

DAX & POWER BI

Guia prático com funções, visualizações e indicações de ferramentas gratuitas para dashboards para estudantes em desenvolvimento



Mini Dicionário DAX & Visualização no Power BI

Um Guia Rápido para o Desenvolvedor Iniciante

🌟 Apresentação

Olá! Meu nome é Karen, sou estudante de Ciência de Dados e estou em constante desenvolvimento para me tornar uma analista de dados com excelência comercial.

Este Mini Dicionário DAX + Guia de Visualização foi desenvolvido durante meus estudos e nasceu do desejo de compartilhar conhecimento com a comunidade e com futuros colegas de profissão.

Compreendo que já existem documentações oficiais muito mais completas, porém algo que me ajuda bastante a aprender é **reescrever os conteúdos com minhas próprias palavras**. Às vezes, ler algo explicado de forma mais simples pode ser uma porta de entrada — ou um complemento — para materiais mais técnicos e detalhados.

Agradeço por dedicarem um tempo à leitura e fico aberta a **correções, sugestões e conselhos de estudo**.

Desejo uma ótima caminhada a todos nós! 🚀

Sumário

1. Funções Básicas e de Agregação.....	4
2. Funções de Filtro e Contexto.....	7
3. Funções de Tempo e Datas.....	10
4. Funções Lógicas e de Texto.....	14
5. Funções Estatísticas e Ranking.....	18
6. Funções Avançadas.....	22
7. Guia de Escolha de Gráficos no Power BI.....	27
8. Dicas Extras.....	28
9. Ferramentas Gratuitas para Dashboards Interativos.....	29
• Google Looker Studio (antigo Data Studio).....	29
• Microsoft Power BI Desktop (versão gratuita).....	30
• Softr.....	30
• Displayr.....	31
• Ubiq Dashboard Creator.....	31
• InfoCaptor.....	32
10. Considerações Finais.....	33

Dicionário DAX — Parte 1: Funções Básicas e de Agregação

♦ SUM

O que faz: soma os valores de uma coluna numérica.

Exemplo:

```
Total Vendas = SUM(Tabela[Vendas])
```

Soma todos os valores da coluna **Vendas**.

♦ AVERAGE

O que faz: calcula a média dos valores de uma coluna.

Exemplo:

```
Média Vendas = AVERAGE(Tabela[Vendas])
```

Retorna a média das vendas.

♦ COUNT

O que faz: conta quantas linhas têm valores **não vazios** na coluna.

Exemplo:

```
Total Clientes = COUNT(Tabela[ClienteID])
```

Conta quantos clientes existem com um ID válido.

♦ COUNTROWS

O que faz: conta o número total de linhas em uma tabela (mesmo que tenha valores vazios).

Exemplo:

```
Qtd Linhas = COUNTROWS(Tabela)
```

Conta todas as linhas da tabela.

♦ MAX / MIN

O que faz: retorna o maior (**MAX**) ou menor (**MIN**) valor da coluna.

Exemplo:

Maior Venda = MAX(Tabela[Vendas])

Menor Venda = MIN(Tabela[Vendas])

♦ DISTINCTCOUNT

O que faz: conta quantos valores únicos existem em uma coluna.

Exemplo:

Clientes Únicos = DISTINCTCOUNT(Tabela[ClienteID])

Conta quantos clientes diferentes fizeram compras.

♦ DIVIDE

O que faz: faz divisão com proteção contra erro (evita erro por divisão por zero).

Exemplo:

Lucro % = DIVIDE(Tabela[Lucro], Tabela[Receita])

Mostra o percentual de lucro.

(Diferente de Lucro / Receita, que dá erro se Receita for 0.)

♦ SUMX

O que faz: soma o resultado de uma **expressão linha por linha**.

Exemplo:

*Total Receita = SUMX(Tabela, Tabela[Preço] * Tabela[Quantidade])*

Multiplica **Preço × Quantidade** em cada linha e soma o total.

