouco Satisfeito”, “Razoavelmente Satisfeito” ou “Muito Satisfeito”;
- Na parte superior do background, de modo a ter-se um pequeno tom artístico, adicionou-se três linhas com cinzas #353535, #262626, e #313131;
- No menu lateral, no botão com ícone de pessoas, adicionou-se um traçado de 1 px de cor cinza (#B0B0B0). Com isso, a ideia é enfatizar que o dashboard “Visão dos Estudantes” é que está aberto;

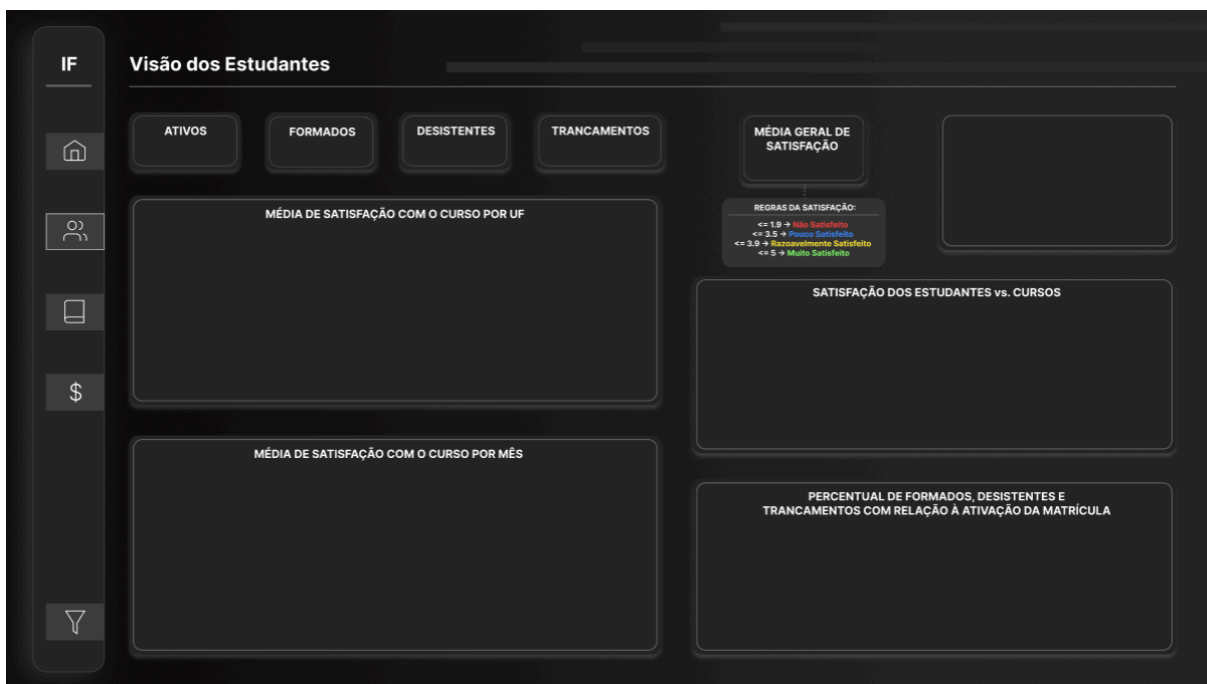


Figura 38 — Design do dashboard Visão dos Estudantes.

Salvou-se o ficheiro do design com o nome “02. Visão dos Estudantes” e com a extensão SVG, na pasta “Backgrounds”, nos diretórios do projeto.

7.4. Dashboard Visão dos Cursos

- Na parte superior, adicionou-se o título “Visão dos Cursos”, em cor branca e fonte Inter de tamanho 20 com efeito Bold (Negrito);
- Logo abaixo, para fazer uma divisão e organizar o visual do dashboard, inseriu-se uma linha de cor cinza (#707070);
- No mesmo padrão dos cards do dashboard Visão dos Estudantes, criou-se, desta vez, 4 cards que para a exibição do Total de Cursos, Total de Vagas, Nome do Curso Mais Caro e Nome do Curso Mais Barato de largura 146 px por altura 61 px, e mais 2 cards para a exibição do Nome do Curso com Mais Vagas e o Nome do Curso com Menos Vagas, respectivamente, um de largura 190 px por altura 61 px e outro de largura 205 px por altura 61 px;
- Adicionou-se 5 retângulos de cantos arredondados: 1 para Média de Vagas Disponíveis por Curso (Largura 389 px / Altura 222 px), 1 para Média de Preço dos Cursos por UF (Largura 389 px / Altura 222 px), 1 para Cursos Disponíveis (Largura = 251 px / Altura = 147 px), 1 para Média de Preços

Totais dos Cursos por Ano (Largura = 562 px / Altura = 232 px) e 1 para Quantidade de Vagas Disponíveis por Curso e por UF (Largura = 511 px / Altura = 232 px);

- Uma consideração: no retângulo de Cursos Disponíveis não adicionou-se um título, pois no Power BI, adicionando a tabela, obrigatoriamente, terá o cabeçalho (não é possível remover) e, assim, o título será aproveitado diretamente lá;
- Na parte interna dos cards, adicionou-se um retângulo de cantos arredondados sem preenchimento, colocando somente um traçado de cor cinza (#6C6C6C), de tamanho 0.5 px, para um efeito moderno;
- Na parte superior do background, de modo a ter-se um pequeno tom artístico, adicionou-se três linhas com cinzas #353535, #262626, e #313131;
- No menu lateral, no botão com ícone de livro, adicionou-se um traçado de 1 px de cor cinza (#B0B0B0). Com isso, a ideia é enfatizar que o dashboard “Visão dos Cursos” é que está aberto;

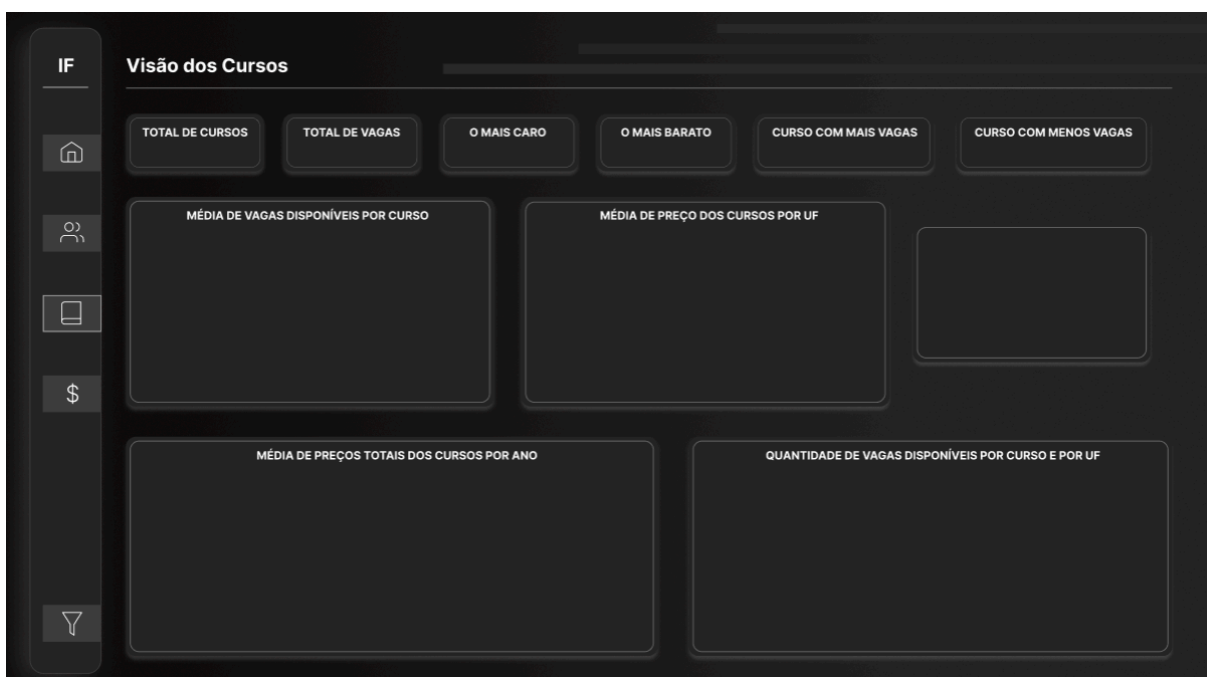


Figura 39 — Design do dashboard Visão dos Cursos.

Salvou-se o ficheiro do design com o nome “03. Visão dos Cursos” e com a extensão SVG, na pasta “Backgrounds”, nos diretórios do projeto.

