

Power BI do Básico ao Avançado

Apostila completa do curso — 14 módulos, 52 aulas e 7 estudos de caso aplicados a Vendas, Financeiro, RH, Logística, Marketing, Saúde e Indústria.

Power Query e linguagem M · Modelagem dimensional · Inteligência temporal · DAX do essencial ao avançado · Design e interatividade de dashboards · RLS e temas JSON · Performance e governança.



Sumário

01 Introdução ao BI e ao Power BI

- O que é Business Intelligence (e o que não é)
- O ecossistema Power BI
- O fluxo de trabalho de todo projeto
- Configurando o Desktop do jeito certo

02 Conexão a dados

- Excel, CSV e o conector Pasta
- Bancos de dados: SQL Server, PostgreSQL e amigos
- Web, APIs e JSON
- Import, DirectQuery e modo Composto

03 Power Query: o ETL do Power BI

- A lógica dos passos aplicados
- Tipos, nulos e erros
- Unpivot, pivot e o formato analítico
- Colunas condicionais, personalizadas e agrupamento

04 Power Query avançado e linguagem M

- Mesclar x Acrescentar: junções na prática
- Parâmetros e funções personalizadas
- Linguagem M essencial
- Performance de atualização

05 Modelagem dimensional e esquema estrela

- Fato e dimensão: a decisão que define o projeto
- Esquema estrela x floco de neve x tabela única
- Relacionamentos: cardinalidade e direção do filtro
- Os erros de modelagem que mais custam caro

06 Calendário e inteligência temporal

- Criando a dCalendario definitiva
- YTD, MTD, QTD e acumulados
- Comparativos: YoY, MoM e período anterior
- Calendário fiscal, semanas e períodos customizados

07 DAX essencial: contexto é tudo

- Contexto de filtro e contexto de linha
- Coluna calculada x medida x tabela calculada
- Agregações, DIVIDE e variáveis
- CALCULATE: a função central do DAX

08 DAX intermediário: filtros, relações e ranking

- FILTER e a família ALL
- VALUES, DISTINCT, RELATED e RELATEDTABLE
- RANKX, TOPN e análise de concentração

Padrões de medida que resolvem 80% dos pedidos

09 **DAX avançado: iteradores, tabelas virtuais e metas**

Iteradores: a família X

Tabelas virtuais: ADDCOLUMNS, SUMMARIZE e GENERATE

TREATAS, medidas dinâmicas e segmentação por medida

Metas, atingimento e semáforos

10 **Design de dashboards que convencem**

Escolhendo o gráfico certo

Hierarquia visual e layout em grid

Cor com propósito e acessibilidade

Storytelling com dados

11 **Visuais avançados e interatividade**

Interações, drill-down e drill-through

Indicadores (bookmarks), botões e navegação

Tooltips de página e visuais especiais

12 **Formatação condicional, RLS e temas JSON**

Formatação condicional avançada

Segurança em nível de linha (RLS)

Temas JSON corporativos

13 **Performance e boas práticas**

Medindo antes de otimizar

Reduzindo o tamanho do modelo

Checklist profissional antes de publicar

14 **Publicação, colaboração e governança**

Workspaces, aplicativos e permissões

Atualização, gateway e fluxos de dados

Ciclo de vida, pipelines e governança

Estudos de caso

Painel Comercial de uma rede varejista com 42 lojas — Vendas e Comercial

DRE gerencial, fluxo de caixa e orçado x realizado — Financeiro e Controladoria

Painel de People Analytics: headcount, turnover e clima — RH e Gestão de Pessoas

Torre de controle logística: OTIF, estoque e custo de frete — Logística e Supply Chain

Funil completo: da mídia paga ao LTV do cliente — Marketing e Growth

Gestão hospitalar: ocupação, tempo de permanência e qualidade assistencial — Saúde

Chão de fábrica: OEE, manutenção e aderência ao plano — Indústria e Manufatura

Introdução ao BI e ao Power BI

O que é Business Intelligence de verdade, como o Power BI se encaixa no ecossistema Microsoft e qual é o fluxo de trabalho que você vai repetir em todo projeto.

Objetivos de aprendizagem

- Entender o papel do BI dentro da empresa e a diferença entre relatório, dashboard e análise
- Conhecer Power BI Desktop, Service, Mobile, Report Builder e Gateway
- Dominar o fluxo Extrair → Transformar → Modelar → Medir → Visualizar → Publicar
- Configurar o Desktop com as opções que evitam dor de cabeça no futuro

O que é Business Intelligence (e o que não é)

A diferença entre dado, informação e decisão — e por que a maioria dos dashboards falha.

Business Intelligence é o conjunto de práticas que transforma dado bruto em decisão. Repare na palavra decisão: se um painel é bonito, rápido e ninguém muda nada por causa dele, o projeto de BI falhou. Todo o curso parte dessa régua.

A escada do dado

Nível	Exemplo	Pergunta que responde
Dado	Nota fiscal 8842, R\$ 1.240, 12/03	O que aconteceu, em bruto?
Informação	Vendas de março: R\$ 2,4 mi	Qual o número consolidado?
Indicador	Vendas março: R\$ 2,4 mi (92% da meta)	Está bom ou ruim?
Insight	A queda vem de 3 clientes da região Sul	Por que está assim?
Decisão	Redirecionar 2 vendedores para o Sul	O que eu faço agora?

BOA PRÁTICA

Antes de abrir o Power BI, escreva as 5 perguntas que o painel precisa responder. Se você não consegue escrever, ainda não é hora de construir.

Relatório, dashboard e análise

- Relatório operacional: lista detalhada, usado para conferir e agir no dia a dia (ex.: pedidos em atraso).
- Dashboard gerencial: poucos indicadores, comparados contra meta e período anterior, leitura em 10 segundos.
- Análise exploratória: superfície ampla com filtros e drill, usada para investigar causa.

O erro clássico é misturar os três na mesma página. Um diretor abre o painel, encontra uma tabela com 40 colunas e fecha. Separe por público: uma página por perfil de usuário.

Os papéis em um projeto de BI

- Patrocinador: define a pergunta de negócio e aprova o indicador.
- Analista de BI (você): modela, mede e visualiza.
- Engenheiro de dados: garante a origem confiável e atualizada.
- Usuário final: consome e, principalmente, decide.

ATENÇÃO

É definir mal o indicador. Faturamento é com ou sem imposto? Devolução entra? A conta certa sobre a definição errada continua errada. Documente a regra junto do patrocinador antes de codar.

EXERCÍCIO PRÁTICO

Escolha uma área da sua empresa (ou uma fictícia). Escreva as 5 perguntas de negócio que um painel deveria responder e, ao lado de cada uma, o indicador que a responde e quem toma a decisão com base nele.

O ecossistema Power BI

Desktop, Service, Mobile, Gateway, Report Builder e Fabric — o que é cada peça.

Power BI não é um programa, é uma plataforma. Confundir as peças é a causa nº 1 de frustração de iniciante ('publiquei e não atualiza', 'meu colega não consegue ver').

Peça	Onde roda	Para que serve
Power BI Desktop	Windows, gratuito	Construir: ETL, modelo, DAX e visuais. É aqui que 90% do trabalho acontece.
Power BI Service	Nuvem (app.powerbi.com)	Publicar, compartilhar, agendar atualização, criar apps e dashboards.
Power BI Mobile	iOS / Android	Consumo em campo, com layout de telefone próprio.
On-premises Gateway	Servidor da empresa	Ponte entre a nuvem e bancos locais para atualização automática.
Report Builder	Windows	Relatórios paginados (paginated), ideal para imprimir/exportar em PDF com layout fixo.
Microsoft Fabric	Nuvem	Plataforma de dados maior: Lakehouse, Warehouse, Data Factory, com Power BI dentro.

Licenças, em português claro

- Free: usa o Desktop à vontade e publica no seu espaço pessoal. Não compartilha com outras pessoas.

- Pro: por usuário. Permite compartilhar em workspaces. É o mais comum em empresas pequenas e médias.
- PPU (Premium Per User): Pro + recursos premium (modelos maiores, atualização mais frequente, deployment pipelines).
- Capacidade (F/P SKU): a empresa paga a capacidade; quem só consome pode ser Free.

NOTA

No vocabulário da Microsoft, 'relatório' é o arquivo com várias páginas que você faz no Desktop. 'Dashboard' é uma tela única do Service feita de blocos fixados de vários relatórios. No mercado, porém, todo mundo chama o relatório de dashboard — e neste curso usamos os dois termos com clareza quando importa.

Como as peças se conectam

1. Você conecta o Desktop às fontes e constrói o modelo.
2. Publica o arquivo .pbix em um workspace do Service.
3. No Service, configura credenciais e agendamento de atualização (via Gateway, se a fonte for local).
4. Distribui via App ou compartilhamento direto do workspace.
5. Usuários consomem no navegador ou no app mobile.

EXERCÍCIO PRÁTICO

Instale o Power BI Desktop e crie uma conta no Power BI Service. Publique um arquivo vazio em um workspace novo chamado 'Laboratório' e confirme que você o enxerga no navegador.

O fluxo de trabalho de todo projeto

As seis etapas que você vai repetir em cada painel, do primeiro ao centésimo.

Todo projeto de Power BI, sem exceção, segue o mesmo caminho. Quem pula etapa paga depois — geralmente em forma de DAX impossível de escrever porque o modelo está errado.

1. Extrair — conectar às fontes (Excel, SQL, API, pasta, SharePoint).
2. Transformar — limpar no Power Query: tipos, nulos, colunas, pivot/unpivot, mesclagens.
3. Modelar — separar fatos e dimensões, criar relacionamentos em esquema estrela, criar a tabela calendário.
4. Medir — escrever medidas DAX: base, comparativas, metas, variações.
5. Visualizar — escolher o gráfico certo, aplicar hierarquia visual, criar navegação.
6. Publicar e governar — workspace, atualização agendada, RLS, documentação.

ATENÇÃO

Iniciantes gastam 80% do tempo escolhendo cores. Profissionais gastam 80% no Power Query e na modelagem — e por isso o DAX deles fica curto e rápido.

Onde resolver cada problema

Problema	Resolva em	Por quê
Coluna com data em texto	Power Query	Transformação estrutural, feita uma vez na carga.
Preciso de uma coluna 'Faixa de Preço'	Power Query (ou coluna calculada)	Valor fixo por linha, usado para filtrar e agrupar.
Total de vendas do período filtrado	DAX (medida)	Depende do contexto de filtro do visual.
Unir vendas de 12 arquivos mensais	Power Query (conector Pasta)	ETL puro, escala sozinho quando chega o 13º arquivo.
Comparar com o mesmo mês do ano passado	DAX (inteligência temporal)	Precisa recalcular conforme o usuário navega.

BOA PRÁTICA

Se dá para fazer na origem, faça na origem. Se não, faça no Power Query. Só use DAX quando o resultado depender do filtro que o usuário aplicar.

EXERCÍCIO PRÁTICO

Liste 6 transformações que o seu conjunto de dados precisaria e classifique cada uma em: origem, Power Query ou DAX. Justifique em uma frase cada escolha.

Configurando o Desktop do jeito certo

As opções que você deve mudar antes do primeiro projeto para não sofrer depois.

Opções essenciais (Arquivo → Opções e configurações → Opções)

- Carregamento de dados → desmarque 'Detecção automática de novas relações'. Relacionamento é decisão sua, não da ferramenta.
- Carregamento de dados → desmarque 'Data/hora automática'. Ela cria uma tabela de datas oculta para cada campo de data e incha o modelo.
- Configurações regionais → defina o Locale do arquivo (Português-Brasil) para que datas e decimais sejam interpretados corretamente.
- Recursos de versão prévia → ative apenas o que você vai usar de fato.
- Segurança de privacidade → entenda os níveis; 'Sempre ignorar' acelera a atualização, mas só use em fontes confiáveis.

ATENÇÃO

Com ela ligada, um modelo com 10 colunas de data ganha 10 tabelas de calendário ocultas. O arquivo cresce, a atualização demora e a inteligência temporal fica inconsistente. Desligue e crie UMA tabela calendário — você aprende no Módulo 6.

Organização de arquivos

- Uma pasta por projeto, com subpastas /dados, /pbix, /documentacao.
- Nome do arquivo com versão: Vendas_v03.pbix. Nunca 'final_final2.pbix'.

- Guarde um arquivo .pbix 'modelo' com tema, calendário e medidas-base prontos para reaproveitar.

Atalhos que economizam horas

Atalho	Ação
Ctrl + C / Ctrl + V em visuais	Duplica o visual mantendo formatação
Ctrl + Shift + C / V	Copia e cola apenas a formatação entre visuais
Alt + arrastar	Move o visual ignorando o encaixe na grade
Ctrl + clique	Ativa o drill/seleção múltipla em segmentações
F11	Modo de exibição de foco

EXERCÍCIO PRÁTICO

Configure seu Desktop conforme a lista acima, crie um arquivo 'Template.pbix' vazio com as opções ajustadas e salve-o em uma pasta padrão para reutilizar nos próximos módulos.

Quiz do módulo

1. Qual é a melhor definição de sucesso em um projeto de BI?

- O painel carregar em menos de 3 segundos
- **✓ Alguém tomar uma decisão diferente por causa do painel**
- Usar o máximo de visuais disponíveis
- Ter todos os dados da empresa em um só arquivo

BI existe para mudar decisão. Performance e beleza são meios, não fim.

2. Onde você deve corrigir uma coluna de data que veio como texto?

- Com DAX, em uma medida
- **✓ No Power Query**
- Em um visual de tabela
- No Power BI Service

É uma transformação estrutural: pertence ao Power Query (ou à origem).

3. Para que serve o On-premises Data Gateway?

- Compartilhar relatórios com usuários Free
- Criar relatórios paginados
- **✓ Permitir que o Service atualize dados de fontes locais da empresa**
- Aumentar o limite de linhas do modelo

O Gateway é a ponte segura entre a nuvem e as fontes que ficam dentro da rede da empresa.

4. Por que desligar a opção 'Data/hora automática'?

- Porque ela impede o uso de segmentações
- **✓ Porque cria uma tabela de calendário oculta por coluna de data, inchando o modelo**

- Porque ela é paga
- Porque impede a publicação no Service

Uma única tabela calendário explícita é mais leve, previsível e correta.

5. Qual etapa costuma consumir a maior parte do tempo em um projeto profissional?

- Escolha de cores
- **✓ Transformação e modelagem**
- Publicação
- Escrita do título das páginas

Modelo bem feito é o que deixa o DAX curto e o painel rápido.

6. Um diretor precisa ver 6 indicadores comparados com meta. O formato mais adequado é:

- Relatório operacional com tabela detalhada
- **✓ Dashboard gerencial**
- Análise exploratória com 20 filtros
- Exportação em CSV

Público executivo pede leitura em segundos: poucos indicadores, contra meta e período anterior.

Conexão a dados

Como trazer dados de Excel, CSV, pastas, SQL Server, web e APIs — e como escolher entre Importação, DirectQuery e modo Composto.

Objetivos de aprendizagem

- Conectar-se às fontes mais comuns do mercado corporativo
- Consolidar dezenas de arquivos com o conector Pasta
- Escolher conscientemente entre Import, DirectQuery e Composto
- Parametrizar caminhos e credenciais para que o projeto sobreviva à mudança de servidor

Excel, CSV e o conector Pasta

O caminho de 90% dos projetos que começam do zero — e como fazer isso escalar.

Excel ainda é a fonte mais comum do Brasil corporativo. O problema não é o Excel: é conectar-se à planilha do jeito frágil, apontando para uma aba solta com células mescladas.

Conectando a um Excel do jeito certo

1. Na origem, transforme o intervalo em Tabela (Ctrl+T) e dê um nome claro, como tbVendas.
2. No Power BI: Obter dados → Pasta de trabalho do Excel.
3. Selecione a Tabela nomeada, não a Planilha. Tabela é estável; planilha quebra quando alguém insere uma linha no topo.
4. Clique em Transformar Dados (nunca em Carregar direto) para revisar tipos.

ATENÇÃO

C:\Users\seu.nome\Desktop\vendas.xlsx só existe no seu computador. Ao publicar, a atualização falha. Use SharePoint, OneDrive ou uma pasta de rede acessível pelo Gateway.