♦ AVERAGEX

O que faz: calcula a média de uma expressão linha a linha.

Exemplo:

Média de Lucro = AVERAGEX(Tabela, Tabela[Lucro])

♦ VAR

O que faz: cria uma variável dentro da medida — deixa os cálculos mais organizados.

Exemplo:

Lucro Médio =

VAR TotalLucro = SUM(Tabela[Lucro])

VAR TotalVendas = SUM(Tabela[Vendas])

RETURN DIVIDE(TotalLucro, TotalVendas)

Dicionário DAX — Parte 2: Funções de Filtro e Contexto

♦ CALCULATE

 **O que faz:** muda o contexto do cálculo aplicando filtros.

 **Exemplo:**

```
Vendas 2024 = CALCULATE(SUM(Tabela[Vendas]), Tabela[Ano] = 2024)
```

👉 Soma apenas as vendas de 2024.

É a **função mais importante do DAX** — praticamente tudo gira em torno dela.

♦ FILTER

 **O que faz:** cria um filtro personalizado para usar dentro do CALCULATE.

 **Exemplo:**

```
Vendas Altas =
```

```
CALCULATE(  
    SUM(Tabela[Vendas]),  
    FILTER(Tabela, Tabela[Vendas] > 500)  
)
```

👉 Soma só as vendas **maiores que 500**.

♦ ALL

 **O que faz:** Remove os filtros aplicados em uma tabela ou coluna.

 **Exemplo:**

```
Total Geral = CALCULATE(SUM(Tabela[Vendas]), ALL(Tabela))
```

👉 Ignora todos os filtros e mostra o total geral.

♦ ALLEXCEPT

🧠 **O que faz:** Remove todos os filtros **exceto** o de uma coluna.

💻 **Exemplo:**

```
Vendas Por Categoria = CALCULATE(SUM(Tabela[Vendas]),  
ALLEXCEPT(Tabela, Tabela[Categoria]))
```

👉 Mantém o filtro de **Categoria**, mas ignora os outros.

♦ ALLSELECTED

🧠 **O que faz:** Considera apenas o que o usuário **selecionou** no relatório (slicer, gráfico etc.).

💻 **Exemplo:**

```
% Selecionado =  
  
DIVIDE(  
  
    SUM(Tabela[Vendas]),  
  
    CALCULATE(SUM(Tabela[Vendas]), ALLSELECTED(Tabela))  
  
)
```

👉 Mostra o **percentual de vendas** em relação ao total **selecionado** pelo usuário.

♦ REMOVEFILTERS


🧠 **O que faz:** igual ao ALL, mas mais moderno.

💻 **Exemplo:**

```
Total Sem Filtro = CALCULATE(SUM(Tabela[Vendas]),  
REMOVEFILTERS(Tabela))
```

👉 Remove todos os filtros de forma clara e legível.

♦ VALUES


 **O que faz:** Retorna uma lista de valores únicos de uma coluna (usada com CALCULATE e outras funções).

 **Exemplo:**

```
Clientes Ativos = COUNTROWS(VALUES(Tabela[Cliente]))
```

👉 Conta quantos clientes **únicos** aparecem no contexto atual.

♦ EARLIER

 **O que faz:** Acessa o valor de uma linha anterior em um cálculo de linha (mais avançado).

 **Exemplo:**

```
Diferença = Tabela[Valor] - EARLIER(Tabela[Valor])
```

👉 Usado para comparar linhas em colunas calculadas.

♦ HASONEVALUE

 **O que faz:** Verifica se há **apenas um valor filtrado** em uma coluna.

 **Exemplo:**

```
Teste = IF(HASONEVALUE(Tabela[Categoria]), "Uma Categoria",  
"Várias Categorias")
```

👉 Retorna texto diferente se o usuário tiver filtrado uma categoria só.