7.5. Dashboard Visão Financeira

- Na parte superior, adicionou-se o título “Visão Financeira”, em cor branca e fonte Inter de tamanho 20 com efeito Bold (Negrito);
- Logo abaixo, para fazer uma divisão e organizar o visual do dashboard, inseriu-se uma linha de cor cinza (#707070);
- No mesmo padrão dos cards dos outros dashboards, criou-se, desta vez, 2 cards para a exibição do número total de Adimplentes e de Inadimplentes, de largura de 146 px por altura de 61 px, e 2 para a exibição do nome do curso com a adimplência e inadimplência em alta, de largura de 199 px por altura de 61 px;
- Adicionou-se também 4 retângulos de cantos arredondados: 1 para Adimplência e Inadimplência por Curso (Largura = 562 px / Altura = 222 px), 1 para Variação de Adimplência e Inadimplência por Curso (Largura = 562 px / Altura = 222 px), 1 para Variação da Adimplência por Mês (Largura = 511 px / Altura = 222 px) e 1 para Variação da Inadimplência por Mês (Largura = 511 px / Altura = 222 px);
- Na parte interna dos cards, adicionou-se um retângulo de cantos arredondados sem preenchimento, colocando somente um traçado de cor cinza (#6C6C6C), de tamanho 0.5 px, para um efeito moderno;
- Na parte superior do background, de modo a ter-se um pequeno tom artístico, adicionou-se três linhas com cinzas #353535, #262626, e #313131;
- No menu lateral, no botão com ícone de cifrão, adicionou-se um traçado de 1 px de cor cinza (#B0B0B0). Com isso, a ideia é enfatizar que o dashboard “Visão Financeira” é que está aberto;

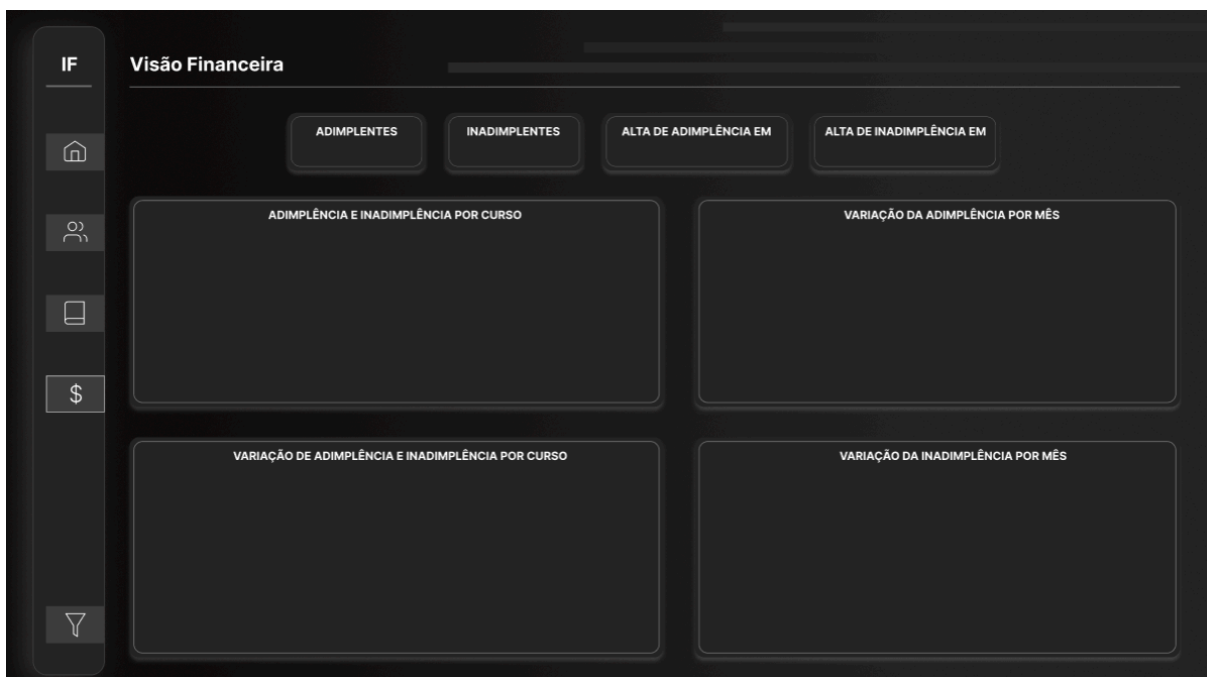


Figura 40 — Design do dashboard Visão Financeira.

Salvou-se o ficheiro do design com o nome “04. Visão Financeira” e com a extensão SVG, na pasta “Backgrounds”, nos diretórios do projeto.

8. CONSTRUÇÃO DOS DASHBOARDS

Agora, será percorrido a respeito da construção dos dashboards no Power BI. Primeiramente, criou-se um novo projeto na opção “Relatório em branco”:



Figura 41 — Menu de criação de um relatório em branco, no Power BI.

Com o relatório já aberto, o primeiro passo da construção foi criar as quatro páginas que irão conter os quatro dashboards.

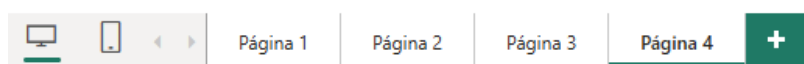


Figura 42 — As quatro páginas criadas no projeto.

E renomeou-se essas quatro páginas para os nomes dos dashboards:

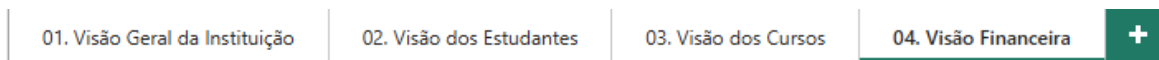


Figura 43 — As quatro páginas renomeadas para os nomes dos dashboards no projeto.

Em Formato da página > Tela de fundo, em cada uma das quatro páginas dos dashboards, inseriu-se os designs de fundo desenvolvidos no PowerPoint:

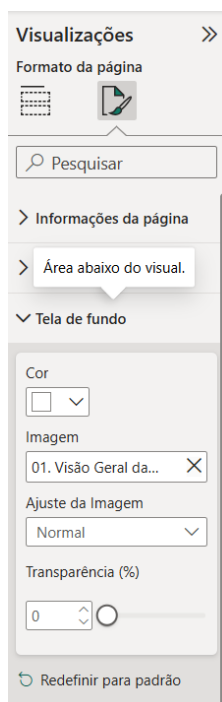


Figura 44 — Adição do plano de fundo no dashboard “Visão Geral da Instituição”, como exemplo, no Power BI. Para todos os outros dashboards (páginas), procedeu-se da mesma forma.

8.1. Processamento dos dados da base de dados no Power BI

Após a etapa inicial anterior, procedeu-se com a importação dos dados da base de dados em SQL Server:

Obter Dados

Tudo

Arquivo

Banco de Dados

Microsoft Fabric

Power Platform

Azure

Serviços Online

Outro

Tudo

Pasta de Trabalho do Excel

Texto/CSV

XML

JSON

Pasta

PDF

Parquet

Pasta do SharePoint

Banco de dados SQL Server

Banco de dados de Access

Oracle Database

Banco de dados IBM Db2

Banco de dados do IBM Informix (Beta)

IBM Netezza

Banco de dados MySQL

Importe dados de um banco de dados do SQL Server.

Conectores Certificados

Aplicativos de Modelo

Conectar

Cancelar

Figura 45 — Tela de seleção de obtenção de dados de um Banco de Dados SQL Server.

Escreveu-se o endereço do servidor que o BD está, que como aqui neste caso trata-se de uma máquina local, bastou colocar o hostname da máquina:

Banco de dados SQL Server

Servidor ⓘ

LAPTOP-KPASMNE2

Banco de Dados (opcional)

Modo de Conectividade de Dados ⓘ

☒ Importar

☐ DirectQuery

Opções avançadas

OK

Cancelar

Figura 46 — Escrita do servidor onde o BD SQL Server se encontra.

No quesito usuário de acesso ao BD, **utilizou-se o padrão de Autenticação do Windows**, mas, em um cenário real, criaria-se um usuário próprio

para isso no SQL Server, com a criptografia de segurança, e utilizaria-se ele para o acesso.

Em seguida, o Banco de Dados e suas tabelas foram carregadas, onde pode-se selecionar com exatidão de onde o Power BI irá processar os dados. Como no BD foram criadas Views, que já reúnem em tabelas virtuais tudo aquilo que será necessário para cada dashboard, selecionou-se elas — as views são as que iniciam com a sigla “vw”:

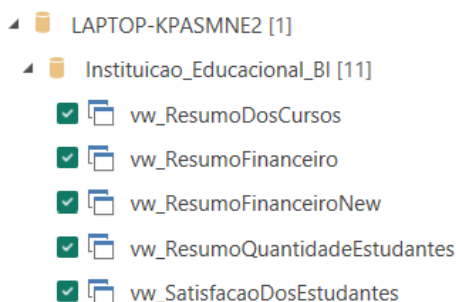


Figura 47 — Seleção para o processamento dos dados das views do BD “Instituicao_Educacional_BI”, criado no SQL Server.

Com as views devidamente selecionadas, primeiramente, procedeu-se para o Power Query para validar se não há quaisquer tipos de erros, linhas duplicadas ou linhas em branco.

Analisando as views, validou-se que não há nenhum dos problemas citados acima. Para tal análise, marcou-se, na seção “Exibição”, o campo “Qualidade da coluna”, de modo que um pequeno quadro passou a ser exibido antes das linhas dos dados de cada coluna de cada view. Abaixo, uma exemplificação com a view “vw_ResumoFinanceiro”.

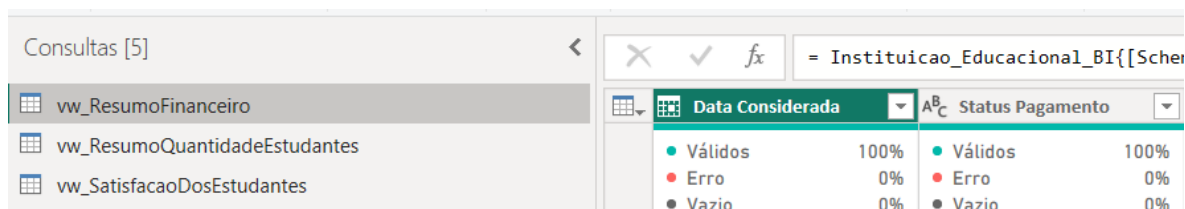


Figura 48 — Tratamento da qualidade dos dados no Power Query.

Com êxito na análise, no Power Query, clicou-se em “Fechar e Aplicar” e, finalmente, procedeu-se para a construção dos dashboards de fato.