O conector Pasta: consolidando 60 arquivos em 1 minuto

Quando a área manda um arquivo por mês, não abra 12 conexões. Use Obter dados → Pasta, aponte para o diretório e combine os arquivos. O Power Query cria automaticamente uma função de transformação aplicada a cada arquivo.

```
M
let
    Fonte = Folder.Files("\\servidor\financeiro\extratos"),
    SomenteCSV = Table.SelectRows(
        Fonte,
        each Text.EndsWith([Extension], ".csv") and not Text.StartsWith([Name], "~$")
    ),
    Combinado = Table.Combine(
        List.Transform(SomenteCSV[Content], each Csv.Document(_, [Delimiter=";", Encoding=65001])
    ))
in
    Combinado
```

BOA PRÁTICA

Arquivos temporários do Excel começam com ~\$ e quebram a combinação. Filtre Extension e Name antes do passo Combinar.

Armadilhas de CSV brasileiro

- Delimitador ponto-e-vírgula: defina no diálogo, não deixe o automático adivinhar.
- Codificação: use 65001 (UTF-8) ou 1252 (Windows) para não perder acentos.
- Decimal com vírgula: ajuste o Locale no passo 'Alterar tipo usando localidade'.
- Cabeçalho em linha 5: use 'Remover linhas superiores' antes de promover cabeçalhos.

EXERCÍCIO PRÁTICO

Crie 3 arquivos CSV mensais com a mesma estrutura em uma pasta. Conecte-se pelo conector Pasta, filtre por extensão e combine. Depois acrescente um 4º arquivo e confirme que ele entra sozinho ao atualizar.

Bancos de dados: SQL Server, PostgreSQL e amigos

Consulta nativa, dobramento de consulta e como não derrubar o banco da empresa.

Ao conectar a um banco, você deixa o mundo dos arquivos e entra no mundo da negociação com a TI. Duas regras: puxe só o que precisa e deixe o banco trabalhar.

SQL

```
SELECT
    v.id_venda,
    v.data_venda,
    v.id_cliente,
    v.id_produto,
    v.quantidade,
    v.valor_unitario,
    v.desconto
FROM dbo.fato_vendas v
WHERE v.data_venda >= DATEADD(YEAR, -3, GETDATE())
    AND v.status <> 'CANCELADA';
```

ATENÇÃO

Traz colunas que você não usa, aumenta o modelo, atrasa a atualização e quebra quando alguém renomeia uma coluna irrelevante. Liste as colunas.

Query folding (dobramento de consulta)

Quando você filtra, remove colunas ou agrupa no Power Query sobre uma fonte relacional, o Power Query tenta traduzir esses passos para SQL e mandar o trabalho para o banco. Isso é o query folding — e é a diferença entre atualizar em 40 segundos ou em 40 minutos.

- Mantém o folding: filtrar linhas, remover colunas, renomear, mesclar, agrupar, alterar tipo.
- Quebra o folding: adicionar coluna de índice, usar funções personalizadas em M, Table.Buffer, algumas operações de texto.

- Como verificar: clique com o botão direito no passo → 'Exibir consulta nativa'. Se estiver disponível, o folding está ativo até ali.

BOA PRÁTICA

Coloque os passos que quebram o folding sempre no FIM da consulta. Assim o banco faz o trabalho pesado e o Power Query só finaliza.

Credenciais e segurança

- Prefira autenticação do Windows/organizacional a usuário e senha fixos.
- Peça à TI um usuário de leitura dedicado ao BI, com acesso somente às views necessárias.
- Nunca escreva senha dentro do código M. Use o gerenciador de credenciais do Power BI.
- Combine uma janela de atualização fora do horário de pico do sistema transacional.

EXERCÍCIO PRÁTICO

Conecte-se a um banco (pode ser um SQL Express local ou um PostgreSQL gratuito na nuvem). Escreva uma consulta enxuta com filtro de data, importe e verifique com 'Exibir consulta nativa' se o folding está ativo após adicionar um filtro no Power Query.

Web, APIs e JSON

Puxando cotações, feriados e dados públicos direto da internet.

Muita análise fica melhor com dado externo: cotação do dólar, IPCA, feriados nacionais, CEP, indicadores setoriais. O Power Query lê tudo isso com Web.Contents.

```
M
let
    Url = "https://olinda.bcb.gov.br/olinda/servico/PTAX/versao/v1/odata/"
        & "CotacaoDolarPeriodo(dataInicial=@di,dataFinalCotacao=@df)"
        & "?@di='01-01-2023'&@df='12-31-2024'&$format=json",
    Fonte = Json.Document(Web.Contents(Url)),
    Valores = Fonte[value],
    Tabela = Table.FromList(Valores, Splitter.SplitByNothing(), null, null, ExtraValues.Error),
    Expandido = Table.ExpandRecordColumn(
        Tabela, "Column1",
        {"cotacaoCompra", "cotacaoVenda", "dataHoraCotacao"},
        {"Compra", "Venda", "DataHora"}
    ),
    Tipos = Table.TransformColumnTypes(
        Expandido,
        {"Compra", type number}, {"Venda", type number}, {"DataHora", type datetime}
    ),
    DataPura = Table.AddColumn(Tipos, "Data", each Date.From([DataHora]), type date)
in
    DataPura
```

ATENÇÃO

Quando você concatena a URL inteira em uma string, o Power BI Service pode não conseguir validar as credenciais na atualização. Use Web.Contents com RelativePath e Query sempre que a API permitir.

```

M
let
    Resposta = Json.Document(
        Web.Contents(
            "https://api.exemplo.com",
            [
                RelativePath = "v1/pedidos",
                Query = [ desde = "2024-01-01", limite = "500" ],
                Headers = [ #"Authorization" = "Bearer " & ChaveApi ]
            ]
        )
    )
in
    Resposta

```

Paginação

APIs sérias devolvem os dados em páginas. A técnica padrão é criar uma função que recebe o número da página e usar `List.Generate` ou `List.Transform` para percorrer todas.

```

M
let
    fnPagina = (pagina as number) as table =>
        let
            Resposta = Json.Document(
                Web.Contents("https://api.exemplo.com",
                    [ RelativePath = "v1/pedidos", Query = [ page = Text.From(pagina) ] ])
            ),
            Dados = Table.FromRecords(Resposta[data])
        in
            Dados,
    Paginas = List.Transform({1..20}, each fnPagina(_)),
    Tudo = Table.Combine(Paginas)
in
    Tudo

```

BOA PRÁTICA

Crie um parâmetro chamado `ChaveApi` no Power Query. Assim você troca a chave em um lugar só e não a expõe espalhada pelo código.

EXERCÍCIO PRÁTICO

Conecte-se a uma API pública (BrasilAPI de feriados é ótima para começar), expanda o JSON em tabela, trate os tipos e traga os feriados de 3 anos. Você vai reusar isso no Módulo 6.

Import, DirectQuery e modo Composto

A decisão de arquitetura que define a performance do seu projeto.

Critério	Importação	DirectQuery
Onde ficam os dados	Comprimidos na memória do modelo (VertiPaq)	Permanecem no banco de origem
Velocidade do visual	Muito rápida	Depende do banco e da rede

Critério	Importação	DirectQuery
Atualidade	Do último refresh	Tempo real
Recursos DAX	Todos	Limitados (várias funções de time intelligence restritas)
Volume	Limitado pelo tamanho do modelo	Praticamente ilimitado
Carga no banco	Só no refresh	A cada clique do usuário

BOA PRÁTICA

Comece sempre por Importação. Só vá para DirectQuery quando houver exigência real de tempo real ou volume que não cabe na memória. DirectQuery escolhido 'por precaução' costuma resultar em painel lento e usuário insatisfeito.

Modo Composto

O modo Composto permite misturar: as dimensões (Produto, Cliente, Calendário) ficam em Importação, rápidas para filtrar, e a tabela fato gigante fica em DirectQuery. Combinado com Agregações (Módulo 13), você entrega respostas instantâneas para os visuais resumidos e detalhe sob demanda.

Atualização incremental

Com Importação e um volume grande, você não precisa recarregar 5 anos toda noite. A atualização incremental recarrega apenas a janela recente.

1. Crie dois parâmetros do tipo data/hora com os nomes exatos RangeStart e RangeEnd.
2. Filtre a coluna de data da tabela fato usando esses parâmetros ($\geq$ RangeStart e $<$ RangeEnd).
3. Na modelagem, clique com o botão direito na tabela → Atualização incremental.
4. Defina, por exemplo: armazenar 5 anos, atualizar os últimos 10 dias.
5. Publique. O particionamento é criado no Service, não no Desktop.

ATENÇÃO

Atualização incremental exige query folding no filtro de data. Se a fonte for um arquivo CSV, não funciona como esperado — é recurso para fontes relacionais.

EXERCÍCIO PRÁTICO

Monte uma matriz de decisão para 3 cenários da sua realidade (ex.: relatório executivo mensal, painel de chão de fábrica em tempo real, análise histórica de 10 anos) e justifique o modo escolhido para cada um.

Quiz do módulo

1. Ao conectar a um Excel, o que você deve selecionar?

- A planilha inteira
- ✓ **A Tabela nomeada**

- Um intervalo de células manual
- O arquivo em PDF

Tabelas nomeadas são estáveis a inserções de linha e mudanças de layout.

2. O que é query folding?

- Compactar o arquivo .pbix
- **✓ Traduzir os passos do Power Query em SQL enviado ao banco**
- Dobrar a quantidade de linhas importadas
- Um tipo de visual

Com folding, o banco faz o trabalho pesado e a atualização fica muito mais rápida.

3. Onde colocar um passo que quebra o folding?

- No início da consulta
- **✓ No fim da consulta**
- Em uma consulta separada sempre
- Tanto faz

Quanto mais tarde ele acontecer, mais passos anteriores conseguem ser enviados ao banco.

4. Qual é o principal motivo para escolher DirectQuery?

- Deixar os visuais mais rápidos
- Poder usar todas as funções DAX
- **✓ Necessidade de dado em tempo real ou volume que não cabe na memória**
- Reduzir a carga no banco de dados

DirectQuery troca velocidade por atualidade e escala; ele aumenta a carga no banco.

5. Quais parâmetros a atualização incremental exige, com nomes exatos?

- DataInicio e DataFim
- Start e End
- **✓ RangeStart e RangeEnd**
- Inicio e Fim

Os nomes são fixos e sensíveis a maiúsculas: RangeStart e RangeEnd.

6. Por que usar RelativePath e Query no Web.Contents?

- Para deixar o código mais curto
- **✓ Para que a atualização agendada no Service consiga validar as credenciais**
- Para aumentar a velocidade da API
- É obrigatório em qualquer conexão web

URLs montadas por concatenação costumam falhar na validação de fonte de dados no Service.

Power Query: o ETL do Power BI

Limpeza, tipos, nulos, pivot/unpivot, colunas condicionais e as boas práticas que deixam sua consulta legível e resistente a mudança.

Objetivos de aprendizagem

- Dominar a interface e a lógica de passos aplicados
- Tratar tipos, nulos, erros e textos sujos com segurança
- Usar unpivot para transformar planilhas em tabelas analíticas
- Criar colunas condicionais e agrupamentos sem escrever código

A lógica dos passos aplicados

Por que o Power Query é uma receita, e não uma planilha.

O Power Query não edita dados: ele grava uma receita. Cada clique vira um passo na linguagem M, e essa receita é reexecutada do zero a cada atualização. Entender isso muda tudo — inclusive te dá coragem de errar, porque basta apagar o passo.

```
M
let
    Fonte = Excel.Workbook(File.Contents("C:\dados\vendas.xlsx"), null, true),
    tbVendas = Fonte[Item="tbVendas", Kind="Table"][Data],
    CabecalhoPromovido = Table.PromoteHeaders(tbVendas, [PromoteAllScalars=true]),
    TiposAjustados = Table.TransformColumnTypes(CabecalhoPromovido, {
        {"Data", type date},
        {"Cliente", type text},
        {"Quantidade", Int64.Type},
        {"ValorUnitario", type number}
    }),
    SemNulos = Table.SelectRows(TiposAjustados, each [Data] <> null),
    ColunaReceita = Table.AddColumn(SemNulos, "Receita",
        each [Quantidade] * [ValorUnitario], type number)
in
    ColunaReceita
```

BOA PRÁTICA

'Personalizar1', 'Tipo Alterado2' não dizem nada. Clique com o botão direito → Renomear e escreva o que o passo faz. Daqui a seis meses, você agradece.

Organização de consultas

- Crie grupos (pastas) na lista de consultas: 00_Parametros, 01_Fontes, 02_Dimensoes, 03_Fatos, 99_Auxiliares.
- Use 'Habilitar carga' apenas nas tabelas que vão para o modelo. Consultas intermediárias ficam desabilitadas.
- Referência x Duplicar: Referência cria um ponto de partida encadeado (recomendado). Duplicar copia todo o código (vira manutenção dobrada).

ATENÇÃO

O passo 'Tipo Alterado' gerado ao carregar lista as colunas pelo nome. Se a origem renomear uma coluna, a atualização quebra. Em fontes instáveis, considere remover esse passo automático e tipar explicitamente no fim.

EXERCÍCIO PRÁTICO

Abra uma consulta existente, renomeie todos os passos com nomes descritivos e organize as consultas em grupos. Depois desabilite a carga das consultas auxiliares e confira a redução no painel de campos.

Tipos, nulos e erros

O tratamento que separa um modelo confiável de um que quebra na sexta-feira à noite.

Tipos de dados

- Número Inteiro: chaves, quantidades. Mais leve que decimal.
- Número Decimal: valores monetários e percentuais.
- Número Decimal Fixo (moeda): 4 casas, evita erro de arredondamento em somas grandes.
- Data: sem hora. Sempre remova a hora da coluna que vai relacionar com o calendário.
- Texto: use só quando for texto mesmo. CPF e CEP são texto, não número.
- Verdadeiro/Falso: flags. Ocupa quase nada.

ATENÇÃO

Uma coluna DataHora com segundos pode ter milhões de valores distintos. Separe em Data (para relacionar) e Hora (se necessário). O modelo pode ficar várias vezes menor.

Localidade: o erro clássico brasileiro

Ao converter '13/05/2024' ou '1.234,56' com a localidade errada, você recebe Error em todas as linhas. Use 'Alterar tipo usando localidade' e escolha Português (Brasil).

```
M
Table.TransformColumnTypes(
    Etapa_Anterior,
    {
        {"DataEmissao", type date},
        {"ValorTotal", type number}
    },
    "pt-BR"
)
```

Nulos: três estratégias

Situação	Estratégia	Como
Valor numérico ausente = zero	Substituir por 0	Substituir Valores → null por 0
Categoria ausente	Substituir por 'Não informado'	Mantém a linha no gráfico e evita categoria em branco

Situação	Estratégia	Como
Linha inteira inválida	Remover	Filtrar a coluna-chave por 'não é nulo'

BOA PRÁTICA

Em DAX, SUM ignora nulos, mas AVERAGE também ignora — o que muda a média. Decida no Power Query se ausência significa zero ou 'desconhecido'. A escolha errada distorce o indicador.

Erros

1. Use 'Manter Erros' para inspecionar quais linhas falharam e por quê.
2. Corrija a causa (tipo, localidade, texto sujo) sempre que possível.
3. Só depois use 'Remover Erros' ou 'Substituir Erros' — e documente a decisão.
4. Crie uma consulta de auditoria com as linhas rejeitadas e publique-a junto: a área de negócio precisa saber o que ficou de fora.

M

```
Table.AddColumn(
    Etapa_Anterior,
    "ValorSeguro",
    each try Number.FromText([ValorTexto], "pt-BR") otherwise null,
    type nullable number
)
```

EXERCÍCIO PRÁTICO

Pegue um CSV com datas em texto e valores com vírgula decimal. Converta usando localidade pt-BR, crie uma consulta de auditoria com as linhas que geraram erro e trate os nulos das colunas numéricas e categóricas com estratégias diferentes.

Unpivot, pivot e o formato analítico

Como transformar a planilha 'de gerente' (uma coluna por mês) em tabela pronta para análise.