♦ KEEPFILTERS

 **O que faz:** mantém filtros existentes **e adiciona novos** (sem sobrescrever).

 **Exemplo:**

```
Vendas Premium =  
  
CALCULATE(  
  
    SUM(Tabela[Vendas]),  
  
    KEEPFILTERS(Tabela[Tipo] = "Premium")  
)
```

👉 Mantém os filtros já aplicados e adiciona mais um.

Dicionário DAX — Parte 3: Funções de Tempo e Datas

♦ YEAR


 **O que faz:** Retorna o **ano** de uma data.

 **Exemplo:**

```
Ano = YEAR(Tabela[Data])
```

👉 Se a data for 2025-10-16, o resultado será 2025.

♦ MONTH

 **O que faz:** Retorna o **número do mês** (1 a 12).

 **Exemplo:**

```
Mês = MONTH(Tabela[Data])
```

👉 Se a data for 2025-10-16, retorna 10.

♦ DAY

 **O que faz:** Retorna o **dia** da data.

 **Exemplo:**

```
Dia = DAY(Tabela[Data])
```

👉 Se a data for 2025-10-16, retorna 16.

♦ TODAY

 **O que faz:** Retorna a **data atual**.

 **Exemplo:**

```
Data Atual = TODAY()
```

👉 Sempre mostra o dia de hoje (automático no Power BI).

♦ NOW


 **O que faz:** Retorna **data e hora atual**.

 **Exemplo:**

DataHora = NOW()

♦ DATE

 **O que faz:** Cria uma data a partir de ano, mês e dia.


 **Exemplo:**

Data = DATE(2025, 10, 16)

 Cria uma data manualmente.

♦ DATEDIFF


 **O que faz:** Calcula a **diferença entre duas datas** (em dias, meses, anos, etc.).

 **Exemplo:**

Diferença = DATEDIFF(Tabela[Data_Início], Tabela[Data_Fim], DAY)

 Mostra a diferença em dias.

♦ YEARFRAC

 **O que faz:** Calcula a diferença entre duas datas em **anos decimais**.

 **Exemplo:**

Idade = YEARFRAC(Tabela[Data_Nascimento], TODAY())

 Mostra a idade exata em anos (ex: 23,5).


♦ SAMEPERIODLASTYEAR

 **O que faz:** Retorna o mesmo período do **ano anterior**.


 **Exemplo:**

Vendas Ano Passado =

```
CALCULATE(SUM(Tabela[Vendas]),  
SAMEPERIODLASTYEAR(Tabela[Data]))
```

 Mostra as vendas do mesmo mês no ano anterior (ótimo pra comparar desempenho).

♦ DATEADD

 **O que faz:** Desloca um período pra frente ou para trás (em dias, meses, anos, etc.).

 **Exemplo:**

Vendas Mês Passado =

```
CALCULATE(SUM(Tabela[Vendas]), DATEADD(Tabela[Data], -1, MONTH))
```

 Soma as vendas do mês anterior.

♦ PREVIOUSMONTH / NEXTMONTH


 **O que faz:** Retorna as datas do mês anterior ou seguinte.

 **Exemplo:**

Vendas Mês Anterior =

```
CALCULATE(SUM(Tabela[Vendas]), PREVIOUSMONTH(Tabela[Data]))
```

♦ TOTALYTD

 **O que faz:** Soma do início do ano até a data atual (Year-To-Date).

 **Exemplo:**


Vendas Acumuladas Ano =

TOTALYTD(SUM(Tabela[Vendas]), Tabela[Data])

 Mostra o acumulado do ano.