8.2. Construção do menu de navegação

Com os planos de fundo aplicados em cada um dos 4 dashboards, procedeu-se da seguinte forma para a criação do menu de navegação entre eles:

- Incluiu-se um botão em branco e, em Botão > Estilo, fez-se a remoção do preenchimento e da borda. A ideia é deixá-lo invisível.

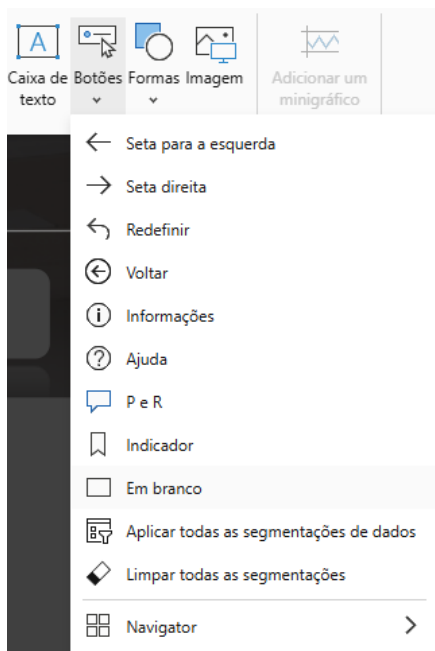


Figura 49 — Menu de inclusão de botão em branco, no Power BI.

- Em cada um dos retângulos com ícones (botões) criados no Power BI, adicionou-se esse botão em branco invisível, com exceção do botão que abre o dashboard que já está aberto: *isto é, por exemplo, no dashboard Visão Geral da Instituição, o botão que abre ele é o que consta o ícone de uma casa. Não colocou o botão em branco invisível sobre esse com o ícone de uma casa, pois o dashboard “Visão Geral da Instituição” é o que já está aberto. E assim fez-se com os outros dashboards também;*
- E em cada um dos botões, fez-se a configuração de navegação entre páginas e qual é a página que são abertas quando eles forem clicados, como no exemplo abaixo:

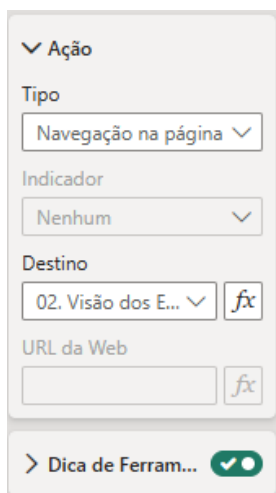


Figura 50 — Configuração de qual página o botão clicável irá carregar, no Power BI. Na imagem acima, fez-se a configuração do botão de abrir o dashboard “Visão dos Estudantes”.

- De modo a informar o usuário, em Botão > Ação > Dica de Ferramenta, adicionou-se uma mensagem do que cada botão abre. Ex.: “Visão dos Estudantes”;

Créditos:

TREINAMENTOS, Hashtag. Menu de Navegação Incrível no Power BI. 2022. 15 min. Disponível em: <https://www.youtube.com/watch?v=Lt9ZWb-Q3Kk>. Acesso em: 21 jul. 2025.

8.3. Construção do painel de filtros

Para a criação do painel de filtros, primeiramente, igual para os demais botões de navegação criados na etapa anterior — invisível, sem preenchimento e sem borda e com a dica de ferramenta “Abrir painel de filtros” — adicionou-se um também no botão com ícone de filtro, na parte inferior do menu de navegação. Em seguida, inseriu-se um retângulo com o canto superior direito arredondado de cor cinza (#333333), onde os elementos do painel de filtro serão inseridos:

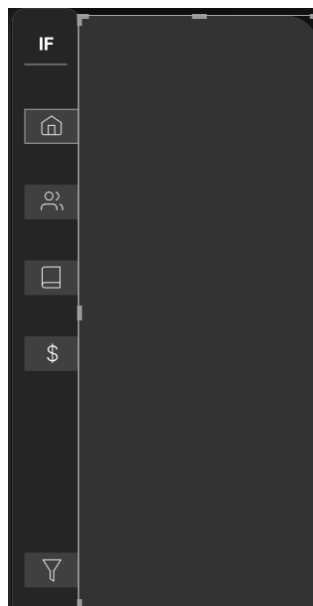


Figura 51 — Criação do retângulo do painel de filtros.

Posteriormente, dentro desse retângulo:

- Na parte superior, inseriu-se uma caixa de texto escrito “FILTROS”, onde colocou-se um espaço entre cada letra da palavra, para uma estética melhor. Também, inseriu-se um botão de X, o qual foi configurado para fechar o painel de filtros quando clicado, e uma linha acinzentada (#808080) de largura 1 px para uma divisão entre o cabeçalho do painel e os elementos de filtro;
- Para a criação de cada um dos 3 elementos de filtro — CIDADE, ESTADO e POLO —, adicionou-se o visual “Segmentação de Dados”, um abaixo do outro, logo em seguida da linha de divisão do cabeçalho:

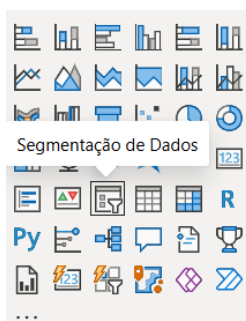


Figura 52 — Elemento de segmentação de dados, no Power BI.



Figura 53 — 3 elementos de segmentação de dados adicionados no painel de filtros, no Power BI.

- Em cada um dos três visuais de Segmentação de Dados adicionados, em “Campo”, arrastou-se a coluna “Cidade Considerada” da view “vw_SatisfacaoDosEstudantes” e clicou-se em “Renomear para este visual” para renomear para “CIDADE”, “ESTADO” e “POLO”, conforme a ordem de cada um, obtendo-se o resultado abaixo:



Figura 54 — 3 elementos de segmentação de dados configurados no painel de filtros, no Power BI.

- Na parte inferior do painel de filtros, para separar e manter um visual organizado, adicionou-se uma linha acinzentada (#666666) e logo abaixo dela um botão de limpar todos os filtros;

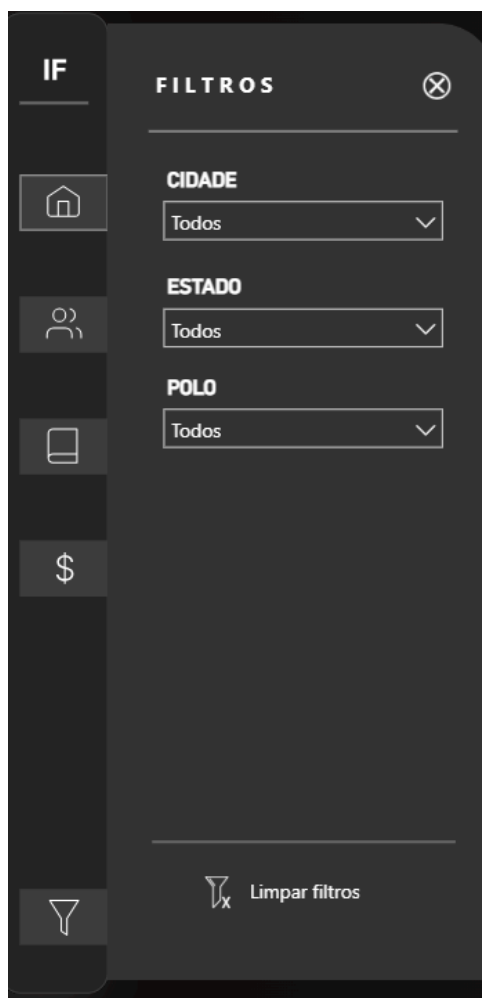


Figura 55 — Resultado final do painel de filtros.

- Para gerenciar a abertura e o fechamento do tal painel, na aba Seleção, agrupou-se todos esses elementos do painel de filtros em um grupo — o qual foi nomeado para “PAINEL DE FILTROS” — e procedeu-se para a criação de Indicadores da seguinte forma: com o PAINEL DE FILTROS visível, criou-se o indicador “Painel Exibido” e com o PAINEL DE FILTROS invisível, criou-se o indicador “Painel Oculto”;

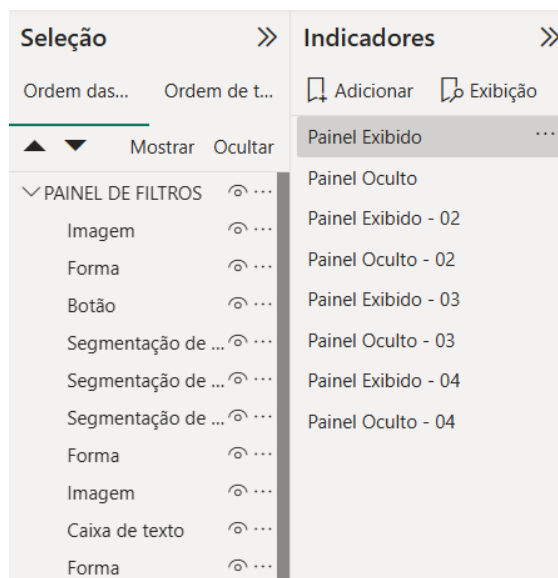


Figura 56 — Administração do Painel de Filtros Exibido nas abas “Seleção” e “Indicadores”.