A planilha que a área te manda quase sempre tem uma coluna por mês: Produto | Jan | Fev | Mar... Isso é ótimo para ler e péssimo para analisar. O Power BI quer o formato longo: Produto | Mês | Valor.

Formato largo (origem)	Formato longo (analítico)
Produto A 100 120 90	Produto A Jan 100
Produto B 80 95 110	Produto A Fev 120
Uma coluna nova por mês novo	Nenhuma coluna nova: só linhas

Fazendo o unpivot

1. Selecione as colunas que devem permanecer (Produto, Categoria).

2. Clique com o botão direito → 'Transformar outras colunas em linhas' (Unpivot Other Columns).
3. Renomeie 'Atributo' para 'Mês' e 'Valor' para 'Receita'.
4. Converta o mês em data de referência, se precisar relacionar com o calendário.

BOA PRÁTICA

Se você selecionar as colunas de mês e fizer unpivot direto, o mês seguinte (Abr) que aparecer no arquivo não será incluído. Ao usar 'outras colunas', a consulta se adapta sozinha.

Outras transformações que você vai usar toda semana

- Dividir coluna: por delimitador, por número de caracteres, por transição de maiúscula/minúscula.
- Extrair: texto antes/depois/entre delimitadores — ótimo para códigos compostos como 'SP-VENDAS-0032'.
- Preenchimento para baixo: resolve relatórios exportados com a categoria escrita só na primeira linha do grupo.
- Agrupar por: cria uma tabela resumida (útil para tabelas de apoio e agregações).
- Colunas de exemplo: você digita o resultado desejado e o Power Query descobre a fórmula.
- Transpor: inverte linhas e colunas — use com cuidado, costuma ser sintoma de origem mal estruturada.

```
M
let
    Fonte = tbOrcamento,
    Unpivot = Table.UnpivotOtherColumns(
        Fonte,
        {"CentroCusto", "ContaContabil"},
        "MesTexto",
        "Valor"
    ),
    MesData = Table.AddColumn(Unpivot, "Competencia",
        each Date.FromText("01-" & [MesTexto], [Format="dd-MMM-yyyy", Culture="pt-BR"]),
        type date)
in
    MesData
```

EXERCÍCIO PRÁTICO

Monte uma planilha com Produto nas linhas e 6 meses nas colunas. Faça o unpivot com 'outras colunas', converta o mês em data e depois acrescente um 7º mês na origem para confirmar que a consulta absorve sozinha.

Colunas condicionais, personalizadas e agrupamento

Classificações, faixas e tabelas resumidas sem uma linha de DAX.

Coluna condicional pela interface

Adicionar Coluna → Coluna Condicional resolve a maior parte das classificações: faixa de ticket, porte de cliente, status de prazo. É visual, legível e mantém o folding em fontes relacionais.

M

```
Table.AddColumn(
    Etapa_Anterior,
    "FaixaTicket",
    each
        if [Receita] < 500 then "1. Até 500"
        else if [Receita] < 2000 then "2. 500 a 2.000"
        else if [Receita] < 10000 then "3. 2.000 a 10.000"
        else "4. Acima de 10.000",
    type text
)
```

BOA PRÁTICA

Escrever '1. Até 500' garante a ordem correta nos gráficos sem precisar de coluna de classificação. Se o prefixo incomodar visualmente, crie uma coluna separada 'OrdemFaixa' e use 'Classificar por coluna' na modelagem.

Coluna personalizada com lógica de negócio

M

```
Table.AddColumn(
    Etapa_Anterior,
    "StatusEntrega",
    each
        let dias = Duration.Days([DataEntrega] - [DataPrometida])
        in
            if [DataEntrega] = null then "Pendente"
            else if dias <= 0 then "No prazo"
            else if dias <= 2 then "Atraso tolerável"
            else "Atrasado",
    type text
)
```

Agrupar por

Transformar → Agrupar por cria resumos. Use para tabelas de apoio (ex.: primeira compra de cada cliente) ou para reduzir o volume de uma fato que não precisa de detalhe.

M

```
Table.Group(
    Etapa_Anterior,
    {"IdCliente"},
    {
        {"PrimeiraCompra", each List.Min([Data]), type date},
        {"UltimaCompra", each List.Max([Data]), type date},
        {"QtdPedidos", each Table.RowCount(_), Int64.Type},
        {"ReceitaTotal", each List.Sum([Receita]), type number}
    }
)
```

ATENÇÃO

Ambas ocupam espaço no modelo. A do Power Query é calculada na carga e comprime melhor; a de DAX pode referenciar outras tabelas por relacionamento. Na dúvida, faça no Power Query.

EXERCÍCIO PRÁTICO

Na sua tabela de vendas, crie: (1) uma faixa de ticket com 4 categorias ordenadas, (2) uma coluna de status de entrega com tolerância de 2 dias e (3) uma tabela agregada por cliente com primeira compra, última compra e receita total.

Quiz do módulo

1. O que o Power Query realmente armazena?

- Os dados transformados
- **✓ Uma receita de passos em linguagem M**
- Apenas o arquivo original
- Um banco de dados temporário

A receita é reexecutada a cada atualização — por isso ela precisa ser resistente a mudanças na origem.

2. Qual a diferença entre Referência e Duplicar uma consulta?

- Não há diferença
- **✓ Referência encadeia a consulta original; Duplicar copia todo o código**
- Duplicar é mais rápido na atualização
- Referência só funciona em fontes SQL

Referência centraliza a manutenção; Duplicar cria duas cópias que precisam ser corrigidas em dobro.

3. Por que separar Data de Hora em colunas diferentes?

- Porque o Power BI não aceita DataHora
- **✓ Para melhorar a compressão e permitir relacionamento limpo com o calendário**
- Para deixar o visual mais bonito
- Porque DAX não lê hora

A alta cardinalidade de DataHora incha o modelo e impede o relacionamento correto com a dimensão Calendário.

4. Qual opção de unpivot sobrevive à chegada de um novo mês na origem?

- Transformar colunas selecionadas em linhas
- **✓ Transformar outras colunas em linhas**
- Transpor
- Dinamizar coluna

'Outras colunas' define o que fica fixo; qualquer coluna nova é automaticamente transformada em linha.

5. Você recebe erro ao converter '1.234,56' para número. O que fazer?

- Remover os erros

- ✓ **Converter usando a localidade pt-BR**
- Transformar a coluna em texto
- Dividir a coluna por vírgula

A localidade define como o separador decimal e de milhar são interpretados.

6. Para classificar clientes em faixas de faturamento, o caminho mais indicado é:

- Medida DAX com SWITCH
- ✓ **Coluna condicional no Power Query**
- Segmentação numérica
- Grupo de visuais

É um atributo fixo por linha, não depende de filtro: pertence ao Power Query.

Power Query avançado e linguagem M

Mesclar e acrescentar, parâmetros, funções personalizadas em M, tratamento de erros e performance de atualização.

Objetivos de aprendizagem

- Dominar os tipos de junção em Mesclar Consultas
- Criar parâmetros e funções personalizadas reaproveitáveis
- Ler e escrever M com confiança: let/in, each, registros e listas
- Diagnosticar e acelerar atualizações lentas

Mesclar x Acrescentar: junções na prática

Trazer colunas de outra tabela ou empilhar tabelas — e quando NÃO fazer nenhum dos dois.

Acrescentar (Append) empilha linhas: 12 arquivos mensais viram uma tabela. Mesclar (Merge) traz colunas de outra tabela com base em uma chave — é o PROCV do Power Query, mas muito mais poderoso.

Os tipos de junção

Junção	Resultado	Uso típico
Externa esquerda	Todas da primeira + correspondentes da segunda	Padrão: enriquecer a fato com dados da dimensão
Externa direita	Todas da segunda + correspondentes da primeira	Raro; geralmente é a esquerda invertida
Externa completa	Tudo dos dois lados	Conciliação entre dois sistemas
Interna	Só as que existem nos dois	Filtrar a fato pelas chaves válidas
Anti esquerda	Só as da primeira que NÃO existem na segunda	Auditoria: produtos vendidos sem cadastro
Anti direita	Só as da segunda sem correspondência	Auditoria: cadastro sem movimento

BOA PRÁTICA

Antes de confiar em qualquer modelo, faça uma anti-junção esquerda entre a fato e cada dimensão. Se voltar linha, existe chave órfã — e o total do seu painel vai divergir do sistema.

```

M
let
    Merge = Table.NestedJoin(
        fVendas, {"IdProduto"},
        dProduto, {"IdProduto"},
        "dProduto", JoinKind.LeftOuter
    ),
    Expandir = Table.ExpandTableColumn(
        Merge, "dProduto",
        {"Categoria", "Subcategoria"},
        {"Categoria", "Subcategoria"}
    )
in
    Expandir

```

ATENÇÃO

Se Produto vira uma dimensão no modelo, NÃO traga Categoria para dentro da fato por mesclagem. Isso duplica dados e engorda a tabela grande. Mesкле apenas quando o campo não puder virar dimensão.

Acrescentar com estruturas diferentes

Ao acrescentar tabelas com nomes de coluna diferentes, o Power Query cria colunas separadas cheias de nulo. Padronize os nomes ANTES do append — idealmente dentro da função de transformação usada no conector Pasta.

EXERCÍCIO PRÁTICO

Faça uma anti-junção esquerda entre sua tabela de vendas e a de produtos. Documente as chaves órfãs encontradas e decida: corrigir na origem, criar um registro 'Não cadastrado' na dimensão, ou excluir as linhas.

Parâmetros e funções personalizadas

Como fazer o projeto sobreviver à troca de servidor, de pasta e de ambiente.

Parâmetros

Um parâmetro é um valor nomeado que você usa no lugar de uma constante espalhada pelo código. Gerenciar Parâmetros → Novo Parâmetro. Use para caminho de pasta, nome de servidor, ambiente (Dev/Prod), chave de API e janela de datas.

```

M
let
    Fonte = Sql.Database(pServidor, pBancoDados),
    Vendas = Fonte[["Schema="&db, Item="fato_vendas"]][Data]
in
    Vendas

```

BOA PRÁTICA

Depois de publicar, você altera o valor do parâmetro direto nas configurações do conjunto de dados — sem republicar o .pbix. É assim que se promove um relatório de homologação para produção.

Funções personalizadas

Uma função é uma consulta que recebe argumentos. Você escreve a lógica uma vez e aplica em qualquer lugar. A forma mais fácil de criar: faça a transformação em uma consulta comum, depois clique com o botão direito → Criar Função.

```
M
// Consulta: fnLimparTexto
(texto as nullable text) as nullable text =>
let
    SemEspacos = Text.Trim(texto ?? ""),
    SemDuplos = Text.Combine(
        List.Select(Text.Split(SemEspacos, " "), each _ <> ""),
        " "
    ),
    Maiuscula = Text.Proper(SemDuplos),
    Resultado = if Maiuscula = "" then null else Maiuscula
in
    Resultado
```

```
M
Table.TransformColumns(
    Etapa_Anterior,
    {
        {"NomeCliente", fnLimparTexto, type nullable text},
        {"Cidade", fnLimparTexto, type nullable text},
        {"Vendedor", fnLimparTexto, type nullable text}
    }
)
```

```
M
// Consulta: fnDiasUteis
(dataInicio as date, dataFim as date, feriados as list) as number =>
let
    Dias = List.Dates(dataInicio, Duration.Days(dataFim - dataInicio) + 1, #duration(1,0,0,0)),
    Uteis = List.Select(Dias, each
        Date.DayOfWeek(_, Day.Monday) < 5 and not List.Contains(feriados, _)
    )
in
    List.Count(Uteis)
```

EXERCÍCIO PRÁTICO

Crie um parâmetro pCaminhoDados e refatore todas as suas consultas para usá-lo. Depois crie a função fnDiasUteis e aplique-a na sua tabela de pedidos para calcular o prazo real de entrega em dias úteis.

Linguagem M essencial

let/in, each, registros, listas e tabelas — o vocabulário que destrava o Power Query.

A estrutura let ... in

Toda consulta M é uma expressão let: uma sequência de variáveis nomeadas e, no final, o in que diz qual delas é o resultado. Não existe 'ordem de execução' obrigatória — o M resolve as dependências.

Os três tipos estruturados

Tipo	Sintaxe	Acesso
Lista	{1, 2, 3} ou {"a".. "z"}	Lista{0} → primeiro item
Registro	[Nome = "Ana", Idade = 30]	Registro[Nome]
Tabela	#table({"A","B"}, {{1,2},{3,4}})	Tabela{0}[A] → célula

```
M
let
    Lista = {10, 20, 30, 40},
    Soma = List.Sum(Lista),                // 100
    Media = List.Average(Lista),          // 25
    Registro = [ Produto = "Teclado", Preço = 199.90 ],
    Preço = Registro[Preço],              // 199.9
    Tabela = #table(
        {"Produto", "Preço"},
        {"Teclado", 199.90}, {"Mouse", 89.90}
    ),
    PrimeiroProduto = Tabela{0}[Produto] // "Teclado"
in
    [ Soma = Soma, Media = Media, Preço = Preço, Primeiro = PrimeiroProduto ]
```

each e o sublinhado

each é açúcar sintático para (_) =>. Dentro de uma transformação de tabela, _ representa a linha atual (um registro). Por isso [Coluna] funciona: é atalho para _[Coluna].

```
M
Table.AddColumn(t, "Total", each [Qtd] * [Preço])
Table.AddColumn(t, "Total", each _[Qtd] * _[Preço])
Table.AddColumn(t, "Total", (linha) => linha[Qtd] * linha[Preço])
```

ATENÇÃO

Table.SelectRows funciona; table.selectrows não. É o erro mais comum de quem vem do Excel.

Tratamento de erro com try/otherwise

M

```
Table.AddColumn(  
    Etapa_Anterior,  
    "Diagnostico",  
    each  
        let tentativa = try Number.FromText([ValorTexto], "pt-BR")  
        in  
            if tentativa[HasError]  
            then "ERRO: " & tentativa[Error][Message]  
            else "OK",  
    type text  
)
```

Funções que você mais vai usar

- Text: Trim, Upper, Proper, Contains, Start, End, BetweenDelimiters, PadStart, Replace.
- Date: From, Year, Month, StartOfMonth, EndOfMonth, AddDays, AddMonths, DayOfWeek, IsInCurrentYear.
- Number: From, Round, RoundDown, Abs, IntegerDivide, Mod.
- List: Sum, Average, Max, Min, Count, Distinct, Select, Transform, Contains, Generate, Dates.
- Table: SelectRows, SelectColumns, AddColumn, Group, Sort, Distinct, RemoveColumns, Buffer.

EXERCÍCIO PRÁTICO

Escreva, na Consulta em Branco, uma expressão M que gere uma tabela com os 12 meses de 2025, o nome do mês em português e um sinalizador de trimestre — usando apenas List.Dates, Table.FromList e Table.AddColumn.

Performance de atualização

Por que sua atualização demora 40 minutos e como reduzir para 3.

Os cinco maiores vilões

- Query folding quebrado cedo: o Power BI baixa tudo e filtra na sua máquina.
- Mesclagens pesadas: junção entre duas tabelas grandes sem chave indexada.
- Colunas desnecessárias: cada coluna carregada custa memória e tempo.
- Fórmulas linha a linha em milhões de linhas, especialmente chamadas externas.
- Reavaliação: uma mesma consulta referenciada por 4 outras pode ser executada 4 vezes.

Diagnóstico

1. Ferramentas → Diagnóstico de Consulta → Iniciar diagnóstico.
2. Atualize a consulta suspeita e pare o diagnóstico.
3. Analise a coluna de duração por passo: um passo que sozinho leva 80% do tempo é o alvo.
4. Verifique 'Exibir consulta nativa' no último passo possível para saber onde o folding parou.

BOA PRÁTICA

Table.Buffer materializa a tabela na memória e impede reavaliações repetidas — mas também mata o query folding daquele ponto em diante. Use só em tabelas pequenas referenciadas muitas vezes.