♦ TOTALMTD

 **O que faz:** Soma do início do mês até a data atual (Month-To-Date).

 **Exemplo:**

Vendas Acumuladas Mês =

TOTALMTD(SUM(Tabela[Vendas]), Tabela[Data])

♦ TOTALQTD

 **O que faz:** Soma do início do trimestre até a data atual (Quarter-To-Date).

 **Exemplo:**

Vendas Trimestre =


TOTALQTD(SUM(Tabela[Vendas]), Tabela[Data])

Essas funções de tempo são **muito usadas juntas com gráficos de linha e cartões no Power BI**, especialmente para mostrar:

- crescimento em relação ao mês/ano anterior,
- totais acumulados (YTD, MTD),
- evolução de vendas ou despesas ao longo do tempo.

Dicionário DAX — Parte 4: Funções Lógicas e de Texto

♦ IF


 **O que faz:** Executa uma condição — se for **verdadeira**, faz uma coisa; se for **falsa**, faz outra.

 **Exemplo:**

```
Classificação = IF(Tabela[Vendas] > 500, "Alta", "Baixa")
```

👉 Se as vendas forem maiores que 500, retorna “Alta”; senão, “Baixa”.

♦ SWITCH

 **O que faz:** Funciona como vários “IF” seguidos, de forma mais limpa.

 **Exemplo:**

```
Categoria =
```

```
SWITCH(  
    TRUE(),  
    Tabela[Vendas] < 100, "Baixa",  
    Tabela[Vendas] < 500, "Média",  
    "Alta"  
)
```

👉 Classifica as vendas em faixas.

♦ AND / OR

 **O que faz:** Combina duas condições.

 **Exemplo:**

```
Condição = IF(Tabela[Vendas] > 500 && Tabela[Lucro] > 100,  
"Bom", "Ruim")
```

👉 Só retorna “Bom” se **as duas condições** forem verdadeiras (&& = “e”, || = “ou”).

♦ NOT


 **O que faz:** Inverte o resultado de uma condição.

 **Exemplo:**

```
SemLucro = IF(NOT(Tabela[Lucro] > 0), "Prejuízo", "Lucro")
```

♦ ISBLANK

 **O que faz:** Verifica se o valor está **vazio (em branco)**.

 **Exemplo:**

```
Sem Dados = IF(ISBLANK(Tabela[Vendas]), "Sem Informação",  
Tabela[Vendas])
```

👉 Mostra “Sem Informação” se a célula estiver vazia.


♦ CONCATENATE

 **O que faz:** Junta dois textos.

 **Exemplo:**

```
NomeCompleto = CONCATENATE(Tabela[Nome], " ", Tabela[Sobrenome])
```

♦ CONCATENATEX


 **O que faz:** Junta vários textos de uma tabela, com separador.

 **Exemplo:**

```
ListaProdutos = CONCATENATEX(Tabela, Tabela[Produto], ", ")
```

 Gera algo como: “Celular, Tablet, Notebook”.

♦ LEFT / RIGHT / MID

 **O que faz:** Extrai partes de um texto.

 **Exemplo:**

```
PrimeirasLetras = LEFT(Tabela[Nome], 3)
```

 Se o nome for “Karin”, retorna “Kar”.

♦ LEN


 **O que faz:** Conta quantos caracteres tem o texto.

 **Exemplo:**

```
QtdLetras = LEN(Tabela[Nome])
```


♦ UPPER / LOWER

 **O que faz:** Deixa o texto em **maiúsculas** ou **minúsculas**.

 **Exemplo:**

```
Maiúsculo = UPPER(Tabela[Nome])
```



♦ FORMAT

 **O que faz:** Formata números, datas ou textos com estilo.


 **Exemplo:**


```
DataFormatada = FORMAT(Tabela[Data], "dd/mm/yyyy")
```

```
LucroFormatado = FORMAT(Tabela[Lucro], "R$ #,##0.00")
```


 Mostra a data ou o valor em formato personalizado.

♦ SEARCH / FIND

 **O que faz:** Procura uma palavra dentro de um texto.

 **Exemplo:**

```
TemDesconto = IF(SEARCH("desconto", Tabela[Descrição],, 0) > 0,  
"Sim", "Não")
```


 Retorna "Sim" se encontrar a palavra "desconto" na descrição.