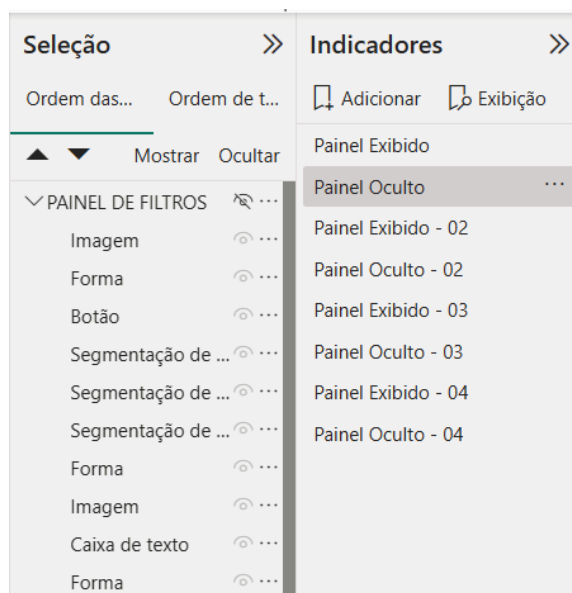


Figura 57 — Administração do Painel de Filtros Oculto nas abas “Seleção” e “Indicadores”.

- Com essas configurações feitas, no botão para abrir o painel de filtros, em “Ação”, adicionou-se o tipo = Indicador e Indicador = Painel Exibido:

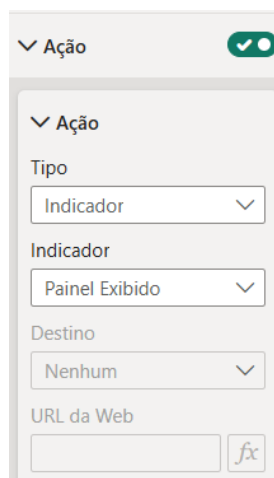


Figura 58 — Configuração da ação do botão invisível do menu de navegação correspondente à abertura do painel de filtros.

- E, com o painel aberto, no ícone de fechar (X), adicionou-se o tipo = Indicador e Indicador = Painel Oculto:

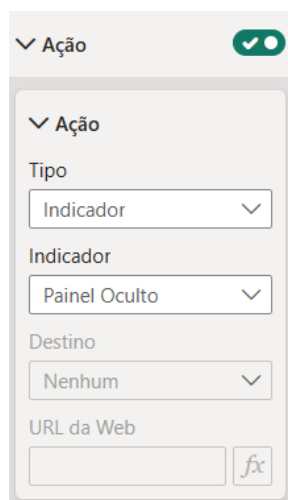


Figura 59 — Configuração da ação do botão X do painel de filtros correspondente ao fechamento do mesmo.

- Como são 4 páginas, de 4 dashboards, repetiu-se esse processo 4 vezes, nas 4 páginas;

Créditos:

SMIRELLI, Leticia. [Power BI] Como Construir um PAINEL DE FILTROS Retrátil e Botão Limpar Filtros. 2024. 15 min. Disponível em: <https://www.youtube.com/watch?v=3YMDgM1dLmY>. Acesso em: 21 jul. 2025.

8.4. Construção do dashboard Visão Geral da Instituição

8.4.1. Cards

Para a criação dos cards, arrastou-se 4 elementos “Cartão”, delimitando seus tamanhos exatamente ao tamanho do card desenhado no design e aplicando as seguintes estilizações:

- **Paleta de cores dos dados:** ATIVOS = Verde (#00FF4D), FORMADOS = Azul (#118DFF), DESISTENTES = Vermelho (#FF0013) e TRANCAMENTOS = Amarelo (#FFD300);
- **Em Visual > Rótulo da categoria:** foi desabilitado tal rótulo, para ficar somente o dado;
- **Em Visual > Valor do balão:** Fonte DIN em tamanho 16, Exibir unidades = Nenhum;
- **Em Geral > Efeitos:** Tela de fundo = Não;
- **Em Geral > Formato de dados:** Formato = Número inteiro, Separador de milhões = Sim;

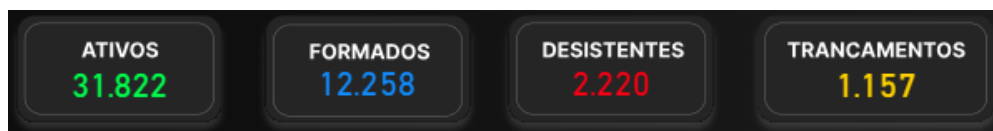


Figura 60 — Construção dos 4 cards no dashboard “Visão Geral da Instituição”, no Power BI.

8.4.2. Gráfico de colunas empilhadas — Média de Preço dos Cursos

Para a criação do gráfico de colunas empilhadas que exibe a média de preço dos cursos, arrastou-se o elemento “Gráfico de colunas empilhadas”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Eixo X:** Nome do Curso, processado da view “vw_ResumoDosCursos”;
- **Estilização:** Fonte = 9, Cor = Branco, Sem Título;
- **Eixo Y:** Média de Preço, processado da view “vw_ResumoDosCursos”;
- **Estilização:** Fonte = 9, Cor = Branco, Sem Título;
- **Cor das colunas:** #CCCCCC;
- **Efeitos:** Sem tela de fundo;

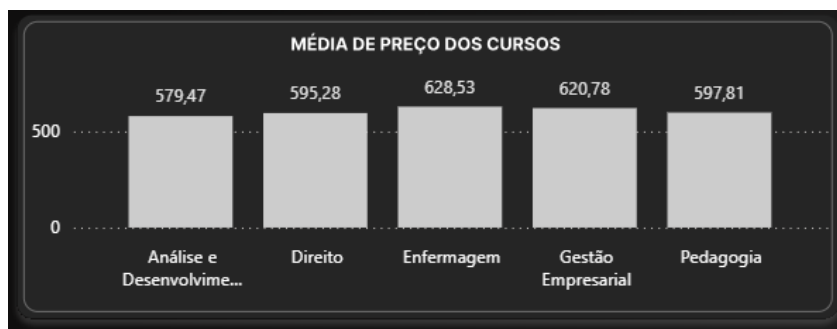


Figura 61 — Construção do gráfico de colunas empilhadas “Média de Preço dos Cursos”, no dashboard “Visão Geral da Instituição”, no Power BI.

8.4.3. Tabela — Cursos vs. Vagas Disponíveis

Para a criação da tabela que exibe os cursos disponíveis e as respectivas quantidades de vagas, arrastou-se o elemento “Tabela”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Colunas:** Cursos Disponíveis e Qtd. de Vagas Disponíveis (renomeado para o visual), campos processados da view “vw_ResumoDosCursos”;
- **Linhas de grade:** Cor #E5E5E5;
- **Valores:**
 - **Texto padrão e texto alternativo:** fonte Segoe UI tamanho 11, cor branca;
 - **Cor da tela de fundo padrão:** #252525;
 - **Cor da tela de fundo alternativa:** #1A1A1A;
 - **Totais:** fonte Segoe UI tamanho 10, cor branca;
- **Cabeçalhos:**
 - **Texto:** fonte Segoe UI tamanho 10, cor branca;
 - **Cor da tela de fundo:** #252525;
 - **Alinhamento:** ao centro;
- **Efeitos:** Sem tela de fundo;

CURSOS vs. VAGAS DISPONÍVEIS	
Cursos Disponíveis	Qtd. de Vagas Disponíveis
Análise e Desenvolvimento de Sistemas	11306
Direito	11564
Enfermagem	11687
Gestão Empresarial	10626
Pedagogia	11502
Total	56685

Figura 62 — Construção da tabela “Cursos vs. Vagas Disponíveis”, no dashboard “Visão Geral da Instituição”, no Power BI.

8.4.4. Gráfico de Pizza — Satisfação dos Estudantes vs. Cursos

Para a criação do gráfico de pizza que exibe o percentual de satisfação dos estudantes com relação aos seus cursos, arrastou-se o elemento “Tabela”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Legenda:**

- Criou-se uma nova medida por nome Classificação Satisfação e inseriu-se como Legenda;
- A medida é:

```

Classificação Satisfação =
SWITCH(
    TRUE(),
    'vw_SatisfacaoDosEstudantes'[Satisfação] <= 1.9, "Não Satisfeito",
    'vw_SatisfacaoDosEstudantes'[Satisfação] <= 3.5, "Pouco Satisfeito",
    'vw_SatisfacaoDosEstudantes'[Satisfação] <= 3.9, "Razoavelmente Satisfeito",
    'vw_SatisfacaoDosEstudantes'[Satisfação] <= 5, "Muito Satisfeito"
)

```

- Ela segue esse padrão abaixo:

Satisfação <= 1.9 → **Não Satisfeito** - cor vermelha #FF0013

Satisfação <= 3.5 → **Pouco Satisfeito** - cor Azul #118DFF

Satisfação <= 3.9 → **Razoavelmente Satisfeito** - cor Amarela #FFD300

Satisfação <= 5 → **Muito Satisfeito** - cor Verde #00FF4D

- **Centralização:** à direita;
- **Valores:** Soma de Satisfação, campo processado da view “vw_SatisfacaoDosEstudantes”;
- **Fatias:**
 - **Não Satisfeito** - cor vermelha #FF0013;
 - **Pouco Satisfeito** - cor Azul #118DFF;
 - **Razoavelmente Satisfeito** - cor Amarela #FFD300;
 - **Muito Satisfeito** - cor Verde #00FF4D;
- **Efeitos:** Sem tela de fundo;

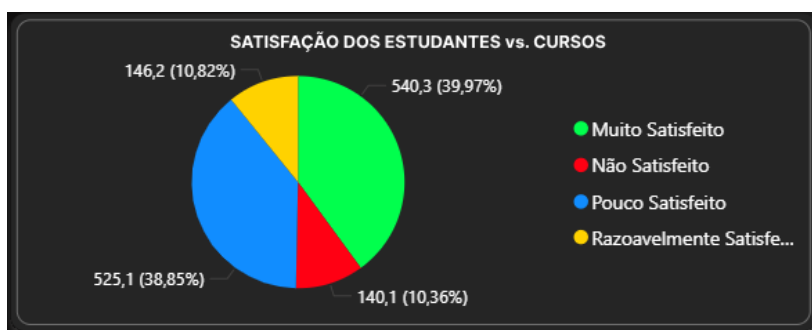


Figura 63 — Construção do gráfico de pizza “Satisfação dos Estudantes vs. Cursos”, no dashboard “Visão Geral da Instituição”, no Power BI.