Checklist antes de publicar

Verificação	Como fazer	Ganho típico
Remover colunas não usadas	Escolher Colunas, logo no início	20-60% de memória
Filtrar o histórico necessário	Filtro de data na origem	Enorme
Desabilitar carga de auxiliares	Botão direito → Habilitar carga	Menos tabelas no modelo
Desativar 'Data/hora automática'	Opções → Carregamento de dados	Menos tabelas ocultas
Reduzir cardinalidade	Separar data/hora, arredondar decimais	Alto
Ativar atualização incremental	Parâmetros RangeStart/RangeEnd	Refresh de minutos → segundos

ATENÇÃO

Quando você combina duas fontes com níveis de privacidade diferentes, o Power BI impede o folding para proteger os dados e baixa tudo para a memória. Se as fontes são confiáveis, ajuste o nível de privacidade — a atualização pode cair de 30 minutos para 2.

EXERCÍCIO PRÁTICO

Rode o Diagnóstico de Consulta na sua consulta mais lenta, identifique o passo mais caro e aplique duas melhorias (remoção de colunas e reordenação de passos). Registre o tempo antes e depois.

Quiz do módulo

1. Você quer encontrar produtos vendidos que não existem no cadastro. Qual junção usar?

- Interna
- Externa completa
- ✓ **Anti esquerda**
- Externa direita

Anti esquerda devolve as linhas da primeira tabela sem correspondência na segunda.

2. Qual a vantagem de usar parâmetro para o nome do servidor?

- Deixa a consulta mais rápida
- ✓ **Permite trocar o valor no Service sem republicar o arquivo**
- É obrigatório em DirectQuery
- Melhora o query folding

Parâmetros são editáveis nas configurações do conjunto de dados publicado.

3. Em M, o que significa each?

- Um laço for
- ✓ **Açúcar sintático para (_) => , onde _ é o item ou linha atual**
- Uma função de agregação
- Um comentário

Por isso [Coluna] dentro de each equivale a _[Coluna].

4. Table.Buffer deve ser usado quando?

- Sempre, no início de toda consulta
- ✓ **Em tabelas pequenas referenciadas várias vezes, aceitando perder o folding dali em diante**
- Apenas em DirectQuery
- Nunca

Ele evita reavaliação, mas interrompe o dobramento — é uma troca consciente.

5. Qual prática dá o maior ganho de memória com o menor esforço?

- Renomear os passos
- ✓ **Remover colunas não utilizadas no início da consulta**
- Usar Transpor
- Duplicar consultas

Coluna que não entra no modelo não custa memória nem tempo de atualização.

6. Por que níveis de privacidade podem deixar a atualização lenta?

- Porque criptografam os dados
- ✓ **Porque impedem o folding entre fontes diferentes, forçando o download completo**
- Porque exigem login a cada atualização
- Porque limitam o número de linhas

O motor evita enviar dados de uma fonte para outra e passa a processar localmente.

Modelagem dimensional e esquema estrela

Fato e dimensão, cardinalidade, direção de filtro, relacionamentos inativos e por que o esquema estrela é inegociável.

Objetivos de aprendizagem

- Separar tabelas fato de tabelas dimensão com critério
- Construir relacionamentos corretos e entender a direção do filtro
- Resolver relações muitos-para-muitos e papéis múltiplos de data
- Reconhecer e corrigir os quatro erros de modelagem mais comuns

Fato e dimensão: a decisão que define o projeto

Como olhar para uma tabela e saber imediatamente onde ela entra no modelo.

Modelagem dimensional é simples de enunciar: eventos vão para tabelas fato, contextos vão para tabelas dimensão. A dificuldade está em aplicar isso quando a área te entrega uma planilha com 90 colunas misturando tudo.

Característica	Tabela Fato	Tabela Dimensão
Conteúdo	Eventos que aconteceram	Descrição de entidades
Exemplos	Vendas, apontamentos, lançamentos, atendimentos	Produto, Cliente, Calendário, Funcionário, Loja
Volume	Milhões de linhas	Dezenas a milhares
Colunas típicas	Chaves + valores numéricos + datas	Chave + atributos descritivos
Uso no visual	Vira medida (soma, contagem)	Vira eixo, legenda, filtro
Crescimento	Cresce todo dia	Cresce devagar

BOA PRÁTICA

Pergunte: 'isso é algo que aconteceu em uma data?' Se sim, é fato. 'Isso é algo que EXISTE e descreve outra coisa?' Se sim, é dimensão.

Granularidade: a pergunta mais importante

Granularidade é o que representa UMA linha da tabela fato. Uma linha por item do pedido? Por pedido? Por dia e produto? Você precisa saber responder em uma frase. Se não sabe, o modelo vai duplicar valores.

- fVendas: uma linha por item de pedido (nota + produto).
- fEstoque: uma linha por produto, depósito e dia (foto diária).
- fMetas: uma linha por vendedor, produto e mês.

- Nunca misture granularidades na mesma tabela fato. Se precisa de duas, faça duas fatos ligadas às mesmas dimensões.

Os tipos de tabela fato

Tipo	O que guarda	Exemplo
Transacional	Um evento por linha	Cada venda, cada apontamento de produção
Snapshot periódico	Uma foto do estado em cada período	Saldo de estoque por dia, headcount por mês
Snapshot acumulado	Uma linha por processo, com várias datas	Pedido: data de criação, aprovação, faturamento, entrega

ATENÇÃO

Somar o saldo de estoque de todos os dias do mês não faz sentido. Snapshots pedem medidas semiaditivas (último saldo, saldo médio) — você aprende isso no Módulo 9.

EXERCÍCIO PRÁTICO

Liste todas as tabelas que você tem em um projeto e classifique cada uma em fato ou dimensão. Para cada fato, escreva em uma frase qual é a granularidade de uma linha.

Esquema estrela x floco de neve x tabela única

Por que a estrela ganha em 95% dos casos, com números.

No esquema estrela, cada dimensão se liga diretamente à fato — nada de dimensão ligada a dimensão. O desenho fica com a fato no centro e as dimensões ao redor, como uma estrela.

Modelo	Como é	Consequência
Tabela única (flat)	Tudo em uma tabela gigante	Repete texto milhões de vezes, arquivo enorme, filtro lento, manutenção impossível
Floco de neve	Produto → Subcategoria → Categoria	Mais junções por consulta, DAX mais chato, ganho de espaço irrelevante
Estrela	Fato no centro, dimensões achatadas ao redor	Consulta mais rápida, DAX simples, usuário entende o painel de campos

BOA PRÁTICA

Traga Subcategoria e Categoria para dentro da dimensão Produto com uma mesclagem no Power Query. O motor VertiPaq comprime texto repetido muito bem — o espaço extra é desprezível e a performance melhora.

Por que a estrela é mais rápida

O VertiPaq comprime coluna a coluna. Em uma tabela única, a coluna 'Nome do Cliente' com 3 milhões de linhas tem que ser varrida em cada filtro. Na estrela, o filtro acontece na dimensão (50 mil linhas), é resolvido para um conjunto de chaves numéricas e só então aplicado à fato — que guarda apenas inteiros.

Regras práticas do esquema estrela

- Nomeie com prefixo: fVendas, dProduto, dCliente, dCalendario. Fica óbvio no painel de campos.
- Esconda todas as colunas de chave (IdProduto na fato e na dimensão). O usuário nunca deve arrastá-las.
- Esconda todas as colunas numéricas cruas da fato e exponha apenas medidas.
- Nunca crie relacionamento entre duas tabelas fato. Ligue as duas às mesmas dimensões.
- Crie uma tabela de medidas vazia para organizar o DAX (Inserir dados → tabela vazia → esconder a coluna).

ATENÇÃO

Se você precisa de um DAX complicadíssimo para uma pergunta simples, o problema quase nunca é o DAX. É o modelo. Volte ao esquema estrela antes de escrever CALCULATE aninhado.

EXERCÍCIO PRÁTICO

Pegue uma tabela única (flat) com 15+ colunas e decomponha em uma fato e pelo menos 3 dimensões no Power Query. Compare o tamanho do arquivo antes e depois.

Relacionamentos: cardinalidade e direção do filtro

1:N, N:N, filtro único x bidirecional e relacionamentos inativos.

Cardinalidade

Tipo	Quando ocorre	Recomendação
Um para muitos (1:N)	Dimensão → Fato	O padrão. 95% dos seus relacionamentos.
Um para um (1:1)	Duas tabelas com a mesma chave	Quase sempre deveriam ser uma tabela só.
Muitos para muitos (N:N)	Nenhum lado tem chave única	Evite. Crie uma dimensão-ponte com valores únicos.

Direção do filtro

Por padrão o filtro flui do lado 'um' para o lado 'muitos': ao selecionar uma categoria em dProduto, a fVendas é filtrada. O caminho inverso não acontece — e isso é uma proteção, não uma limitação.

ATENÇÃO

Ativar filtro cruzado 'Ambos' parece resolver um problema pontual e cria três outros: ambiguidade de caminho, resultados inesperados em medidas e queda de performance. Resolva com CROSSFILTER dentro da medida específica que precisa — não no relacionamento.

DAX

```
Clientes com Compra =  
CALCULATE(  
    DISTINCTCOUNT( dCliente[IdCliente] ),  
    CROSSFILTER( fVendas[IdCliente], dCliente[IdCliente], BOTH )  
)
```

Relacionamentos inativos e USERELATIONSHIP

Uma fato de pedidos tem várias datas: pedido, faturamento, entrega. Só uma pode estar ativa contra o calendário. As outras ficam pontilhadas (inativas) e você as ativa dentro de medidas específicas.

DAX

```
Pedidos Realizados =          -- usa o relacionamento ativo (DataPedido)  
COUNTROWS( fPedidos )  
  
Pedidos Faturados =  
CALCULATE(  
    COUNTROWS( fPedidos ),  
    USERELATIONSHIP( fPedidos[DataFaturamento], dCalendario[Data] )  
)  
  
Pedidos Entregues =  
CALCULATE(  
    COUNTROWS( fPedidos ),  
    USERELATIONSHIP( fPedidos[DataEntrega], dCalendario[Data] )  
)
```

BOA PRÁTICA

Quando o usuário precisa alternar livremente entre 'analisar por data de pedido / faturamento / entrega', crie uma tabela desconectada com essas três opções e use SWITCH + SELECTEDVALUE para escolher a medida. Fica muito mais intuitivo que três medidas separadas.

Chaves compostas

O Power BI não aceita relacionamento por duas colunas. Quando a chave real é Loja + Produto, crie uma coluna concatenada nos dois lados — de preferência no Power Query.

M

```
Table.AddColumn(  
    Etapa_Anterior,  
    "ChaveLojaProduto",  
    each Text.From([IdLoja]) & "|" & Text.From([IdProduto]),  
    type text  
)
```

EXERCÍCIO PRÁTICO

Monte uma fato de pedidos com três datas diferentes e uma única dCalendario. Crie as três medidas usando USERRELATIONSHIP e valide que os totais batem com a contagem em cada data.

Os erros de modelagem que mais custam caro

Diagnóstico e correção dos problemas que aparecem em quase todo projeto herdado.

Erro 1 — Duas fatos relacionadas entre si

Ligar fVendas diretamente a fMetas parece prático e produz duplicação: cada linha de meta se multiplica pelas vendas. Solução: ligue as duas às mesmas dimensões (dCalendario, dVendedor, dProduto).

Erro 2 — Calendário vindo da coluna de data da fato

Se o calendário só tem os dias em que houve venda, o gráfico esconde os dias sem movimento e a inteligência temporal calcula errado. O calendário precisa ser contínuo, sem buracos.

Erro 3 — Colunas calculadas onde deveriam ser medidas

DAX

```
-- ERRADO: coluna calculada na fato, ocupa memória e não responde ao filtro
Margem % (coluna) = DIVIDE( fVendas[Receita] - fVendas[Custo], fVendas[Receita] )
```

```
-- CERTO: medida, calculada no contexto do visual
Margem % =
VAR Receita = [Receita Total]
VAR Custo   = [Custo Total]
RETURN DIVIDE( Receita - Custo, Receita )
```

ATENÇÃO

Somar a coluna 'Margem %' de cada linha e dividir pela contagem dá um número que não significa nada. Percentual sempre se calcula sobre os totais, e por isso precisa ser medida.

Erro 4 — Chaves órfãs e a linha em branco

Quando a fato tem uma chave que não existe na dimensão, o Power BI cria silenciosamente uma linha em branco na dimensão. Seus totais batem, mas o detalhamento por categoria 'perde' valores em uma linha vazia.

1. Detecte com anti-junção no Power Query (Módulo 4) ou com uma medida de auditoria.
2. Corrija na origem quando possível.
3. Se não for possível, adicione na dimensão um registro 'Não cadastrado' com a chave -1 e substitua os nulos da fato por -1.
4. Nunca ignore: valor em linha em branco é o que faz o gerente dizer 'esse painel não bate'.

DAX

```
Linhas sem Produto =  
CALCULATE(  
    COUNTROWS( fVendas ),  
    ISBLANK( RELATED( dProduto[IdProduto] ) )  
)
```

Checklist de modelo saudável

- Todas as dimensões ligadas direto à fato, sem floco.
- Nenhum relacionamento bidirecional sem justificativa escrita.
- Uma dCalendario contínua, marcada como tabela de datas.
- Chaves escondidas, medidas organizadas em uma tabela própria.
- Zero chaves órfãs.
- Nenhuma coluna calculada que poderia ser medida.

EXERCÍCIO PRÁTICO

Faça uma auditoria completa em um modelo existente usando o checklist acima. Documente cada problema encontrado, o impacto no número exibido e a correção aplicada.

Quiz do módulo

1. Qual tabela é uma fato?

- Cadastro de produtos
- ✓ **Lançamentos contábeis do mês**
- Lista de centros de custo
- Tabela de feriados

Lançamentos são eventos com data e valor — fato clássica.

2. Granularidade é:

- A quantidade de colunas da tabela
- ✓ **O que UMA linha da tabela fato representa**
- O nível de compressão do modelo
- A quantidade de relacionamentos

Definir a granularidade evita duplicação e ambiguidade no modelo.

3. Por que evitar filtro cruzado bidirecional?

- Porque consome licença Pro
- ✓ **Porque gera ambiguidade, resultados inesperados e perda de performance**
- Porque não funciona no Service
- Porque impede o uso de segmentações

Quando precisar, use CROSSFILTER dentro da medida específica.

4. A fato tem data de pedido, de faturamento e de entrega. Como analisar pelas três?

- Criar três tabelas calendário

- ✓ **Criar relacionamentos inativos e usar USERELATIONSHIP nas medidas**
- Usar filtro bidirecional
- Duplicar a tabela fato três vezes

Um calendário, relacionamentos inativos e USERELATIONSHIP é o padrão profissional.

5. Por que 'Margem %' deve ser medida e não coluna calculada?

- Porque coluna calculada não aceita divisão
- ✓ **Porque o percentual precisa ser calculado sobre os totais do contexto, não linha a linha**
- Porque medidas são mais bonitas
- Porque colunas não podem ter %

Somar percentuais de linhas gera um número sem significado.

6. O painel mostra valores em uma categoria em branco. A causa mais provável é:

- Erro de visual
- ✓ **Chave órfã na fato sem correspondência na dimensão**
- Falta de licença
- Relacionamento 1:1

O Power BI cria uma linha em branco na dimensão para acomodar as chaves não encontradas.

Calendário e inteligência temporal

Construir uma dCalendario completa (inclusive com dias úteis e feriados) e dominar YTD, MTD, YoY, MoM, acumulados e médias móveis.

Objetivos de aprendizagem

- Criar uma tabela calendário completa em DAX ou em M
- Marcar corretamente a tabela de datas e entender por que isso importa
- Escrever as medidas comparativas que todo painel gerencial precisa
- Tratar calendário fiscal, semanas e períodos incompletos

Criando a dCalendario definitiva

Uma tabela, todos os atributos, pronta para copiar e colar em qualquer projeto.

A dCalendario é a dimensão mais reaproveitável do Power BI. Construa uma vez com capricho e leve para todos os projetos. Ela precisa ser contínua (sem dias faltando) e cobrir todo o período da fato.