Essas funções são super úteis para:

- Categorizar dados automaticamente (IF e SWITCH)
- Criar colunas descritivas (CONCATENATE, FORMAT)
- Personalizar rótulos de gráficos (UPPER, LEFT, etc.)

Dicionário DAX — Parte 5: Estatísticas & Ranking


♦ RANKX

 **O que faz:** Atribui uma posição (rank) a cada linha, com base em um valor.

 **Exemplo:**


Ranking Vendas =

```
RANKX(  
  
    ALL(Tabela[Produto]),  
  
    SUM(Tabela[Vendas]),  
  
    ,  
  
    DESC  
  
)
```

 Ordena os produtos por vendas do maior para o menor, dando “1” para o que vendeu mais.

♦ TOPN


 **O que faz:** Retorna as **top N** linhas de uma tabela com base em um critério.

 **Exemplo:**

Top3 Produtos = TOPN(3, Tabela, Tabela[Vendas], DESC)


 Retorna os 3 produtos com maiores vendas.

♦ PERCENTILE.EXC / PERCENTILE.INC


 **O que faz:** Retorna o valor que separa uma porcentagem dos dados (percentil).

 **Exemplo:**

Percentil 90 = PERCENTILE.EXC(Tabela[Vendas], 0.9)

 O valor acima do qual estão os 10% maiores valores de vendas.


♦ MEDIAN


 **O que faz:** Retorna o valor do meio em uma lista de números (metade está acima, metade abaixo).

 **Exemplo:**

Mediana Vendas = MEDIAN(Tabela[Vendas])


♦ STDEV.P / STDEV.S


 **O que faz:** Calcula o desvio padrão (populacional ou amostral).

 **Exemplo:**

DesvioPadrao = STDEV.P(Tabela[Vendas])

♦ VAR.P / VAR.S

 **O que faz:** Calcula a variância (populacional ou amostral).

 **Exemplo:**

Variancia = VAR.P(Tabela[Vendas])

♦ SUMMARIZE

🧠 **O que faz:** Cria uma tabela agregada por categorias com expressões.

💻 **Exemplo:**

Resumo =

```
SUMMARIZE(  
    Tabela,  
    Tabela[Categoria],  
    "TotalVendas", SUM(Tabela[Vendas]),  
    "MédiaVendas", AVERAGE(Tabela[Vendas])  
)
```

👉 Retorna uma tabela com colunas: Categoria, TotalVendas, MédiaVendas.

♦ ADDCOLUMNS

🧠 **O que faz:** Adiciona colunas calculadas a uma tabela.


💻 **Exemplo:**

Tabela Nova =

```
ADDCOLUMNS(  
    Tabela,  
    "LucroPorcentagem", DIVIDE(Tabela[Lucro], Tabela[Receita])  
)
```

👉 Cria uma nova tabela com coluna "LucroPorcentagem".

♦ **EARLIER** (*já vimos na parte 2*)

 **O que faz:** Acessa valor da linha anterior num contexto de cálculo de linha — às vezes aparece em casos de ranking mais complexos.

Essas funções permitem que você faça coisas como:

- Mostrar quem são os **top 5 produtos** em vendas.
- Criar um ranking de clientes por valor gasto.
- Calcular percentis (ex: “quem está no top 10%”).
- Fazer análises estatísticas de dispersão (desvio padrão) para ver “quão regular são os valores de venda”.

Dicionário DAX — Parte 6: Funções Avançadas e “Trapaças” Inteligentes

♦ ISINSCOPE


 **O que faz:** Verifica em qual nível de hierarquia você está (útil em gráficos com drill down).

 **Exemplo:**

```
VerificaNivel =  
  
IF(  
  
    ISINSCOPE(Produtos[Categoria]),  
  
    "Categoria",  
  
    IF(  
  
        ISINSCOPE(Produtos[Subcategoria]),  
  
        "Subcategoria",  
  
        "Produto"  
  
    )  
  
)
```

 Mostra se o usuário está vendo Categoria, Subcategoria ou Produto.