8.4.5. Gráfico de colunas clusterizado — Adimplência e Inadimplência por Curso

Para a criação do gráfico de barras clusterizado que exibe o número de estudantes com pagamentos adimplentes e inadimplentes com relação aos cursos, arrastou-se o elemento “Gráfico de colunas clusterizado”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Eixo X:** Nome do Curso, processado da view “vw_ResumoFinanceiro”;
- **Eixo Y:** Adimplência e Inadimplência, renomeados para o visual os campos Adimplente e Inadimplente processados da view “vw_ResumoFinanceiro”;
- **Valores:**
 - **Texto:** fonte Segoe UI tamanho 9, cor branca;
- **Legenda:** posição = Superior central;
- **Colunas:**
 - **Adimplência** = cor #00FF4D;

- Inadimplência = cor #FF0013;

- **Efeitos:** Sem tela de fundo;

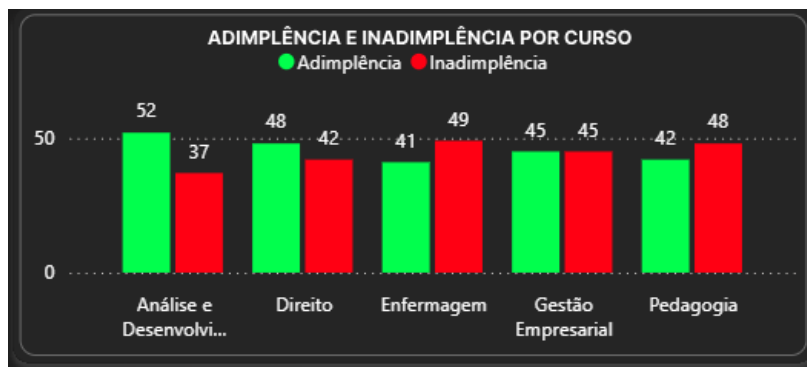


Figura 64 — Construção do gráfico de colunas clusterizado “Adimplência e Inadimplência por Curso”, no dashboard “Visão Geral da Instituição”, no Power BI.

8.5. Construção do dashboard Visão dos Estudantes

8.5.1. Cards

Para a criação dos cards, arrastou-se 4 elementos “Cartão”, delimitando seus tamanhos exatamente ao tamanho do card desenhado no design e aplicando as seguintes estilizações:

- **Paleta de cores dos dados:** ATIVOS = Verde (#00FF4D), FORMADOS = Azul (#118DFF), DESISTENTES = Vermelho (#FF0013) e TRANCAMENTOS = Amarelo (#FFD300);
- **Em Visual > Rótulo da categoria:** foi desabilitado tal rótulo, para ficar somente o dado;
- **Em Visual > Valor do balão:** Fonte DIN em tamanho 16, Exibir unidades = Nenhum;
- **Em Geral > Efeitos:** Tela de fundo = Não;
- **Em Geral > Formato de dados:** Formato = Número inteiro, Separador de milhões = Sim;

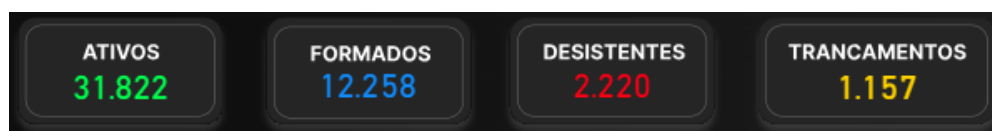


Figura 65 — Construção dos 4 cards no dashboard “Visão dos Estudantes”, no Power BI.

8.5.2. Gráfico de colunas empilhadas — Média de Satisfação com o Curso por UF

Para a criação do gráfico de colunas empilhadas que exibe a média de satisfação com relação aos cursos por UF, arrastou-se o elemento “Gráfico de colunas empilhadas”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Eixo X:** UF Respectiva, processado da view “vw_SatisfacaoDosEstudantes”;
- **Eixo Y:** Satisfação, processado da view “vw_SatisfacaoDosEstudantes”;
- **Linhas de grade:** Cor #C8C6C4;
- **Valores:**
 - **Texto:** fonte Segoe UI tamanho 9, cor branca;
- **Efeitos:** Sem tela de fundo;



Figura 66 — Construção do gráfico de colunas empilhadas “Média de Satisfação com o Curso por UF”, no dashboard “Visão dos Estudantes”, no Power BI.

8.5.3. Gráfico de linhas — Média de Satisfação com o Curso por Mês

Para a criação do gráfico de linhas que exibe a média de satisfação geral da instituição com o passar dos meses, arrastou-se o elemento “Gráfico de linhas”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Eixo X:** Data Considerada (Ano, Mês), processado da view “vw_SatisfacaoDosEstudantes”;
- **Eixo Y:** Satisfação, processado da view “vw_SatisfacaoDosEstudantes”;
- **Valores:**
 - **Texto:** fonte Segoe UI tamanho 9, cor branca;
- **Linhas:** cor #118DFF;

- **Efeitos:** Sem tela de fundo;

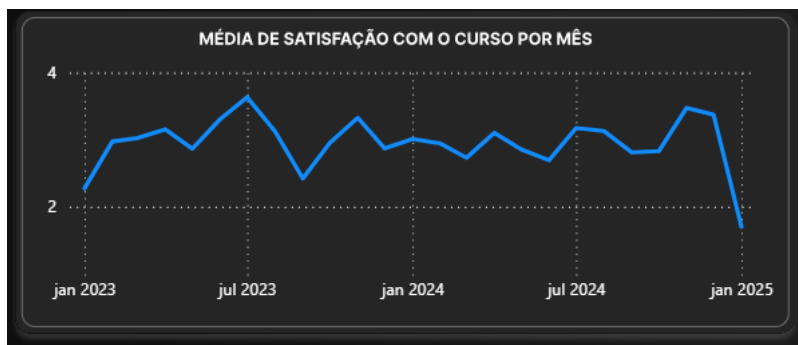


Figura 67 — Construção do gráfico de linhas “Média de Satisfação com o Curso por Mês”, no dashboard “Visão dos Estudantes”, no Power BI.

8.5.4. Card — Média Geral de Satisfação da Instituição

Para a criação do card que exibe a média de satisfação geral da instituição, arrastou-se o elemento “Cartão”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Campos:** Satisfação, processado da view “vw_SatisfacaoDosEstudantes”;
- A cor do valor exibido altera-se conforme as regras de satisfação, já explicitadas anteriormente. O valor possui fonte DIN tamanho 16;
- Criou-se as regras:

Cor - Valor do balão



Estilo do formato

Regras

Em que campo devemos basear isso?

Média de Satisfação

Resumo

Média

Regras

↕ Inverter a ordem de cores

+ Nova regra

Se o valor	>=	0	Número	e	<=	1,9	Número	então	vermelho	↑ ↓ ×
Se o valor	>=	2	Número	e	<=	3,5	Número	então	azul	↑ ↓ ×
Se o valor	>=	3,6	Número	e	<=	3,9	Número	então	amarelo	↑ ↓ ×
Se o valor	>=	4	Número	e	<=	5	Número	então	verde	↑ ↓ ×

Figura 68 — Configuração da regra para mudar a cor do valor exibido no card, conforme as regras de satisfação estabelecidas, no Power BI.

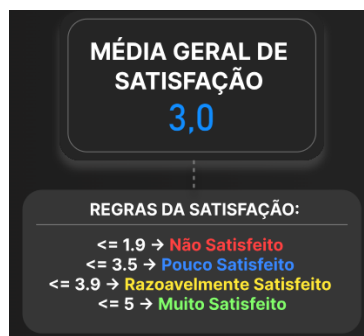


Figura 69 — Construção do card “Média Geral de Satisfação”, no dashboard “Visão dos Estudantes”, no Power BI.

8.5.5. Tabela — Cursos Disponíveis

Para a criação da tabela que exibe os cursos disponíveis, arrastou-se o elemento “Tabela”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Colunas:** Cursos Disponíveis (renomeado para o visual), campo “Nome do Curso” processado da view “vw_ResumoDosCursos”;
- **Linhas de grade:** Cor #E5E5E5;
- **Valores:**
 - **Texto padrão e texto alternativo:** fonte Segoe UI tamanho 11, cor branca;
 - **Cor da tela de fundo padrão:** #252525;
 - **Cor da tela de fundo alternativa:** #1A1A1A;
 - **Totais:** fonte Segoe UI tamanho 10, cor branca;
- **Cabeçalhos:**
 - **Texto:** fonte Segoe UI tamanho 10, cor branca;
 - **Cor da tela de fundo:** #252525;
 - **Alinhamento:** ao centro;
- **Efeitos:** Sem tela de fundo;

Cursos Disponíveis
Análise e Desenvolvimento de Sistemas
Direito
Enfermagem
Gestão Empresarial
Pedagogia

Figura 70 — Construção da tabela “Cursos Disponíveis”, no dashboard “Visão dos Estudantes”, no Power BI.