DAX

```
dCalendario =
VAR DataMin = MIN( fVendas[Data] )
VAR DataMax = MAX( fVendas[Data] )
VAR Base =
    CALENDAR( DATE( YEAR(DataMin), 1, 1 ), DATE( YEAR(DataMax), 12, 31 ) )
RETURN
ADDCOLUMNS(
    Base,
    "Ano", YEAR( [Date] ),
    "NumMes", MONTH( [Date] ),
    "Mes", FORMAT( [Date], "mmm", "pt-BR" ),
    "MesCompleto", FORMAT( [Date], "mmmm", "pt-BR" ),
    "AnoMes", FORMAT( [Date], "yyyy-mm" ),
    "AnoMesTexto", FORMAT( [Date], "mmm/yy", "pt-BR" ),
    "Trimestre", "T" & FORMAT( QUARTER([Date]), "0" ),
    "AnoTrimestre", YEAR([Date]) & "-T" & FORMAT( QUARTER([Date]), "0" ),
    "Semestre", "S" & IF( MONTH([Date]) <= 6, "1", "2" ),
    "NumDiaSemana", WEEKDAY( [Date], 2 ),
    "DiaSemana", FORMAT( [Date], "ddd", "pt-BR" ),
    "EhFimDeSemana", WEEKDAY( [Date], 2 ) > 5,
    "InicioMes", EOMONTH( [Date], -1 ) + 1,
    "FimMes", EOMONTH( [Date], 0 ),
    "AnoAtual", YEAR([Date]) = YEAR( TODAY() ),
    "MesAtual", EOMONTH([Date],0) = EOMONTH( TODAY(), 0 ),
    "OffsetMes", DATEDIFF( TODAY(), [Date], MONTH ),
    "OffsetAno", YEAR([Date]) - YEAR( TODAY() ),
    "PassadoOuHoje", [Date] <= TODAY()
)
```

BOA PRÁTICA

OffsetMes = 0 é o mês atual, -1 o anterior, e assim por diante. Com ela você cria filtros do tipo 'últimos 12 meses' que se movem sozinhos, sem precisar reescrever nada todo mês.

Ordenação de mês e dia da semana

Sem ajuste, o gráfico ordena os meses em ordem alfabética (abr, ago, dez...). Selecione a coluna Mes → Ferramentas de Coluna → Classificar por coluna → NumMes. Faça o mesmo para DiaSemana usando NumDiaSemana.

Marcar como tabela de datas

1. Selecione a dCalendario no painel de dados.
2. Ferramentas de Tabela → Marcar como tabela de datas.
3. Escolha a coluna Date.
4. O Power BI valida: valores únicos, sem nulos e sem intervalos faltando.

ATENÇÃO

Funções como DATESYTD e SAMEPERIODLASTYEAR assumem uma tabela de datas marcada. Sem isso, o filtro de contexto pode não ser removido corretamente e os resultados ficam sutilmente errados — o pior tipo de erro.

Feriados e dias úteis

DAX

```
EhDiaUtil =  
VAR Dia = dCalendario[Date]  
VAR FimDeSemana = WEEKDAY( Dia, 2 ) > 5  
VAR EhFeriado = Dia IN VALUES( dFeriados[Data] )  
RETURN NOT ( FimDeSemana || EhFeriado )
```

EXERCÍCIO PRÁTICO

Crie a dCalendario completa no seu modelo, marque-a como tabela de datas, ajuste a ordenação de mês e dia da semana e traga os feriados da API que você conectou no Módulo 2 para criar a coluna EhDiaUtil.

YTD, MTD, QTD e acumulados

As medidas de acumulado que todo painel gerencial exige.

DAX

```
Receita = SUM( fVendas[Receita] )  
  
Receita YTD = -- acumulado do ano  
CALCULATE( [Receita], DATESYTD( dCalendario[Date] ) )  
  
Receita QTD = -- acumulado do trimestre  
CALCULATE( [Receita], DATESQTD( dCalendario[Date] ) )  
  
Receita MTD = -- acumulado do mês  
CALCULATE( [Receita], DATESMTD( dCalendario[Date] ) )  
  
Receita YTD Fiscal = -- exercício iniciando em julho  
CALCULATE( [Receita], DATESYTD( dCalendario[Date], "06-30" ) )
```

NOTA

TOTALYTD([Receita], dCalendario[Date]) equivale a CALCULATE([Receita], DATESYTD(...)). Prefira a forma com CALCULATE: fica mais claro para quem lê e permite adicionar outros filtros.

Acumulado desde sempre (running total)

DAX

```
Receita Acumulada =  
VAR UltimaData = MAX( dCalendario[Date] )  
RETURN  
CALCULATE(  
    [Receita],  
    dCalendario[Date] <= UltimaData,  
    REMOVEFILTERS( dCalendario )  
)
```

ATENÇÃO

Se você remover o filtro apenas de dCalendario[Date], as colunas Ano e Mês continuam filtrando e o acumulado reinicia a cada ano sem você perceber.

Média móvel

DAX

```
Receita Média 3M =  
VAR UltimoMes = MAX( dCalendario[Date] )  
VAR Janela =  
    DATESINPERIOD( dCalendario[Date], UltimoMes, -3, MONTH )  
RETURN  
CALCULATE(  
    DIVIDE(  
        [Receita],  
        DISTINCTCOUNT( dCalendario[AnoMes] )  
    ),  
    Janela  
)
```

DAX

```
Pedidos Média 7D =  
VAR UltimoDia = MAX( dCalendario[Date] )  
RETURN  
CALCULATE(  
    AVERAGEX(  
        DATESINPERIOD( dCalendario[Date], UltimoDia, -7, DAY ),  
        [Qtd Pedidos]  
    )  
)
```

EXERCÍCIO PRÁTICO

Crie as medidas Receita, Receita YTD, Receita Acumulada e Receita Média 3M. Monte um gráfico de linhas com Receita e Média 3M por mês e verifique visualmente o efeito de suavização.

Comparativos: YoY, MoM e período anterior

Como comparar com o passado sem cair nas armadilhas de período incompleto.

DAX

```
Receita LY = -- mesmo período do ano anterior  
CALCULATE( [Receita], SAMEPERIODLASTYEAR( dCalendario[Date] ) )
```

```
Var. YoY =  
[Receita] - [Receita LY]
```

```
Var. YoY % =  
DIVIDE( [Receita] - [Receita LY], [Receita LY] )
```

DAX

```
Receita MesAnterior =  
CALCULATE( [Receita], DATEADD( dCalendario[Date], -1, MONTH ) )
```

```
Var. MoM % =  
DIVIDE( [Receita] - [Receita MesAnterior], [Receita MesAnterior] )
```

```
Receita PY YTD = -- acumulado do ano passado até a mesma data  
CALCULATE( [Receita], DATESYTD( SAMEPERIODLASTYEAR( dCalendario[Date] ) ) )
```

ATENÇÃO

No dia 10 de março, comparar março/2026 (10 dias) contra março/2025 (31 dias) produz uma queda de 70% que não existe. Sempre compare períodos equivalentes.

DAX

```
Receita LY Equivalente =  
VAR UltimoDiaComDado = MAX( fVendas[Data] )  
VAR DiaDoMes = DAY( UltimoDiaComDado )  
VAR PerodoAnterior =  
    FILTER(  
        ALL( dCalendario ),  
        YEAR( dCalendario[Date] ) = YEAR( UltimoDiaComDado ) - 1  
        && MONTH( dCalendario[Date] ) = MONTH( UltimoDiaComDado )  
        && DAY( dCalendario[Date] ) <= DiaDoMes  
    )  
RETURN  
CALCULATE( [Receita], PerodoAnterior )
```

Esconder o futuro

Um gráfico de linha que despenca para zero nos meses futuros assusta o diretor sem motivo. Faça a medida devolver BLANK quando o período ainda não aconteceu.

DAX

```
Receita (sem futuro) =  
VAR InicioPeriodo = MIN( dCalendario[Date] )  
RETURN  
IF( InicioPeriodo <= TODAY(), [Receita] )
```

BOA PRÁTICA

DATEADD desloca o intervalo exato selecionado. SAMEPERIODLASTYEAR é um DATEADD de -1 ano. PARALLELPERIOD sempre devolve o período inteiro (mês/trimestre/ano completo), mesmo que você tenha selecionado 5 dias — use com consciência.

EXERCÍCIO PRÁTICO

Monte uma matriz com Ano-Mês nas linhas e as colunas Receita, Receita LY, Var. YoY % e Receita LY Equivalente. Compare o comportamento no mês corrente e explique por escrito a diferença entre as duas comparações.

Calendário fiscal, semanas e períodos customizados

Quando o ano da empresa não começa em janeiro e a semana não começa no domingo.

Ano fiscal

DAX

```
AnoFiscal =  
IF( MONTH( dCalendario[Date] ) >= 7,  
    YEAR( dCalendario[Date] ) + 1,  
    YEAR( dCalendario[Date] ) )  
  
MesFiscal =  
MOD( MONTH( dCalendario[Date] ) - 7, 12 ) + 1  
  
TrimestreFiscal =  
"TF" & FORMAT( ROUNDUP( ( MOD( MONTH(dCalendario[Date]) - 7, 12 ) + 1 ) / 3, 0 ), "0" )
```

BOA PRÁTICA

DATESYTD(dCalendario[Date], "06-30") já entende o ano fiscal encerrando em 30 de junho. Você não precisa reescrever a lógica de acumulado.

Semanas

Semana é o período mais traiçoeiro: a semana 1 de janeiro pode pertencer ao ano anterior. Trabalhe sempre com a data de início da semana, não com o número dela.

DAX

```
InicioSemana =  
dCalendario[Date] - WEEKDAY( dCalendario[Date], 2 ) + 1  
  
AnoSemana =  
YEAR( dCalendario[InicioSemana] ) & "-S"  
    & FORMAT( WEEKNUM( dCalendario[InicioSemana], 2 ), "00" )
```

DAX

```
Receita Semana Anterior =  
VAR SemanaAtual = MAX( dCalendario[InicioSemana] )  
RETURN  
CALCULATE(  
    [Receita],  
    REMOVEFILTERS( dCalendario ),  
    dCalendario[InicioSemana] = SemanaAtual - 7  
)
```

Períodos dinâmicos escolhidos pelo usuário

DAX

```
-- dPeriodo: tabela com Rotulo e Dias ( "Últimos 7 dias" = 7, "Últimos 30 dias" = 30, ... )  
Receita Período Selecionado =  
VAR DiasEscolhidos = SELECTEDVALUE( dPeriodo[Dias], 30 )  
VAR DataFinal = MAX( dCalendario[Date] )  
RETURN  
CALCULATE(  
    [Receita],  
    DATESINPERIOD( dCalendario[Date], DataFinal, -DiasEscolhidos, DAY )  
)
```

ATENÇÃO

Ela existe só para o usuário escolher uma opção. O efeito é criado dentro da medida com SELECTEDVALUE. Nunca a relacione com a dCalendario.

EXERCÍCIO PRÁTICO

Implemente um ano fiscal iniciando em abril e valide se Receita YTD Fiscal reinicia no mês correto. Depois crie a tabela dPeriodo com quatro opções e uma segmentação que troque a janela de análise.

Quiz do módulo

1. Por que a dCalendario precisa ser contínua?

- Para o arquivo ficar menor
- ✓ **Porque a inteligência temporal depende de uma sequência de datas sem buracos**
- Porque o Power BI exige 365 linhas
- Para permitir DirectQuery

Dias faltando fazem funções como DATESYTD e DATEADD calcularem intervalos errados.

2. O gráfico mostra os meses em ordem alfabética. A correção é:

- Usar FORMAT diferente
- ✓ **Classificar a coluna Mes pela coluna NumMes**
- Criar uma medida de ordenação
- Mudar o idioma do Windows

'Classificar por coluna' é o recurso próprio para isso.

3. Qual medida compara com o mesmo período do ano anterior?

- PARALLELPERIOD sempre
- ✓ **SAMEPERIODLASTYEAR**
- DATESYTD
- PREVIOUSMONTH

SAMEPERIODLASTYEAR desloca exatamente o intervalo selecionado em um ano.

4. No dia 10 do mês, comparar o mês atual com o mês inteiro do ano passado:

- É a prática correta
- ✓ **Distorce a comparação; é preciso limitar o ano anterior aos mesmos dias decorridos**
- Só é problema em anos bissextos
- Depende do visual usado

Comparação justa exige o mesmo número de dias decorridos.

5. Para um acumulado que nunca reinicia, o que usar dentro de CALCULATE?

- DATESYTD
- ✓ **Filtro de data <= data máxima com REMOVEFILTERS na tabela calendário inteira**
- PREVIOUSYEAR
- ALLSELECTED na fato

Remover a tabela inteira evita que Ano e Mês continuem filtrando e reiniciem o acumulado.

6. Uma tabela desconectada de períodos deve ser:

- Relacionada à dCalendario com filtro bidirecional
- ✓ **Mantida sem relacionamento, lida com SELECTEDVALUE dentro da medida**
- Relacionada à fato
- Uma coluna calculada

O papel dela é oferecer uma escolha ao usuário, aplicada por código na medida.

DAX essencial: contexto é tudo

Contexto de linha e contexto de filtro, coluna calculada x medida, as funções de agregação e o CALCULATE — a função mais importante da linguagem.

Objetivos de aprendizagem

- Entender de verdade o que é contexto de filtro e contexto de linha
- Saber quando usar coluna calculada e quando usar medida
- Dominar SUM, COUNT, DISTINCTCOUNT, DIVIDE e o uso de variáveis
- Escrever CALCULATE com confiança, entendendo o que ele faz com o contexto

Contexto de filtro e contexto de linha

O conceito que separa quem 'usa DAX' de quem entende DAX.

Uma medida DAX nunca tem um resultado único. Ela tem um resultado por célula, porque cada célula do visual aplica um conjunto diferente de filtros. Esse conjunto é o contexto de filtro.

Se [Receita] = SUM(fVendas[Receita]) mostra R\$ 12 mi no total e R\$ 800 mil na linha 'Sudeste / Março', é porque a segunda célula somou a mesma coluna com dois filtros aplicados: Região = Sudeste e Mês = Março.

De onde vem o contexto de filtro	Exemplo
Linhas e colunas do visual	Categoria no eixo de um gráfico de barras
Segmentações (slicers)	Ano = 2025
Filtros de visual, página e relatório	Loja = Matriz
Interação entre visuais (cross-filter)	Clique em uma fatia da rosca
Filtros escritos em CALCULATE	CALCULATE([Receita], dProduto[Categoria] = "Bebidas")
Segurança em nível de linha (RLS)	Vendedor só enxerga a própria carteira

Contexto de linha

Contexto de linha é diferente: é a noção de 'linha atual'. Ele existe automaticamente dentro de uma coluna calculada e dentro de funções iteradoras (SUMX, AVERAGEX, FILTER). Em uma medida simples, ele NÃO existe.

DAX

```
-- ERRO: não existe 'linha atual' em uma medida
Receita Errada = fVendas[Quantidade] * fVendas[PrecoUnitario]

-- CERTO: o iterador cria contexto de linha
Receita = SUMX( fVendas, fVendas[Quantidade] * fVendas[PrecoUnitario] )
```

ATENÇÃO

Estar na linha do produto 'Teclado' dentro de um SUMX não filtra automaticamente outras tabelas. Para transformar contexto de linha em contexto de filtro, é preciso transição de contexto — que acontece quando você usa CALCULATE (ou chama uma medida) dentro do iterador.

Transição de contexto, na prática

DAX

-- Sem transição: soma a receita da tabela inteira para CADA cliente. Errado.

Teste Sem Transicao =

```
SUMX( dCliente, SUM( fVendas[Receita] ) )
```

-- Com transição: a medida [Receita] dispara CALCULATE implícito,

-- e a linha atual de dCliente vira filtro. Correto.

Teste Com Transicao =

```
SUMX( dCliente, [Receita] )
```

BOA PRÁTICA

Toda vez que você escreve o nome de uma medida entre colchetes dentro de um iterador, há transição de contexto. Isso é poderoso e caro: em tabelas com milhões de linhas, evite.

EXERCÍCIO PRÁTICO

Crie as medidas 'Teste Sem Transicao' e 'Teste Com Transicao' em uma tabela por cliente. Observe a diferença nos números e escreva com suas palavras por que ela acontece.