♦ USERELATIONSHIP

 **O que faz:** Ativa uma **relação inativa** entre tabelas (ótimo em tabelas de datas com mais de uma relação).

 **Exemplo:**


VendasPorEntrega =

```
CALCULATE(
    SUM(Tabela[Vendas]),
    USERELATIONSHIP(Tabela[DataEntrega], Datas[Data])
)
```

 Usa a relação baseada na data de entrega, não na data de venda.

♦ CROSSFILTER

 **O que faz:** Muda a **direção de filtro** entre duas tabelas durante um cálculo.


 **Exemplo:**

ReceitaBidirecional =

```
CALCULATE(
    SUM(Vendas[Valor]),
    CROSSFILTER(Clientes[ID], Vendas[ClienteID], BOTH)
)
```

 Permite que o filtro flua nos dois sentidos (Clientes ↔ Vendas).

♦ TREATAS


 **O que faz:** Cria um filtro virtual entre tabelas **sem precisar de relacionamento físico**.


 **Exemplo:**

```
FiltroVirtual =  
  
CALCULATE(  
  
    SUM(Vendas[Valor]),  
  
    TREATAS(Valores[Região], Vendas[Região])  
  
)
```

 Aplica o filtro da tabela “Valores” sobre “Vendas”, mesmo sem relação direta.

♦ GENERATE / GENERATEALL


 **O que faz:** Combina tabelas, como se fosse um “CROSS JOIN”.

 **Exemplo:**

```
Combinacao = GENERATE(  
  
    Tabela1,  
  
    Tabela2  
  
)
```

 Cria uma tabela com todas as combinações entre Tabela1 e Tabela2.

♦ CONCATENATEX

 **O que faz:** Junta valores de uma coluna em **uma única célula**, com separador.


 **Exemplo:**

```
ListaProdutos =  
  
CONCATENATEX(  
  
    VALUES(Produtos[Nome]),  
  
    Produtos[Nome],  
  
    ", "  
  
)
```


 Retorna algo como: "Produto A, Produto B, Produto C".

♦ HASONEVALUE / SELECTEDVALUE

 **O que faz:** Verifica se há **um único valor selecionado** (usado para evitar erros).

 **Exemplo:**

```
CategoriaSelecionada =  
  
IF(  
  
    HASONEVALUE(Produtos[Categoria]),  
  
    SELECTEDVALUE(Produtos[Categoria]),  
  
    "Várias Categorias"  
  
)
```

 Mostra o nome da categoria se apenas uma estiver selecionada.

♦ SWITCH

🧠 **O que faz:** Substitui vários IFs — mais limpo e elegante.

💻 **Exemplo:**

```
Classificação =  
  
SWITCH(  
  
    TRUE(),  
  
    [Lucro] > 100000, "Excelente",  
  
    [Lucro] > 50000, "Bom",  
  
    [Lucro] > 10000, "Regular",  
  
    "Baixo"  
  
)
```

♦ **FORMAT**

🧠 **O que faz:** Formata números, datas e textos.

💻 **Exemplo:**

```
DataFormatada = FORMAT(Tabela[Data], "dd/mm/yyyy")
```

👉 Exibe a data no formato brasileiro.

♦ **UNION / INTERSECT / EXCEPT**

🧠 **O que faz:** Operações entre tabelas — unir, cruzar ou excluir linhas.


💻 **Exemplo:**

```
TabelaUnida = UNION(TabelaA, TabelaB)
```

Guia Rápido — Como Escolher o Gráfico Certo no Power BI


1. Colunas e Barras

Use para comparar categorias (produtos, regiões, meses).

- **Eixo X:** Categorias (ex: Produto, Mês, Região)
- **Eixo Y:** Valor numérico (ex: Vendas, Lucro, Quantidade)
 Dica: use **coluna vertical** para evolução no tempo e **barra horizontal** para comparação direta.