8.5.6. Gráfico de Pizza — Satisfação dos Estudantes vs. Cursos

Para a criação do gráfico de pizza que exibe o percentual de satisfação dos estudantes com relação aos seus cursos, arrastou-se o elemento “Tabela”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Legenda:**

- Criou-se uma nova medida por nome Classificação Satisfação e inseriu-se como Legenda;
- A medida é:

```
Classificação Satisfação =  
SWITCH(  
  TRUE(),  
  'vw_SatisfacaoDosEstudantes'[Satisfação] <= 1.9, "Não Satisfeito",  
  'vw_SatisfacaoDosEstudantes'[Satisfação] <= 3.5, "Pouco Satisfeito",  
  'vw_SatisfacaoDosEstudantes'[Satisfação] <= 3.9, "Razoavelmente Satisfeito",  
  'vw_SatisfacaoDosEstudantes'[Satisfação] <= 5, "Muito Satisfeito"  
)
```

- Ela segue esse padrão abaixo:

Satisfação <= 1.9 → **Não Satisfeito** - cor vermelha #FF0013

Satisfação <= 3.5 → **Pouco Satisfeito** - cor Azul #118DFF

Satisfação <= 3.9 → **Razoavelmente Satisfeito** - cor Amarela #FFD300

Satisfação <= 5 → **Muito Satisfeito** - cor Verde #00FF4D

- **Centralização:** à direita;

- **Valores:** Soma de Satisfação, campo processado da view “vw_SatisfacaoDosEstudantes”;
- **Fatias:**
 - **Não Satisfeito** - cor vermelha #FF0013;
 - **Pouco Satisfeito** - cor Azul #118DFF;
 - **Razoavelmente Satisfeito** - cor Amarela #FFD300;
 - **Muito Satisfeito** - cor Verde #00FF4D;
- **Efeitos:** Sem tela de fundo;

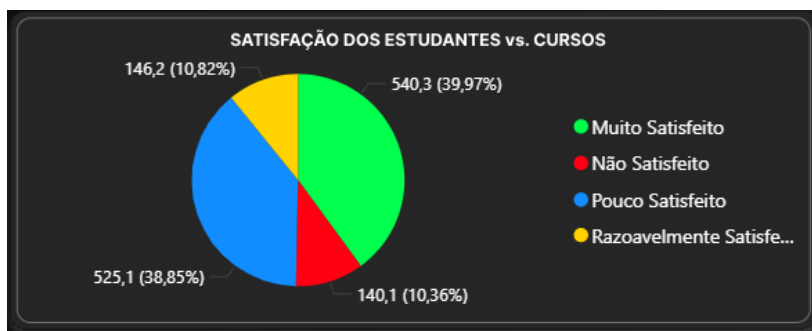


Figura 71 — Construção do gráfico de pizza “Satisfação dos Estudantes vs. Cursos”, no dashboard “Visão dos Estudantes”, no Power BI.

8.5.7. Funil — Percentual de Formados, Desistentes e Trancamentos com Relação à Ativação da Matrícula

Para a criação do funil que exibe o percentual de formados, desistente e trancamentos por parte dos estudantes com relação à ativação da matrícula, arrastou-se o elemento “Funil”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Valores:**

- **Formados, Desistentes e Trancamentos:** criou-se as medidas abaixo.

```

Proporcao_Formados_New = DIVIDE(SUM(vw_ResumoQuantidadeEstudantes[Total_Formados]),
SUM(vw_ResumoQuantidadeEstudantes[Total_Ativos]), 0);

Proporcao_Desistentes_New = DIVIDE(SUM(vw_ResumoQuantidadeEstudantes[Total_Desistentes]),
SUM(vw_ResumoQuantidadeEstudantes[Total_Ativos]), 0);

Proporcao_Trancamentos_New = DIVIDE(SUM(vw_ResumoQuantidadeEstudantes[Total_Trancamentos]),
SUM(vw_ResumoQuantidadeEstudantes[Total_Ativos]), 0);

```

- **Texto:** fonte Segoe UI tamanho 10, cor branca;

- **Formato de dados:** Porcentagem, 2 casas decimais;

- **Cores:** para Formados = #118DFF, para Desistentes = #FF0013 e para Trancamentos = #FFD300;

- **Efeitos:** Sem tela de fundo;

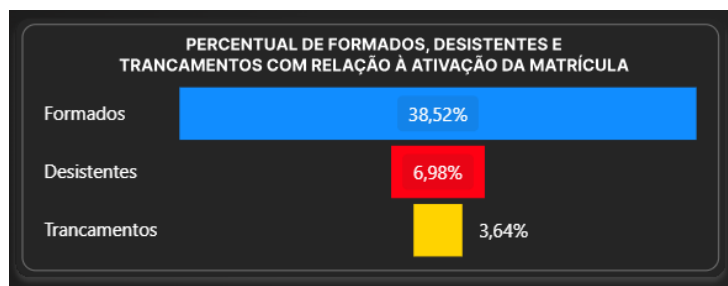


Figura 72 — Construção do funil “Percentual de Formados, Desistentes e Trancamentos com Relação à Matrícula”, no dashboard “Visão dos Estudantes”, no Power BI.

8.6. Construção do dashboard Visão dos Cursos

8.6.1. Cards

Para a criação dos cards, arrastou-se 6 elementos “Cartão”, delimitando seus tamanhos exatamente ao tamanho do card desenhado no design e aplicando as seguintes estilizações:

- **Campos:** Nome do Curso e Vagas Disponíveis, processados da view “vw_ResumoDosCursos”;
- **Fórmulas criadas:** Nome do Curso Mais Caro, Nome do Curso Mais Barato, Nome do Curso com Mais Vagas, Nome do Curso com Menos Vagas;

```

NomeCursoMaisCaro =
VAR PrecoMax =
    CALCULATE(
        MAX('vw_ResumoDosCursos'[Preço]),
        ALLSELECTED('vw_ResumoDosCursos')
    )
RETURN
    CALCULATE(
        FIRSTNONBLANK(
            SELECTCOLUMNS(
                FILTER(
                    ALLSELECTED('vw_ResumoDosCursos'),
                    'vw_ResumoDosCursos'[Preço] = PrecoMax
                ),
                "Curso", 'vw_ResumoDosCursos'[Nome do Curso]
            ),
            1
        )
    )
  
```

```
)  
);  
  
NomeCursoMaisBarato =  
VAR PrecoMin =  
    CALCULATE(  
        MIN('vw_ResumoDosCursos'[Preço]),  
        ALLSELECTED('vw_ResumoDosCursos')  
    )  
RETURN  
CALCULATE(  
    FIRSTNONBLANK(  
        SELECTCOLUMNS(  
            FILTER(  
                ALLSELECTED('vw_ResumoDosCursos'),  
                'vw_ResumoDosCursos'[Preço] = PrecoMin  
            ),  
            "Curso", 'vw_ResumoDosCursos'[Nome do Curso]  
        ),  
        1  
    )  
);  
  
NomeCursoComMaisVagas =  
VAR VagasMax =  
    CALCULATE(  
        MAX('vw_ResumoDosCursos'[Vagas Disponíveis]),  
        ALLSELECTED('vw_ResumoDosCursos')  
    )  
RETURN  
CALCULATE(  
    FIRSTNONBLANK(  
        SELECTCOLUMNS(  
            FILTER(  
                ALLSELECTED('vw_ResumoDosCursos'),  
                'vw_ResumoDosCursos'[Vagas Disponíveis] = VagasMax  
            ),  
            "Curso", 'vw_ResumoDosCursos'[Nome do Curso]  
        ),  
        1  
    )  
);
```

```

)
);

NomeCursoComMenosVagas =
VAR VagasMin =
    CALCULATE(
        MIN('vw_ResumoDosCursos'[Vagas Disponíveis]),
        ALLSELECTED('vw_ResumoDosCursos')
    )
RETURN
CALCULATE(
    FIRSTNONBLANK(
        SELECTCOLUMNS(
            FILTER(
                ALLSELECTED('vw_ResumoDosCursos'),
                'vw_ResumoDosCursos'[Vagas Disponíveis] = VagasMin
            ),
            "Curso", 'vw_ResumoDosCursos'[Nome do Curso]
        ),
        1
    )
);

```

- **Em Visual > Rótulo da categoria:** foi desabilitado tal rótulo, para ficar somente o dado;
- **Em Visual > Valor do balão:** Fonte DIN em tamanho 12, Exibir unidades = Nenhum;
- **Em Geral > Efeitos:** Tela de fundo = Não;
- **Em Geral > Formato de dados:** Formato = Número inteiro, Separador de milhões = Sim;

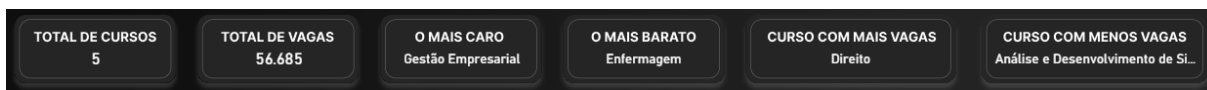


Figura 73 — Construção dos 6 cards, no dashboard “Visão dos Cursos”, no Power BI.