Coluna calculada x medida x tabela calculada

O erro conceitual mais comum de quem vem do Excel.

Aspecto	Coluna calculada	Medida
Quando calcula	Na atualização dos dados	No momento em que o visual pede
Onde fica	Armazenada na memória do modelo	Só a fórmula é armazenada
Contexto	De linha	De filtro
Responde a filtro do usuário	Não (o valor é fixo)	Sim
Uso	Eixo, filtro, segmentação, relacionamento	Valor exibido no visual
Custo	Memória	CPU no momento da consulta

BOA PRÁTICA

'Esse valor muda quando o usuário filtra?' Se sim → medida. Se não → coluna (de preferência feita no Power Query).

Exemplos lado a lado

DAX

```
-- Precisa referenciar outra tabela via relacionamento
Categoria Produto = RELATED( dProduto[Categoria] )

-- Vai virar eixo do gráfico
Faixa Etária =
SWITCH( TRUE(),
    dCliente[Idade] < 25, "1. Até 24",
    dCliente[Idade] < 40, "2. 25 a 39",
    dCliente[Idade] < 60, "3. 40 a 59",
    "4. 60 ou mais"
)

-- Coluna de ordenação para outra coluna de texto
Ordem Faixa Etária =
SWITCH( TRUE(),
    dCliente[Idade] < 25, 1,
    dCliente[Idade] < 40, 2,
    dCliente[Idade] < 60, 3,
    4
)
```

Tabelas calculadas

DAX

```
-- Tabela de apoio para segmentação sem relacionamento
dPeriodo =
DATATABLE(
    "Rotulo", STRING, "Dias", INTEGER, "Ordem", INTEGER,
    {
        { "Últimos 7 dias", 7, 1 },
        { "Últimos 30 dias", 30, 2 },
        { "Últimos 90 dias", 90, 3 },
        { "Últimos 365 dias", 365, 4 }
    }
)

-- Dimensão criada a partir de duas fatos
dVendedor =
DISTINCT(
    UNION(
        SELECTCOLUMNS( fVendas, "Vendedor", fVendas[Vendedor] ),
        SELECTCOLUMNS( fMetas, "Vendedor", fMetas[Vendedor] )
    )
)
```

ATENÇÃO

Ela é recalculada a cada atualização e não faz query folding. Para transformação de dados, use o Power Query. Reserve tabelas calculadas para apoio, parâmetros e dimensões derivadas do próprio modelo.

EXERCÍCIO PRÁTICO

Identifique no seu modelo três colunas calculadas que deveriam ser medidas e duas medidas que na verdade deveriam ser colunas. Faça a migração e meça o tamanho do arquivo antes e depois.

Agregações, DIVIDE e variáveis

As funções do dia a dia escritas do jeito profissional.

DAX

```
Receita Total      = SUM( fVendas[Receita] )
Custo Total        = SUM( fVendas[Custo] )
Qtd Vendida        = SUM( fVendas[Quantidade] )
Qtd Pedidos        = DISTINCTCOUNT( fVendas[NumeroPedido] )
Qtd Clientes       = DISTINCTCOUNT( fVendas[IdCliente] )
Linhas de Venda    = COUNTROWS( fVendas )
Ticket Médio       = DIVIDE( [Receita Total], [Qtd Pedidos] )
Preço Médio        = DIVIDE( [Receita Total], [Qtd Vendida] )
Margem Bruta       = [Receita Total] - [Custo Total]
Margem %           = DIVIDE( [Margem Bruta], [Receita Total] )
```

ATENÇÃO

[Receita] / [Pedidos] devolve erro ou infinito quando o denominador é zero, e isso aparece feio no visual. DIVIDE trata o caso e ainda aceita um terceiro argumento para o valor alternativo.

Variáveis: legibilidade e performance

VAR calcula uma vez e guarda o resultado. Além de deixar o código legível, evita recalculá-la a mesma expressão várias vezes — ganho real de performance em medidas complexas.

DAX

```
-- Difícil de ler e recalcula tudo três vezes
Crescimento Ruim =
DIVIDE(
    SUM(fVendas[Receita]) - CALCULATE(SUM(fVendas[Receita]), SAMEPERIODLASTYEAR(dCalendario[Date]
)),
    CALCULATE(SUM(fVendas[Receita]), SAMEPERIODLASTYEAR(dCalendario[Date]))
)

-- Legível, testável e mais rápido
Crescimento YoY % =
VAR Atual      = [Receita Total]
VAR Anterior   = CALCULATE( [Receita Total], SAMEPERIODLASTYEAR( dCalendario[Date] ) )
VAR Delta      = Atual - Anterior
RETURN
    IF( NOT ISBLANK( Anterior ), DIVIDE( Delta, Anterior ) )
```

BOA PRÁTICA

Uma VAR é avaliada no contexto em que foi declarada e não muda depois. Isso confunde no início, mas é justamente o que torna o código previsível. Se você precisa de um valor recalculado em outro contexto, calcule dentro do CALCULATE, não fora.

Contagens que você precisa distinguir

Função	Conta	Cuidado
COUNT(coluna)	Valores não vazios da coluna	Ignora nulos
COUNTA(coluna)	Valores não vazios, inclusive texto	—
COUNTROWS(tabela)	Linhas da tabela	Mais rápida e clara que COUNT em uma chave
DISTINCTCOUNT(coluna)	Valores distintos	Custosa em colunas de alta cardinalidade
COUNTBLANK(coluna)	Valores vazios	Ótima para auditoria de qualidade

Formatação e organização

- Crie uma tabela vazia chamada '_Medidas' e mova todas as medidas para lá.
- Use pastas de exibição: Base, Comparativos, Metas, Auxiliares.
- Defina o formato na própria medida (Ferramentas de Medida → Formato), não no visual.
- Padronize o nome: substantivo + qualificador. 'Receita LY', 'Receita YTD', 'Receita Meta'.

EXERCÍCIO PRÁTICO

Crie o conjunto completo de medidas-base acima em uma tabela '_Medidas', organize-as em pastas de exibição, defina os formatos e reescreva uma medida antiga usando variáveis.

CALCULATE: a função central do DAX

O que ele realmente faz com o contexto, e a ordem em que as coisas acontecem.

CALCULATE avalia uma expressão em um contexto de filtro modificado. Essa frase resume toda a linguagem. Tudo que é difícil em DAX é, no fundo, uma pergunta sobre o que CALCULATE fez com o contexto.

DAX

```
CALCULATE(  
    <expressão>,  
    <filtro1>,  
    <filtro2>,  
    ...  
)
```

O que acontece com o contexto existente

Por padrão, um filtro escrito em CALCULATE SUBSTITUI o filtro existente naquela coluna e MANTÉM os filtros das demais colunas.

DAX

```
-- Se o visual está filtrado em Categoria = "Bebidas" e Ano = 2025,  
-- esta medida devolve Alimentos de 2025 (a categoria é substituída, o ano permanece).  
Receita Alimentos =  
CALCULATE( [Receita Total], dProduto[Categoria] = "Alimentos" )
```

DAX

```
-- KEEPFILTERS respeita o filtro que já existia: se o usuário
-- selecionou "Bebidas", o resultado fica em branco (interseção vazia).
Receita Alimentos (respeitando) =
CALCULATE(
    [Receita Total],
    KEEPFILTERS( dProduto[Categoria] = "Alimentos" )
)
```

Os modificadores de filtro

Modificador	Efeito
REMOVEFILTERS(tabela ou coluna)	Remove os filtros — a forma moderna de ALL dentro de CALCULATE
ALL(tabela)	Ignora todos os filtros da tabela
ALLEXCEPT(tabela, colunas)	Ignora tudo, exceto as colunas listadas
ALLSELECTED()	Respeita o que o usuário selecionou, ignora o que o visual filtrou
KEEPFILTERS(filtro)	Interseção com o filtro existente em vez de substituição
USERELATIONSHIP(c1, c2)	Ativa um relacionamento inativo durante o cálculo
CROSSFILTER(c1, c2, direção)	Muda a direção do filtro durante o cálculo

DAX

```
-- % sobre o total geral, ignorando qualquer filtro de produto
% do Total Geral =
DIVIDE(
    [Receita Total],
    CALCULATE( [Receita Total], REMOVEFILTERS( dProduto ) )
)

-- % sobre o total do que está visível/selecionado no visual
% do Total Visível =
DIVIDE(
    [Receita Total],
    CALCULATE( [Receita Total], ALLSELECTED( dProduto ) )
)

-- % dentro da categoria (a subcategoria varia, a categoria fica fixa)
% da Categoria =
DIVIDE(
    [Receita Total],
    CALCULATE( [Receita Total], ALLEXCEPT( dProduto, dProduto[Categoria] ) )
)
```

BOA PRÁTICA

CALCULATE([Receita], dProduto[Categoria]="Bebidas") é açúcar para CALCULATE([Receita], FILTER(ALL(dProduto[Categoria]), dProduto[Categoria]="Bebidas")). Saber disso explica por que o filtro simples substitui o contexto naquela coluna.

ATENÇÃO

Para condições com duas colunas ou com medidas, você precisa escrever FILTER explicitamente. Ex.: `CALCULATE([Receita], FILTER(fVendas, fVendas[Qtd] > 10 && fVendas[Desconto] = 0))`.

EXERCÍCIO PRÁTICO

Crie as três medidas de participação (% do Total Geral, % do Total Visível e % da Categoria) e monte uma matriz com Categoria e Subcategoria. Aplique uma segmentação de categoria e explique a diferença entre os três resultados.

Quiz do módulo

1. O que é contexto de filtro?

- A ordem das colunas na tabela
- **✓ O conjunto de filtros aplicados quando a medida é avaliada**
- Um tipo de relacionamento
- A formatação do visual

Cada célula do visual avalia a medida em um contexto de filtro próprio.

2. Por que 'Receita = fVendas[Qtd] * fVendas[Preco]' falha como medida?

- Porque falta SUM em volta
- **✓ Porque medidas não têm contexto de linha — é preciso um iterador como SUMX**
- Porque as colunas são de tipos diferentes
- Porque falta CALCULATE

SUMX cria o contexto de linha necessário para multiplicar coluna por coluna.

3. Quando ocorre transição de contexto?

- Sempre que uma medida é usada
- **✓ Quando CALCULATE (explícito ou implícito pela chamada de uma medida) é executado dentro de um contexto de linha**
- Apenas em colunas calculadas
- Quando há relacionamento bidirecional

A transição converte a linha atual em filtro sobre o modelo.

4. Qual a vantagem de DIVIDE sobre a barra /?

- É mais rápida sempre
- **✓ Trata divisão por zero e permite definir um resultado alternativo**
- Aceita mais de dois argumentos
- Formata o resultado como percentual

Evita erros e infinitos aparecendo no visual.

5. Para calcular o % sobre o total do que o usuário selecionou (e não o total geral), use:

- REMOVEFILTERS

- ✓ **ALLSELECTED**
- ALLEXCEPT
- KEEPFILTERS

ALLSELECTED respeita as segmentações e ignora apenas o filtro imposto pelo próprio visual.

6. KEEPFILTERS serve para:

- Impedir que o usuário mude segmentações
- ✓ **Fazer o filtro do CALCULATE se somar (interseção) ao filtro existente em vez de substituí-lo**
- Congelar o visual
- Manter a formatação

Sem ele, o filtro do CALCULATE substitui o filtro existente naquela coluna.

DAX intermediário: filtros, relações e ranking

FILTER, ALL e variantes, VALUES, RELATED e RELATEDTABLE, RANKX, TOPN e as funções de informação que salvam medidas.

Objetivos de aprendizagem

- Escolher entre filtro simples e FILTER com consciência de performance
- Dominar a família ALL e saber exatamente o que cada uma remove
- Navegar entre tabelas com RELATED e RELATEDTABLE
- Construir rankings corretos, inclusive com desempate e em nível de total

FILTER e a família ALL

Quando escrever FILTER, quando não escrever e o que cada ALL remove de verdade.

FILTER

FILTER devolve uma tabela com as linhas que atendem a uma condição. Ele é um iterador: percorre linha a linha. Por isso é poderoso e caro.

DAX

```
-- Condição com duas colunas
Receita Grandes Pedidos Sem Desconto =
CALCULATE(
    [Receita Total],
    FILTER( fVendas, fVendas[Quantidade] > 10 && fVendas[Desconto] = 0 )
)

-- Condição que usa uma medida
Clientes Acima da Média =
VAR MediaGeral =
    AVERAGEX( ALL( dCliente ), [Receita Total] )
RETURN
COUNTROWS(
    FILTER( VALUES( dCliente[IdCliente] ), [Receita Total] > MediaGeral )
)
```

ATENÇÃO

Em uma fato de 20 milhões de linhas, isso é caro. Sempre que a condição envolver uma única coluna, prefira o filtro simples ou FILTER(VALUES(coluna), ...) — o motor trabalha sobre os valores distintos, não sobre as linhas.

DAX

```
-- Caro
CALCULATE( [Receita Total], FILTER( fVendas, fVendas[Quantidade] > 10 ) )

-- Barato: itera só os valores distintos de quantidade
CALCULATE( [Receita Total], FILTER( VALUES( fVendas[Quantidade] ), fVendas[Quantidade] > 10 ) )

-- Mais barato ainda quando possível
CALCULATE( [Receita Total], fVendas[Quantidade] > 10 )
```

A família ALL

Função	O que remove	Uso típico
ALL(Tabela)	Todos os filtros da tabela inteira	Total geral
ALL(Tabela[Coluna])	Filtros apenas daquela coluna	Total dentro de uma hierarquia
ALLEXCEPT(Tabela, Col)	Todos menos os das colunas listadas	% dentro do grupo
ALLSELECTED()	O filtro do visual, preservando as segmentações	% do visível
ALLNOBLANKROW(Tabela)	Como ALL, mas sem a linha em branco de chave órfã	Contagens de dimensão
REMOVEFILTERS(...)	Sinônimo moderno de ALL dentro de CALCULATE	Legibilidade

BOA PRÁTICA

Fora de CALCULATE, ALL devolve uma TABELA. Dentro de CALCULATE, funciona como um modificador que REMOVE filtros. Entender isso resolve metade das confusões com DAX.

DAX

```
-- Matriz com Produto nas linhas e segmentação de Categoria.
-- O usuário selecionou 3 categorias.
% do Visível =
DIVIDE(
    [Receita Total],
    CALCULATE( [Receita Total], ALLSELECTED( dProduto ) )
)
-- Denominador = total das 3 categorias selecionadas (não o total geral,
-- e não o valor do próprio produto).
```

EXERCÍCIO PRÁTICO

Monte uma matriz com Categoria e Produto. Crie quatro medidas de participação usando ALL, ALLEXCEPT, ALLSELECTED e REMOVEFILTERS e explique por escrito, com números, a diferença de cada uma.

VALUES, DISTINCT, RELATED e RELATEDTABLE

Navegando pelo modelo dentro do DAX.

VALUES e DISTINCT

As duas devolvem os valores únicos de uma coluna no contexto atual. A diferença: VALUES inclui a linha em branco criada por chaves órfãs; DISTINCT não.

DAX

```
-- Detectar se há apenas um valor selecionado
Título Dinâmico =
IF(
    HASONEVALUE( dProduto[Categoria] ),
    "Análise de " & VALUES( dProduto[Categoria] ),
    "Todas as categorias"
)

-- Forma moderna e segura da mesma coisa
Título Dinâmico 2 =
"Análise de " & SELECTEDVALUE( dProduto[Categoria], "todas as categorias" )

-- Contar categorias presentes no contexto
Qtd Categorias = COUNTROWS( VALUES( dProduto[Categoria] ) )
```

RELATED e RELATEDTABLE

Função	Direção	Onde usar
RELATED(coluna)	Do lado muitos para o lado um	Coluna calculada na fato buscando atributo da dimensão
RELATEDTABLE(tabela)	Do lado um para o lado muitos	Coluna calculada na dimensão agregando a fato

DAX

```
-- Coluna em fVendas (lado muitos): puxa um único valor da dimensão
Custo Padrão = RELATED( dProduto[CustoPadrao] )

-- Coluna em dCliente (lado um): agrega a fato relacionada
Qtd Pedidos do Cliente = COUNTROWS( RELATEDTABLE( fVendas ) )
Última Compra          = MAXX( RELATEDTABLE( fVendas ), fVendas[Data] )
```

ATENÇÃO

Ele só funciona dentro de uma coluna calculada ou de um iterador. Em uma medida simples, use CALCULATE ou uma medida com transição de contexto.