2. Linhas

Use para mostrar **tendência ao longo do tempo**.

- **Eixo X:** tempo (Ano, Mês, Dia)
- **Eixo Y:** valor contínuo (Vendas, Lucro, Média)
 Dica: use uma linha por categoria (ex: uma linha para cada produto).


3. Pizza e Rosca

Use para mostrar **proporções** de um todo.

- **Exemplo:** porcentagem de vendas por categoria.
 Dica: use **poucas fatias** (no máximo 5). Evite se os valores forem muito próximos.

4. Gráfico de Área

Use para mostrar **mudança ao longo do tempo + volume total**.

- **Eixo X:** tempo
- **Eixo Y:** valor acumulado
 Dica: ideal para mostrar crescimento ou tendência geral.

◆ 5. Dispersão (Scatter plot)

Use para mostrar **relação entre duas variáveis numéricas**.

- **Eixo X:** variável 1 (ex: Receita)
- **Eixo Y:** variável 2 (ex: Lucro)
🔮 Dica: use tamanho ou cor do ponto para representar uma terceira variável (ex: região).

🧩 6. Matriz e Tabela

Use para de **detalhes e valores exatos**.

🔮 Dica: ideal para dashboards administrativos; combine com filtros e segmentação.

🌐 7. Mapa

Use para Houver dados geográficos (país, estado, cidade, coordenadas).

🔮 Dica: verifique ortografia dos nomes para evitar pontos duplicados.

💡 Dicas Extras:

- Prefira **cores neutras + 1 cor destaque**.
- Dê um título **claro** e use rótulos apenas quando realmente ajuda.
- Sempre **verifique o contexto do filtro** antes de tirar conclusões (ex: períodos diferentes).
- **Consistência é Chave:** Evite misturar muitos tipos de gráficos na mesma página.
- **Títulos e Rótulos:** Os títulos do gráfico devem ser **claros e descritivos**. Use rótulos de dados (valores nas barras/linhas) apenas quando for estritamente necessário para a leitura, para não poluir o visual.
- **Verifique o Contexto:** Antes de tirar conclusões, **sempre verifique** se os filtros da página e o contexto das suas medidas DAX estão funcionando como esperado.



Ferramentas Gratuitas para Dashboards Interativos

(feito por uma estudante de Análise de Dados em desenvolvimento)

Enquanto estudava Análise de Dados, percebi que nem sempre temos acesso às mesmas oportunidades.

Algumas ferramentas pedem e-mails institucionais que muitas faculdades não oferecem, e outras são pagas — o que acaba limitando quem está começando.

Mas isso **não significa que não podemos treinar e desenvolver nossos projetos.**

Por isso, reuni aqui algumas **plataformas acessíveis para criar dashboards interativos** e explorar visualizações de dados, sem precisar pagar assinaturas ou depender de licenças.

Este material foi pensado especialmente para **quem está aprendendo, testando e construindo seu portfólio aos poucos**, assim como eu.



Google Looker Studio (antigo Data Studio)

Descrição: Ferramenta gratuita do Google para criação de relatórios e dashboards interativos.

Funções/Recursos:

- **CASE WHEN:** Cria categorias personalizadas, como:
CASE WHEN Vendas > 1000 THEN "Alta" ELSE "Baixa" END
- **REGEXP_MATCH():** Filtra dados com expressões regulares.
- **Date functions:** Como YEAR(), MONTH(), DATETIME_DIFF() para manipular datas.
- **Filtros interativos e controles de data.**
Dica: Use o Google Sheets como base de dados para praticar a construção de painéis dinâmicos.
Indicado para: Estudantes e analistas iniciantes que querem praticar

visualização de dados de forma simples e gratuita.

Microsoft Power BI Desktop (versão gratuita)

Descrição: Aplicativo gratuito para criar relatórios e dashboards profissionais.