8.6.2. Gráfico de barras clusterizado — Média de Vagas Disponíveis por Curso

Para a criação do gráfico de barras clusterizado que exibe a média de vagas disponíveis em cada curso considerando todos os polos, arrastou-se o elemento “Gráfico de barras clusterizado”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Eixo X:** Vagas Disponíveis, processado da view “vw_ResumoDosCursos”;
 - **Valores:** fonte Segoe UI tamanho 9, cor branca;
 - **Título:** Número de Vagas, fonte Segoe UI tamanho 9, cor #B3B3B3;
- **Eixo Y:** Nome do Curso, processado da view “vw_ResumoDosCursos”;
 - **Valores:** fonte Segoe UI tamanho 9, cor branca;
 - **Título:** Cursos, fonte Segoe UI tamanho 9, cor #B3B3B3;
- **Rótulo de dados:** habilitado, fonte Segoe UI tamanho 9, cor #B3B3B3;
- **Efeitos:** Sem tela de fundo;

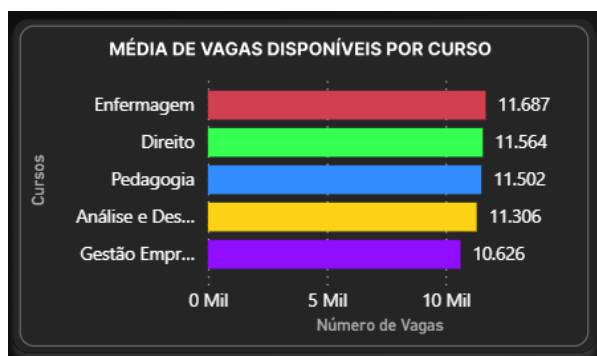


Figura 74 — Construção do gráfico de barras clusterizado “Média de Vagas Disponíveis por Curso”, no dashboard “Visão dos Cursos”, no Power BI.

8.6.3. Gráfico de colunas clusterizado — Média de Preço dos Cursos por UF

Para a criação do gráfico de colunas clusterizado que exibe a média de preço dos cursos com relação a UF, arrastou-se o elemento “Gráfico de colunas clusterizado”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Eixo X:** UF, processado da view “vw_ResumoDosCursos”;
 - **Valores:** fonte Segoe UI tamanho 9, cor branca;
 - **Título:** UF, fonte Segoe UI tamanho 9, cor #B3B3B3;
- **Eixo Y:** Média de Preço, processado da view “vw_ResumoDosCursos”;
 - **Valores:** fonte Segoe UI tamanho 9, cor branca;

- **Título:** Média de Preços (em R\$), fonte Segoe UI tamanho 9, cor #B3B3B3;
- **Rótulos de dados:** habilitado, fonte Segoe UI tamanho 9, cor #B3B3B3;
- **Colunas:** cor #B3B3B3;
- **Efeitos:** Sem tela de fundo;

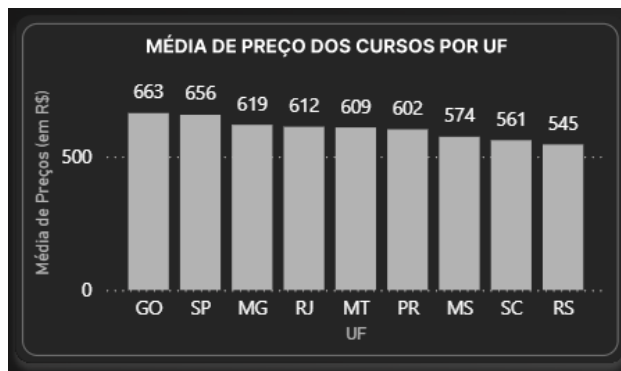


Figura 75 — Construção de colunas clusterizado “Média de Preço dos Cursos por UF”, no dashboard “Visão dos Cursos”, no Power BI.

8.6.4. Tabela — Cursos Disponíveis

Para a criação da tabela que exibe os cursos disponíveis, arrastou-se o elemento “Tabela”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Colunas:** Cursos Disponíveis (renomeado para o visual), campo “Nome do Curso” processado da view “vw_ResumoDosCursos”;
- **Linhas de grade:** Cor #E5E5E5;
- **Valores:**
 - **Texto padrão e texto alternativo:** fonte Segoe UI tamanho 11, cor branca;
 - **Cor da tela de fundo padrão:** #252525;
 - **Cor da tela de fundo alternativa:** #1A1A1A;
 - **Totais:** fonte Segoe UI tamanho 10, cor branca;
- **Cabeçalhos:**
 - **Texto:** fonte Segoe UI tamanho 10, cor branca;
 - **Cor da tela de fundo:** #252525;
 - **Alinhamento:** ao centro;
- **Efeitos:** Sem tela de fundo;

Cursos Disponíveis
Análise e Desenvolvimento de Sistemas
Direito
Enfermagem
Gestão Empresarial
Pedagogia

Figura 76 — Construção da tabela “Cursos Disponíveis”, no dashboard “Visão dos Cursos”, no Power BI.

8.6.5. Gráfico de linhas — Média de Preços Totais dos Cursos por Ano

Para a criação do gráfico de linhas que exibe a média de preços totais dos cursos por ano, arrastou-se o elemento “Gráfico de linhas”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Eixo X:** Data Considerada (Ano, Mês), processado da view “vw_ResumoDosCursos”;
 - **Valores:** fonte Segoe UI tamanho 9, cor branca;
 - **Título:** Cursos, fonte Segoe UI tamanho 9, cor #B3B3B3;
- **Eixo Y:** Média de Preço, processado da view “vw_ResumoDosCursos”;
 - **Valores:** fonte Segoe UI tamanho 9, cor branca;
 - **Título:** Média de Preços (em R\$), fonte Segoe UI tamanho 9, cor #B3B3B3;
- **Legenda:** Nome do Curso, processado da view “vw_ResumoDosCursos”;
 - **Linhas:** Análise e Desenvolvimento de Sistemas (cor #FFD300), Direito (cor #00FF4D), Enfermagem (cor #D64550), Gestão Empresarial (cor #9010FF) e Pedagogia (cor #118DFF);
- **Efeitos:** Sem tela de fundo;

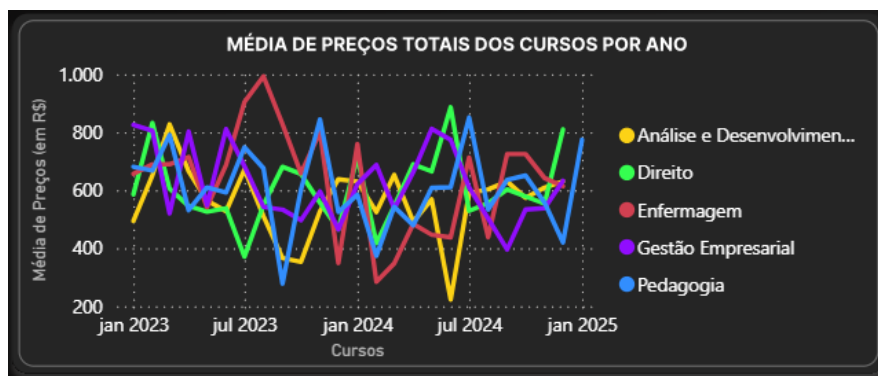


Figura 76 — Construção do gráfico de linhas “Média de Preços Totais dos Cursos por Ano”, no dashboard “Visão dos Cursos”, no Power BI.

8.6.6. Tabela — Quantidade de Vagas Disponíveis por Curso e por UF

Para a criação da tabela que exibe a quantidade de vagas disponíveis por curso e por UF, arrastou-se o elemento “Tabela”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Linhas:** Cursos Disponíveis (renomeado para o visual), campo “Nome do Curso” processado da view “vw_ResumoDosCursos”;
- **Colunas:** UF Respectiva, processado da view “vw_ResumoDosCursos”;
- **Valores:** Vagas Disponíveis, processado da view “vw_ResumoDosCursos”;
- **Linhas de grade:** Cor #E5E5E5;
- **Valores:**
 - **Texto padrão e texto alternativo:** fonte Segoe UI tamanho 11, cor branca;
 - **Cor da tela de fundo padrão:** #252525;
 - **Cor da tela de fundo alternativa:** #1A1A1A;
 - **Totais:** fonte Segoe UI tamanho 10, cor branca;
- **Cabeçalhos:**
 - **Texto:** fonte Segoe UI tamanho 10, cor branca;
 - **Cor da tela de fundo:** #252525;
 - **Alinhamento:** ao centro;
- **Efeitos:** Sem tela de fundo;

QUANTIDADE DE VAGAS DISPONÍVEIS POR CURSO E POR UF						
Nome do Curso	GO	MG	MS	MT	PR	R
Análise e Desenvolvimento de Sistemas	1034	1220	1268	1147	1368	12
Direito	1212	1429	1164	1296	1198	12
Enfermagem	1283	1358	1303	1250	1475	14
Gestão Empresarial	1221	1317	1144	1059	1312	12
Pedagogia	1539	1192	1483	1311	1036	12
Total	6289	6516	6362	6063	6389	65

Figura 77 — Construção da tabela “Quantidade de Vagas Disponíveis por Curso e por UF”, no dashboard “Visão dos Cursos”, no Power BI.