LOOKUPVALUE: quando não há relacionamento

DAX

```
Meta do Vendedor =
LOOKUPVALUE(
    fMetas[ValorMeta],
    fMetas[Vendedor], fVendas[Vendedor],
    fMetas[AnoMes],   FORMAT( fVendas[Data], "yyyy-mm" ),
    0                -- valor quando não encontra
)
```

BOA PRÁTICA

Ele é lento em volume. Se você está usando muito, o modelo está pedindo um relacionamento ou uma mesclagem no Power Query.

Funções de informação

DAX

```
Comentário Automático =  
VAR Cresceu = [Var. YoY %] > 0  
RETURN  
SWITCH( TRUE(),  
    ISBLANK( [Receita Total] ), "Sem movimento no período",  
    NOT HASONEVALUE( dProduto[Produto] ), "Selecione um produto para ver o detalhe",  
    ISINSCOPE( dProduto[Produto] ) && Cresceu,  
        SELECTEDVALUE( dProduto[Produto] ) & " cresceu " & FORMAT( [Var. YoY %], "0.0%" ),  
    SELECTEDVALUE( dProduto[Produto] ) & " caiu " & FORMAT( ABS([Var. YoY %]), "0.0%" )  
)
```

EXERCÍCIO PRÁTICO

Crie um título dinâmico que muda conforme a seleção do usuário (1 categoria, várias, ou nenhuma) e aplique-o com 'Formatação condicional → Título' em um visual.

RANKX, TOPN e análise de concentração

Rankings que funcionam também na linha de total — e a curva ABC.

DAX

```
Ranking Produto =  
RANKX(  
    ALL( dProduto[Produto] ), -- universo do ranking  
    [Receita Total],          -- expressão avaliada para cada item  
    ,                          -- valor (opcional)  
    DESC,                     -- ordem  
    DENSE                      -- empates ocupam a mesma posição, sem pular  
)
```

ATENÇÃO

No total não há um produto específico, e RANKX devolve algo sem sentido. Sempre proteja com ISINSCOPE ou HASONEVALUE.

DAX

```
Ranking Produto =  
IF(  
    ISINSCOPE( dProduto[Produto] ),  
    RANKX( ALL( dProduto[Produto] ), [Receita Total], , DESC, DENSE )  
)
```

DAX

```

Ranking na Categoria =
IF(
    ISINSCOPE( dProduto[Produto] ),
    RANKX(
        ALLEXCEPT( dProduto, dProduto[Categoria] ),
        [Receita Total], , DESC, DENSE
    )
)

```

DAX

```

Ranking Visível =
IF(
    ISINSCOPE( dProduto[Produto] ),
    RANKX( ALLSELECTED( dProduto[Produto] ), [Receita Total], , DESC, DENSE )
)

```

TOPN e o agrupamento 'Outros'

DAX

```

Receita Top 10 Clientes =
CALCULATE(
    [Receita Total],
    TOPN( 10, ALL( dCliente[Cliente] ), [Receita Total], DESC )
)

Receita Demais Clientes =
CALCULATE( [Receita Total], ALL( dCliente[Cliente] ) ) - [Receita Top 10 Clientes]

```

Curva ABC (Pareto)

DAX

```

Receita Acumulada Produtos =
VAR ReceitaAtual = [Receita Total]
VAR TabelaProdutos =
    ADDCOLUMNS( ALLSELECTED( dProduto[Produto] ), "@Receita", [Receita Total] )
RETURN
SUMX(
    FILTER( TabelaProdutos, [@Receita] >= ReceitaAtual ),
    [@Receita]
)

% Acumulado =
DIVIDE(
    [Receita Acumulada Produtos],
    CALCULATE( [Receita Total], ALLSELECTED( dProduto[Produto] ) )
)

Classe ABC =
VAR Acum = [% Acumulado]
RETURN
SWITCH( TRUE(),
    Acum <= 0.80, "A",
    Acum <= 0.95, "B",
    "C"
)

```

BOA PRÁTICA

Descobrir que 12% dos produtos respondem por 80% da receita reposiciona estoque, negociação com fornecedor e foco comercial. É um dos entregáveis de BI com maior retorno imediato.

EXERCÍCIO PRÁTICO

Monte uma tabela de produtos com Receita, Ranking, % Acumulado e Classe ABC. Ordene por receita decrescente e identifique quantos produtos formam a classe A.

Padrões de medida que resolvem 80% dos pedidos

Um repertório pronto para copiar, entender e adaptar.

Comparação com a média

DAX

```
Média por Categoria =  
AVERAGEX( ALLSELECTED( dProduto[Categoria] ), [Receita Total] )  
  
Desvio da Média % =  
DIVIDE( [Receita Total] - [Média por Categoria], [Média por Categoria] )
```

Contagens condicionais

DAX

```
Clientes Ativos =  
CALCULATE(  
    DISTINCTCOUNT( fVendas[IdCliente] ),  
    DATESINPERIOD( dCalendario[Date], MAX( dCalendario[Date] ), -90, DAY )  
)  
  
Clientes Inativos =  
VAR Todos = CALCULATE( DISTINCTCOUNT( fVendas[IdCliente] ), ALL( dCalendario ) )  
RETURN Todos - [Clientes Ativos]  
  
Produtos sem Venda =  
COUNTROWS(  
    FILTER( ALL( dProduto[Produto] ), ISBLANK( [Receita Total] ) )  
)
```

Novo x recorrente

DAX

```
Cientes Novos =
VAR InicioPeriodo = MIN( dCalendario[Date] )
RETURN
COUNTROWS(
    FILTER(
        VALUES( fVendas[IdCliente] ),
        CALCULATE(
            MIN( fVendas[Data] ),
            REMOVEFILTERS( dCalendario )
        ) >= InicioPeriodo
    )
)

Clientes Recorrentes = [Qtd Clientes] - [Clientes Novos]

Receita de Clientes Novos =
CALCULATE(
    [Receita Total],
    FILTER(
        VALUES( fVendas[IdCliente] ),
        CALCULATE( MIN( fVendas[Data] ), REMOVEFILTERS( dCalendario ) ) >= MIN( dCalendario[Date]
    ] )
)
)
```

Último valor conhecido (semiaditivo)

DAX

```
Saldo Estoque =
CALCULATE(
    SUM( fEstoque[Quantidade] ),
    LASTNONBLANK( dCalendario[Date], CALCULATE( SUM( fEstoque[Quantidade] ) ) )
)

Saldo Estoque (fim do mês) =
CALCULATE(
    SUM( fEstoque[Quantidade] ),
    LASTDATE( dCalendario[Date] )
)
```

BOA PRÁTICA

Guarde essas medidas em um arquivo .pbix 'Biblioteca DAX'. Com o tempo você para de escrever DAX do zero e passa a adaptar padrões — que é exatamente o que fazem os profissionais experientes.

EXERCÍCIO PRÁTICO

Implemente as medidas de Clientes Novos, Recorrentes e Receita de Clientes Novos. Monte um gráfico de colunas empilhadas por mês mostrando a composição da receita entre novos e recorrentes.

Quiz do módulo

1. Por que FILTER(fVendas, ...) pode ser lento?

- Porque não aceita medidas
- **✓ Porque itera linha a linha a tabela inteira**
- Porque só funciona em DirectQuery
- Porque quebra o relacionamento

Quando possível, filtre sobre VALUES(coluna) ou use filtro simples.

2. Qual função respeita as segmentações do usuário mas ignora o filtro do próprio visual?

- ALL
- ALLEXCEPT
- **✓ ALLSELECTED**
- ALLNOBLANKROW

É exatamente o comportamento necessário para '% do visível'.

3. RELATEDTABLE é usada:

- Na fato, para buscar atributo da dimensão
- **✓ Na dimensão, para agregar as linhas relacionadas da fato**
- Apenas em medidas
- Para criar relacionamentos

Ela vai do lado 'um' para o lado 'muitos'.

4. O ranking mostra '1' na linha de total. Como corrigir?

- Trocar DESC por ASC
- **✓ Envolver em IF com ISINSCOPE ou HASONEVALUE**
- Usar RANK em vez de RANKX
- Remover o ALL

No total não existe um item específico; a medida deve devolver vazio.

5. Na curva ABC, a classe A normalmente representa:

- Os produtos mais caros
- **✓ Os itens que acumulam até cerca de 80% da receita**
- Os últimos 20% da lista
- Os produtos sem venda

É o princípio de Pareto aplicado ao portfólio.

6. Qual a diferença entre VALUES e DISTINCT?

- VALUES é mais rápida
- **✓ VALUES inclui a linha em branco de chaves órfãs; DISTINCT não**
- DISTINCT só funciona em texto
- Não há diferença

Isso importa em contagens de dimensão com integridade referencial imperfeita.

DAX avançado: iteradores, tabelas virtuais e metas

SUMX e a família X, tabelas virtuais com ADDCOLUMNS e SUMMARIZE, TREATAS, medidas dinâmicas com SWITCH e todo o padrão de meta e semáforo.

Objetivos de aprendizagem

- Usar iteradores com consciência de custo e de contexto
- Construir tabelas virtuais para cálculos que o modelo não entrega direto
- Aplicar TREATAS para relacionar tabelas sem relacionamento físico
- Implementar o padrão completo de meta, atingimento, gap e semáforo

Iteradores: a família X

SUMX, AVERAGEX, MAXX, RANKX, CONCATENATEX — e o custo real de iterar.

Um iterador percorre uma tabela linha a linha, avalia uma expressão em cada linha e agrega o resultado. É o que permite cálculos que a agregação simples não alcança.

DAX

```
-- Não existe coluna de receita: ela nasce da multiplicação linha a linha
Receita = SUMX( fVendas, fVendas[Quantidade] * fVendas[PrecoUnitario] )

-- Cálculo com regra por linha
Receita Líquida =
SUMX(
    fVendas,
    fVendas[Quantidade] * fVendas[PrecoUnitario] * ( 1 - fVendas[PercDesconto] )
)

-- Margem por linha, considerando o custo vigente na data
Margem Real =
SUMX(
    fVendas,
    VAR Preco = fVendas[PrecoUnitario]
    VAR Custo = RELATED( dProduto[CustoPadrao] )
    RETURN fVendas[Quantidade] * ( Preco - Custo )
)
```

Iterando dimensões em vez da fato

DAX

```
-- ERRADO: média das linhas de venda
Média Errada = AVERAGE( fVendas[Receita] )

-- CERTO: média do total de cada loja
Receita Média por Loja =
AVERAGEX( VALUES( dLoja[Loja] ), [Receita Total] )
```

ATENÇÃO

AVERAGEX sobre 200 lojas faz 200 avaliações — tranquilo. O mesmo sobre 3 milhões de linhas de venda derruba o relatório. Itere a menor tabela que responde à pergunta.

Outros iteradores muito úteis

DAX

```
Melhor Mês do Ano =  
MAXX( VALUES( dCalendario[AnoMes] ), [Receita Total] )  
  
Nome do Melhor Mês =  
VAR Melhor = [Melhor Mês do Ano]  
RETURN  
CONCATENATEX(  
    FILTER( VALUES( dCalendario[AnoMes] ), [Receita Total] = Melhor ),  
    dCalendario[AnoMes], " , "  
)  
  
Categorias Seleccionadas =  
CONCATENATEX( VALUES( dProduto[Categoria] ), dProduto[Categoria], " • ", dProduto[Categoria], AS  
C )  
  
Dias com Venda = COUNTX( VALUES( dCalendario[Date] ), IF( [Receita Total] > 0, 1 ) )
```

BOA PRÁTICA

Mostrar 'Filtros aplicados: Bebidas • Higiene • Limpeza' no canto do relatório elimina metade das dúvidas dos usuários sobre 'por que esse número está diferente'.

EXERCÍCIO PRÁTICO

Crie as medidas Receita Média por Loja e Melhor Mês do Ano. Depois escreva um texto dinâmico com CONCATENATEX mostrando quais categorias estão seleccionadas no momento.

Tabelas virtuais: ADDCOLUMNS, SUMMARIZE e GENERATE

Criar uma tabela temporária na memória para responder ao que o modelo não responde direto.

Tabela virtual é uma tabela que existe apenas durante a avaliação da medida. É o recurso mais poderoso do DAX avançado: você monta o dataset que precisa e então agrega sobre ele.

DAX

```
Clientes Acima de 100 mil =  
VAR Base =  
    ADDCOLUMNS(  
        SUMMARIZE( fVendas, dCliente[Cliente] ),  
        "@Receita", [Receita Total]  
    )  
RETURN  
COUNTROWS( FILTER( Base, [@Receita] > 100000 ) )
```

ATENÇÃO

SUMMARIZE(fVendas, dCliente[Cliente], "Receita", SUM(...)) funciona, mas tem comportamento de contexto imprevisível e é oficialmente desaconselhado. Use SUMMARIZE só para agrupar e ADDCOLUMNS para calcular.

DAX

```
Receita do Segundo Melhor Mês =  
VAR Meses =  
    ADDCOLUMNS( ALLSELECTED( dCalendario[AnoMes] ), "@Receita", [Receita Total] )  
VAR Ordenado = TOPN( 2, Meses, [@Receita], DESC )  
VAR Segundo = MINX( Ordenado, [@Receita] )  
RETURN Segundo
```

Análise de coorte com GENERATE

DAX

```
Retenção da Coorte =  
VAR MesAnalise = SELECTEDVALUE( dCalendario[AnoMes] )  
VAR ClientesComPrimeiraCompra =  
    ADDCOLUMNS(  
        VALUES( fVendas[IdCliente] ),  
        "@PrimeiraCompra",  
        CALCULATE( MIN( fVendas[Data] ), REMOVEFILTERS( dCalendario ) )  
    )  
VAR CoorteSelecionada =  
    FILTER(  
        ClientesComPrimeiraCompra,  
        FORMAT( [@PrimeiraCompra], "yyyy-mm" ) = MesAnalise  
    )  
RETURN  
COUNTROWS( CoorteSelecionada )
```

Comparação linha a linha entre dois períodos

DAX

```
Produtos em Crescimento =  
VAR Base =  
    ADDCOLUMNS(  
        VALUES( dProduto[Produto] ),  
        "@Atual", [Receita Total],  
        "@Anterior", CALCULATE( [Receita Total], SAMEPERIODLASTYEAR( dCalendario[Date] ) )  
    )  
RETURN  
COUNTROWS(  
    FILTER( Base, [@Atual] > [@Anterior] && NOT ISBLANK( [@Anterior] ) )  
)
```

BOA PRÁTICA

Para conferir uma tabela virtual, crie temporariamente uma tabela calculada com a mesma expressão (EVALUATE mental). Você vê as linhas e as colunas @ e descobre o erro em segundos.

EXERCÍCIO PRÁTICO

Crie a medida 'Produtos em Crescimento' e a complementar 'Produtos em Queda'. Depois monte um cartão para cada uma e valide o total contra a contagem de produtos com venda nos dois períodos.

TREATAS, medidas dinâmicas e segmentação por medida

Relacionar sem relacionamento e deixar o usuário escolher o que a medida calcula.

TREATAS

TREATAS aplica os valores de uma tabela como filtro sobre uma coluna de outra tabela. É a forma mais limpa de criar um relacionamento virtual, sem poluir o modelo com relacionamentos extras.

DAX

```
Meta do Período =  
CALCULATE(  
    SUM( fMetas[ValorMeta] ),  
    TREATAS(  
        VALUES( dCalendario[AnoMes] ),  
        fMetas[AnoMes]  
    ),  
    TREATAS(  
        VALUES( dVendedor[Vendedor] ),  
        fMetas[Vendedor]  
    )  
)
```

BOA PRÁTICA

Relacionamento físico é mais rápido e deve ser o padrão. Use TREATAS quando o relacionamento criaria ambiguidade, quando a granularidade não bate, ou quando você precisa de um vínculo apenas dentro de uma medida específica.