Funções/Recursos:

- **DAX (Data Analysis Expressions):**
 - CALCULATE() → altera o contexto de filtro para criar medidas.
 - IF(), SWITCH() → criam condições lógicas.
 - SUMX(), AVERAGEX() → iteram sobre tabelas.
 - RELATED() → busca dados de outra tabela relacionada.
- **Power Query:** limpeza de dados com transformações como **Remover Duplicatas, Mesclar Consultas, Extrair Texto**.

Dica: pratique fórmulas simples de DAX para entender a lógica de medidas e contexto.

Indicado para: estudantes que desejam aprender recursos mais avançados e criar painéis completos.

Softr

Descrição: Plataforma *no-code* que cria sites e dashboards interativos a partir de planilhas.

Funções/Recursos:

- **Filtros personalizados:** Permitem mostrar apenas registros com base em categorias.
- **Blocos de gráfico:** Exibe dados diretamente de uma planilha.
- **Cálculos automáticos:** Soma, média e contagem de registros vindos do Google Sheets.
- **Condições de visibilidade:** exibe certos elementos apenas quando critérios são atendidos.

Dica: use para criar um painel público mostrando resultados de estudos ou comparações entre conjuntos de dados.

Indicado para: quem quer praticar visualização e design sem precisar programar.

Displayr

Descrição: Ferramenta que combina análise de dados e criação de dashboards interativos online.

Funções/Recursos:

- **Fórmulas de transformação:** log(), mean(), sum() e ifelse().
- **Gráficos dinâmicos** conectados a tabelas de dados.
- **Criação de variáveis derivadas:** para cálculos de percentuais e médias.
- **Filtros condicionais:** permitem comparar segmentos específicos (ex: região, faixa etária).

Dica: explore o painel de “Transformations” para criar colunas calculadas sem precisar usar código.

Indicado para: estudantes que querem praticar estatística e visualização combinadas.

Ubiq Dashboard Creator

Descrição: Ferramenta online que conecta-se a bancos de dados MySQL, PostgreSQL e arquivos CSV para criar dashboards.

Funções/Recursos:

Consultas SQL diretas:

SELECT region, SUM(sales) AS Total_Sales

FROM vendas

GROUP BY region;

- **Filtros de data e pesquisa interativa.**
- **Cálculos automáticos:** média, soma, contagem e percentuais.
- **Visualização em tempo real** (auto atualização de dados).
- **Dica:** pratique consultas SQL básicas e veja como elas se refletem em gráficos automaticamente.
- **Indicado para:** estudantes que estão aprendendo SQL e querem aplicá-lo em dashboards.

InfoCaptor

Descrição: Plataforma para criar dashboards com foco em integração de dados e relatórios analíticos.

Funções/Recursos:

- **SQL Editor integrado** para consultas diretas.
- **Funções de agregação:** SUM(), AVG(), MAX(), MIN().
- **KPI Widgets:** criam indicadores de desempenho rapidamente.
- **Filtros cruzados** entre gráficos.
- **Dica:** pratique unindo dados de fontes diferentes (por exemplo, CSV + MySQL) para entender integrações reais.
- **Indicado para:** quem quer aprender sobre fluxos de dados e dashboards corporativos.

Considerações Finais

Este mini dicionário começou como uma forma prática de consolidar meu aprendizado em **DAX e visualização de dados no Power BI**, e foi ampliado para incluir **outras ferramentas gratuitas** que também possibilitam o desenvolvimento de dashboards e análises interativas.

A ideia é reunir em um só lugar **recursos acessíveis para quem está aprendendo e construindo seu portfólio**, mostrando que existem diversas maneiras de praticar e evoluir na área de dados.

Aprender é um processo contínuo — e cada projeto, por menor que pareça, é um passo em direção ao crescimento.

Sigamos estudando, explorando e compartilhando conhecimento. 🌻