8.7. Construção do dashboard Visão Financeira

8.7.1. Cards

Para a criação dos cards, arrastou-se 6 elementos “Cartão”, delimitando seus tamanhos exatamente ao tamanho do card desenhado no design e aplicando as seguintes estilizações:

- **Campos:** Adimplente e Inadimplente, processados da view “vw_ResumoFinanceiro”;
- **Fórmulas criadas:** Curso com Mais Adimplência e Curso Mais Inadimplência

```
CursoMaisAdimplente =  
VAR CursosAdimplentes =  
    SUMMARIZE(  
        FILTER(  
            'vw_ResumoFinanceiroNew',  
            'vw_ResumoFinanceiroNew'[Adimplente] = 1  
        ),  
        'vw_ResumoFinanceiroNew'[Nome do Curso],  
        "QuantidadeAdimplentes", COUNTROWS('vw_ResumoFinanceiroNew')  
    )  
VAR CursoMaxAdimplentes =  
    CALCULATE(  
        MAXX(CursosAdimplentes, [QuantidadeAdimplentes])  
    )  
RETURN  
    CALCULATE(  
        FIRSTNONBLANK('vw_ResumoFinanceiroNew'[Nome do Curso], 1),  
        FILTER(CursosAdimplentes, [QuantidadeAdimplentes] = CursoMaxAdimplentes)  
    );  
  
CursoMaisInadimplente =  
VAR CursosInadimplentes =  
    SUMMARIZE(  
        FILTER(  
            'vw_ResumoFinanceiroNew',  
            'vw_ResumoFinanceiroNew'[Inadimplente] = 1  
        ),  
        'vw_ResumoFinanceiroNew'[Nome do Curso],  
        "QuantidadeInadimplentes", COUNTROWS('vw_ResumoFinanceiroNew')  
    )
```



```
'vw_ResumoFinanceiroNew'[Nome do Curso],
"QuantidadeInadimplentes", COUNTROWS('vw_ResumoFinanceiroNew')
)
VAR CursoMaxInadimplentes =
    CALCULATE(
        MAXX(CursosInadimplentes, [QuantidadeInadimplentes])
    )
RETURN
    CALCULATE(
        FIRSTNONBLANK('vw_ResumoFinanceiroNew'[Nome do Curso], 1),
        FILTER(CursosInadimplentes, [QuantidadeInadimplentes] = CursoMaxInadimplentes)
    );
```

- **Em Visual > Rótulo da categoria:** foi desabilitado tal rótulo, para ficar somente o dado;
- **Em Visual > Valor do balão:** Fonte DIN em tamanho 12, Exibir unidades = Nenhum;
- **Em Geral > Efeitos:** Tela de fundo = Não;
- **Em Geral > Formato de dados:** Formato = Número inteiro, Separador de milhões = Sim;



Figura 78 — Construção dos 4 cards, no dashboard “Visão Financeira”, no Power BI.

8.7.2. Gráfico de colunas clusterizado — Adimplência e Inadimplência por Curso

Para a criação do gráfico de barras clusterizado que exibe o número de estudantes com pagamentos adimplentes e inadimplentes com relação aos cursos, arrastou-se o elemento “Gráfico de colunas clusterizado”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Eixo X:** Nome do Curso, processado da view “vw_ResumoFinanceiro”;
- **Eixo Y:** Adimplência e Inadimplência, renomeados para o visual os campos Adimplente e Inadimplente processados da view “vw_ResumoFinanceiro”;

- **Valores:**
 - **Texto:** fonte Segoe UI tamanho 9, cor branca;
- **Legenda:** posição = Superior central;
- **Colunas:**
 - **Adimplência** = cor #00FF4D;
 - **Inadimplência** = cor #FF0013;
- **Efeitos:** Sem tela de fundo;

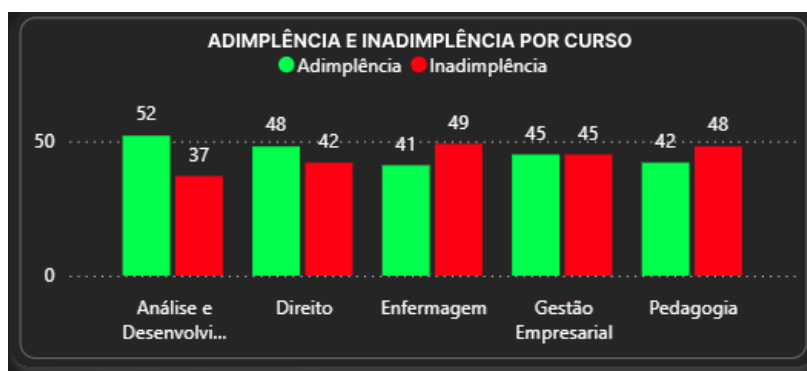


Figura 79 — Construção do gráfico de colunas clusterizado “Adimplência e Inadimplência por Curso”, no dashboard “Visão Financeira”, no Power BI.

8.7.3. Gráfico de linhas — Variação de Adimplência e Inadimplência por Curso

Para a criação do gráfico de linhas que exibe o número de adimplentes e inadimplentes com relação aos cursos, arrastou-se o elemento “Gráfico de linhas”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Eixo X:** Nome do Curso, processado da view “vw_ResumoFinanceiro”;
 - **Texto:** fonte Segoe UI tamanho 9, cor branca;
- **Eixo Y:** Adimplência, processado da view “vw_ResumoFinanceiro”;
 - **Texto:** fonte Segoe UI tamanho 9, cor branca;
- **Eixo Y secundário:** Inadimplência processado da view “vw_ResumoFinanceiro”;
- **Legenda:**
 - Posição: Superior central;
- **Linhas:**
 - Adimplência, cor #00FF4D;
 - Inadimplência, cor #D64550;

- **Efeitos:** Sem tela de fundo;

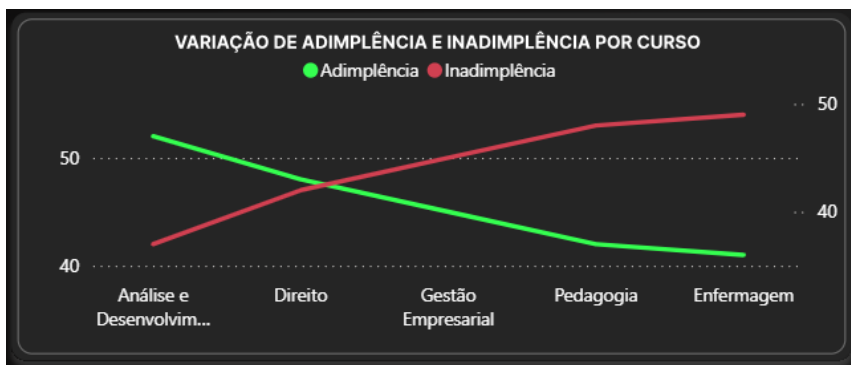


Figura 80 — Construção do gráfico de linhas “Variação de Adimplência e Inadimplência por Curso”, no dashboard “Visão Financeira”, no Power BI.

8.7.4. Gráfico de área — Variação da Adimplência por Mês

Para a criação do gráfico de área que exibe a variação da adimplência por mês, arrastou-se o elemento “Gráfico de linhas”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Eixo X:** Data Considerada (Ano, Mês), processado da view “vw_ResumoFinanceiro”;
- **Texto:** fonte Segoe UI tamanho 9, cor branca;
- **Eixo Y:** Adimplência, processado da view “vw_ResumoFinanceiro”;
- **Texto:** fonte Segoe UI tamanho 9, cor branca;
- **Linhas:** cor #00FF4D;
- **Efeitos:** Sem tela de fundo;

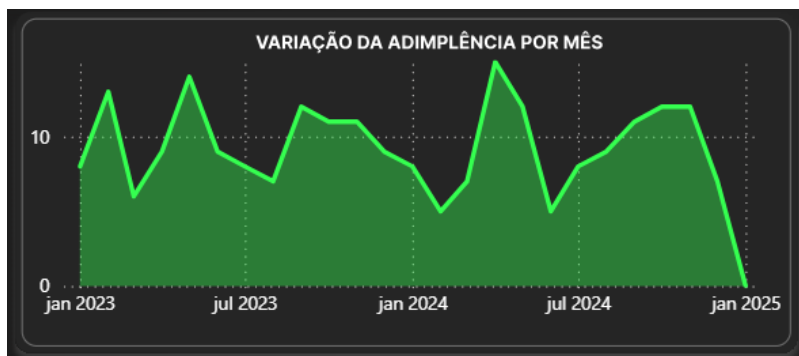


Figura 81 — Construção do gráfico de área “Variação da Adimplência por Mês”, no dashboard “Visão Financeira”, no Power BI.

8.7.4. Gráfico de área — Variação da Inadimplência por Mês

Para a criação do gráfico de área que exibe a variação da inadimplência por mês, arrastou-se o elemento “Gráfico de linhas”, delimitando seu tamanho exatamente ao tamanho desenhado no design e aplicando as seguintes estilizações e determinações:

- **Eixo X:** Data Considerada (Ano, Mês), processado da view “vw_ResumoFinanceiro”;
 - **Texto:** fonte Segoe UI tamanho 9, cor branca;
- **Eixo Y:** Inadimplência, processado da view “vw_ResumoFinanceiro”;
 - **Texto:** fonte Segoe UI tamanho 9, cor branca;
- **Linhas:** cor #D64550;
- **Efeitos:** Sem tela de fundo;

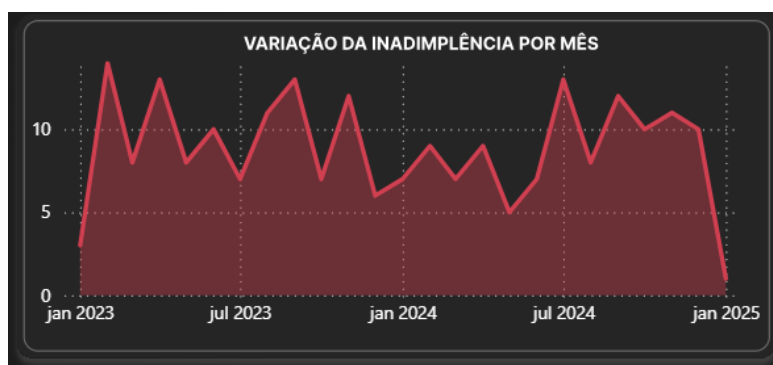


Figura 82 — Construção do gráfico de área “Variação da Inadimplência por Mês”, no dashboard “Visão Financeira”, no Power BI.