Medidas dinâmicas com SWITCH

Em vez de criar 4 páginas iguais mudando só o indicador, crie uma tabela de seleção e deixe o usuário escolher. Menos manutenção, mais liberdade.

DAX

```
-- dIndicador: tabela desconectada com Indicador (texto) e Ordem (número)
Indicador Selecionado =
VAR Escolha = SELECTEDVALUE( dIndicador[Indicador], "Receita" )
RETURN
SWITCH( Escolha,
    "Receita",      [Receita Total],
    "Margem",       [Margem Bruta],
    "Margem %",     [Margem %],
    "Qtd Pedidos",  [Qtd Pedidos],
    "Ticket Médio", [Ticket Médio],
    BLANK()
)

-- Formato dinâmico (Power BI moderno: Ferramentas de Medida → Formato Dinâmico)
Formato do Indicador =
SWITCH( SELECTEDVALUE( dIndicador[Indicador], "Receita" ),
    "Margem %", "0.0%",
    "Qtd Pedidos", "#,##0",
    "\R$ #,##0"
)
```

ATENÇÃO

Ao usar medida dinâmica, o eixo Y continua escrito 'Indicador Selecionado'. Use formatação condicional no título do visual com uma medida de texto para exibir o nome real do indicador escolhido.

Parâmetros de campo (Field Parameters)

Recurso nativo: Modelagem → Novo parâmetro → Campos. Ele cria uma tabela onde cada linha é um campo ou medida. O usuário troca via segmentação e o visual muda de dimensão — sem nenhum SWITCH.

- Use parâmetro de campo quando quiser trocar a DIMENSÃO do eixo (produto, região, vendedor).
- Use SWITCH quando quiser trocar a MEDIDA com lógica ou formatação própria.
- Você pode combinar os dois na mesma página: um seletor de eixo e um de indicador.

DAX

```
-- dFaixaTicket: tabela desconectada com Rotulo, Min, Max, Ordem
Pedidos na Faixa =
VAR MinFaixa = SELECTEDVALUE( dFaixaTicket[Min] )
VAR MaxFaixa = SELECTEDVALUE( dFaixaTicket[Max] )
RETURN
CALCULATE(
    [Qtd Pedidos],
    FILTER(
        VALUES( fVendas[NumeroPedido] ),
        VAR ValorPedido = CALCULATE( [Receita Total] )
        RETURN ValorPedido >= MinFaixa && ValorPedido < MaxFaixa
    )
)
```

EXERCÍCIO PRÁTICO

Crie a tabela dIndicador com 5 opções, a medida dinâmica com SWITCH e um título que mostra o nome do indicador escolhido. Depois adicione um parâmetro de campo para trocar a dimensão do eixo.

Metas, atingimento e semáforos

O padrão completo de gestão por indicador, do gap ao ícone colorido.

O conjunto de medidas de meta

DAX

```
Meta = SUM( fMetas[ValorMeta] )

Atingimento % = DIVIDE( [Receita Total], [Meta] )

Gap para Meta = [Receita Total] - [Meta]

Meta Proporcional ao Período =
-- No dia 12 de um mês de 30 dias, a meta esperada é 40% do total
VAR DiasDecorridos =
    CALCULATE(
        COUNTROWS( dCalendario ),
        dCalendario[Date] <= MIN( MAX( dCalendario[Date] ), TODAY() )
    )
VAR DiasTotais = COUNTROWS( dCalendario )
RETURN [Meta] * DIVIDE( DiasDecorridos, DiasTotais )

Atingimento Proporcional % =
DIVIDE( [Receita Total], [Meta Proporcional ao Período] )

Projeção de Fechamento =
VAR DiasDecorridos =
    CALCULATE( COUNTROWS( dCalendario ), dCalendario[Date] <= TODAY() )
VAR DiasTotais = COUNTROWS( dCalendario )
RETURN DIVIDE( [Receita Total], DiasDecorridos ) * DiasTotais
```

BOA PRÁTICA

No dia 5 do mês, 20% da meta pode ser excelente. Comparar o realizado contra a meta cheia todo dia gera alarme falso e faz a gestão ignorar o painel.

Semáforo

DAX

```
Status Meta =  
VAR Ating = [Atingimento Proporcional %]  
RETURN  
SWITCH( TRUE(),  
    ISBLANK( [Meta] ), "Sem meta",  
    Ating >= 1,        "Atingido",  
    Ating >= 0.9,      "Atenção",  
    "Crítico"  
)  
  
Cor do Status =  
SWITCH( [Status Meta],  
    "Atingido", "#2E9E6B",  
    "Atenção",  "#E0A526",  
    "Crítico",  "#D0453B",  
    "#8A8F98"  
)  
  
Ícone Status =  
SWITCH( [Status Meta],  
    "Atingido", "▲",  
    "Atenção",  "■",  
    "Crítico",  "▼",  
    "_"  
)
```

ATENÇÃO

Cerca de 8% dos homens têm alguma deficiência na percepção de cores. Combine sempre cor + ícone + texto. Isso não é detalhe de acessibilidade: é requisito de um painel profissional.

Aplicando no visual

1. Selecione o visual → Formatar → Cores dos dados → fx (formatação condicional).
2. Formatar por: Valor do campo.
3. Escolha a medida [Cor do Status].
4. Para tabelas e matrizes, repita em 'Cor da fonte' e 'Cor do plano de fundo'.
5. Adicione uma coluna com [Ícone Status] para reforço não-cromático.

Gráfico de bala (bullet chart)

O visual KPI nativo é limitado. Para comparar realizado, meta e faixas de desempenho em pouco espaço, use o gráfico de linhas e colunas agrupadas: colunas = realizado, linha = meta. Ou o visual 'Bullet chart' do AppSource quando precisar de faixas.

EXERCÍCIO PRÁTICO

Implemente o conjunto completo de meta (Meta, Atingimento %, Meta Proporcional, Gap, Projeção, Status e Cor) e aplique a formatação condicional em uma matriz por vendedor. Valide o comportamento em um mês em andamento.

Quiz do módulo

1. Por que AVERAGE(fvendas[Receita]) não dá a média por loja?

- Porque AVERAGE não aceita moeda
- **✓ Porque calcula a média das linhas de venda, não do total de cada loja**
- Porque falta CALCULATE
- Porque a coluna está oculta

Para média por entidade, itere a dimensão com AVERAGEX.

2. Por que não agregar dentro de SUMMARIZE?

- Porque a sintaxe não permite
- **✓ Porque o comportamento de contexto é imprevisível; use ADDCOLUMNS para calcular**
- Porque é mais lento sempre
- Porque não aceita mais de uma coluna

O padrão recomendado é SUMMARIZE para agrupar e ADDCOLUMNS para calcular.

3. TREATAS serve para:

- Criar colunas calculadas
- **✓ Aplicar os valores de uma tabela como filtro sobre a coluna de outra, sem relacionamento físico**
- Formatar números
- Ordenar tabelas

É o relacionamento virtual, criado apenas dentro da medida.

4. No dia 10 de um mês de 30 dias, comparar o realizado contra a meta cheia:

- É a prática correta
- **✓ Gera alarme falso; use meta proporcional aos dias decorridos**
- Só vale para vendas
- Depende do visual

Meta proporcional dá leitura honesta do ritmo durante o período.

5. Por que não comunicar status apenas por cor?

- Porque cores não aparecem no celular
- **✓ Porque parte dos usuários tem deficiência na percepção de cores; combine cor, ícone e texto**
- Porque cores deixam o relatório lento
- Porque o Power BI não suporta

Acessibilidade é requisito, não enfeite.

6. Para trocar a DIMENSÃO do eixo conforme a escolha do usuário, o recurso mais direto é:

- SWITCH com medidas
- **✓ Parâmetro de campo**
- Bookmark
- Drill-through

Parâmetros de campo são nativos e trocam campos do eixo sem código.

Design de dashboards que convencem

Escolha do gráfico certo, hierarquia visual, grid, cor com propósito, tipografia e a diferença entre um painel bonito e um painel eficaz.

Objetivos de aprendizagem

- Escolher o visual certo para cada tipo de pergunta
- Aplicar hierarquia visual e o padrão de leitura em Z
- Usar cor como informação, não como decoração
- Construir um layout em grid que funciona em qualquer resolução

Escolhendo o gráfico certo

Um mapa de decisão por tipo de pergunta — e os visuais que você deve evitar.

Pergunta	Visual recomendado	Por quê
Quanto, comparando categorias?	Barras horizontais ordenadas	O olho compara comprimento com precisão; rótulos longos cabem
Como evoluiu no tempo?	Linha	Continuidade temporal é o que a linha comunica
Composição ao longo do tempo?	Área empilhada ou colunas empilhadas	Mostra total e partes simultaneamente
Parte do todo, em um momento?	Barras 100% empilhadas (ou rosca com até 4 fatias)	Pizza com 9 fatias é ilegível
Relação entre duas variáveis?	Dispersão (scatter)	Revela correlação e outliers
Um número contra uma meta?	Cartão + gráfico de bala ou KPI	Contexto junto do número
Distribuição de valores?	Histograma ou box plot	Mostra concentração e cauda
Detalhe linha a linha?	Tabela ou matriz	Quando o usuário precisa do número exato
Onde, geograficamente?	Mapa de bolhas ou coroplético	Só se a geografia for relevante para a decisão
Fluxo entre etapas?	Funil ou gráfico de cascata	Mostra perda entre estágios

Visuais que você deve evitar (quase sempre)

- Pizza com mais de 4 fatias: o olho humano não compara ângulos bem.
- Gráfico 3D: a perspectiva distorce os valores. Não existe justificativa técnica.
- Medidor (gauge) sem faixa de referência: ocupa muito espaço para exibir um número.
- Eixo Y que não começa em zero em gráfico de barras: exagera diferenças e é considerado manipulação.
- Nuvem de palavras: bonita, pouco informativa.

- Dois eixos Y com escalas diferentes: cria correlações visuais que não existem.

BOA PRÁTICA

Um gráfico de barras ordenado por receita decrescente responde 'quem são os maiores?' instantaneamente. Ordenado por nome, obriga o usuário a comparar item por item. A exceção é a série temporal, que segue a ordem cronológica.

Densidade de informação

A regra prática: 4 a 6 visuais por página. Acima disso, o usuário deixa de ler e passa a escanear. Se a página tem 12 visuais, provavelmente são duas páginas — ou uma página com drill-through.

ATENÇÃO

Mostre a página por 10 segundos para alguém da área e pergunte: 'está bom ou ruim, e por quê?'. Se a pessoa não souber responder, a hierarquia visual falhou — não é problema de dado.

EXERCÍCIO PRÁTICO

Pegue um painel existente e liste cada visual com a pergunta que ele responde. Substitua pelo menos dois visuais por opções mais adequadas e compare a leitura antes e depois.

Hierarquia visual e layout em grid

Como guiar o olho do usuário do mais importante para o detalhe.

O padrão de leitura

No ocidente, o olho entra pelo canto superior esquerdo e percorre a página em Z. Use isso: o indicador mais importante fica em cima à esquerda; o detalhe operacional embaixo à direita.

Zona	Conteúdo	Altura sugerida (1080p)
Topo (faixa)	Título, período analisado, filtros aplicados	60-80 px
Faixa de KPIs	4 a 6 cartões com valor, variação e meta	120-140 px
Meio	Gráfico principal (evolução) + composição	320-380 px
Base	Detalhe: tabela, ranking, drill	260-320 px
Lateral esquerda	Navegação e segmentações	220-260 px de largura

Grid de 12 colunas

Trabalhe com uma grade mental de 12 colunas e espaçamento constante (8 ou 16 px). Visuais que ocupam 3, 4, 6 ou 12 colunas se alinham naturalmente e o painel ganha aparência profissional sem esforço.

1. Exibição → Ajustar à grade e Mostrar linhas de grade, ativados.

2. Defina o tamanho da página em Formatar página → Tamanho da tela (16:9, 1280x720 é o padrão seguro).
3. Use o painel Seleção para nomear e organizar as camadas.
4. Use Alinhar → Distribuir horizontalmente para espaçamento idêntico.
5. Agrupe visuais relacionados (Ctrl + G) para mover em bloco.

BOA PRÁTICA

Respiro entre blocos é o que torna a informação legível. Painéis apertados parecem 'cheios de dados' e comunicam menos. Reduza o número de visuais antes de reduzir as margens.

Tipografia

- Use no máximo duas famílias tipográficas — na prática, Segoe UI resolve tudo.
- Escala clara: título 20-24 pt, subtítulo 14 pt, rótulo 10-11 pt, valor de cartão 28-36 pt.
- Nunca use texto abaixo de 9 pt: é ilegível em projetor e em celular.
- Alinhe números à direita; texto à esquerda. Sempre.
- Abrevie valores grandes: R\$ 2,4 mi é mais legível que R\$ 2.412.983,44 em um cartão.

ATENÇÃO

'Gráfico 1' não ajuda ninguém. Prefira títulos que já contam a conclusão: 'Receita cresceu 12% no trimestre'. Com uma medida de texto dinâmica, o título se atualiza sozinho.

EXERCÍCIO PRÁTICO

Redesenhe uma página existente seguindo a divisão em zonas e a grade de 12 colunas. Fotografe antes e depois e liste três melhorias concretas de leitura.

Cor com propósito e acessibilidade

Paleta, contraste e o uso de cor como informação.

A regra das três funções da cor

- Cor de marca: identidade, usada em elementos estruturais (cabeçalho, linha divisória).
- Cor de destaque: apenas no que precisa de atenção. Se tudo é destaque, nada é.
- Cor semântica: verde/amarelo/vermelho para status, azul para o período atual, cinza para o comparativo.

BOA PRÁTICA

Deixe as séries de contexto em cinza e destaque em cor apenas a série que importa. É a mudança mais barata e mais eficaz que você pode fazer em um painel.

Construindo a paleta

Papel	Quantidade	Observação
Neutros	4 tons	Fundo, superfície, borda, texto secundário

Papel	Quantidade	Observação
Primária	1 + 2 variações	Identidade e séries principais
Semânticas	3	Positivo, atenção, negativo
Catégoricas	6 a 8	Para séries sem ordem; devem ser distinguíveis entre si
Sequenciais	1 escala	Para mapas de calor e coropléticos

ATENÇÃO

É exatamente o par que pessoas com daltonismo deuteranopia não distinguem. Prefira azul/laranja para oposição, e reforce sempre com ícone ou texto.

Contraste

- Texto normal: contraste mínimo de 4,5:1 contra o fundo.
- Texto grande (18 pt+) e elementos gráficos: mínimo de 3:1.
- Teste seu tema com um verificador de contraste antes de publicar.
- Em fundo escuro, evite branco puro (#FFFFFF): use um cinza muito claro para reduzir o brilho.

Checklist de acessibilidade do Power BI

1. Defina o texto alternativo (Alt Text) em cada visual — leitores de tela usam isso.
2. Ajuste a ordem de tabulação no painel Seleção para uma navegação lógica por teclado.
3. Adicione marcadores diferentes nas séries de linha, não só cores.
4. Use a exibição 'Mostrar como tabela' (Alt+Shift+F11) para conferir que o dado é acessível.
5. Teste o relatório com zoom de 200%.

EXERCÍCIO PRÁTICO

Construa uma paleta completa para o seu projeto (neutros, primária, semânticas, catégoricas), verifique todos os contrastes e aplique em uma página. Depois teste-a simulando daltonismo.

Storytelling com dados

Transformar uma página de números em um argumento que move a reunião.

Painel bom responde perguntas. Painel excelente antecipa a próxima. A diferença está em organizar a informação como um argumento: contexto → o que aconteceu → por quê → o que fazer.

A estrutura em quatro camadas

Camada	Pergunta	Elemento
Contexto	Do que estamos falando?	Título com período, filtros aplicados visíveis

Camada	Pergunta	Elemento
Sinal	Está bom ou ruim?	KPIs com variação e meta, cor semântica
Explicação	Por que está assim?	Decomposição, comparativos, cascata
Ação	O que fazer agora?	Ranking dos piores, lista de exceções, drill para o detalhe

Textos dinâmicos que explicam

DAX

```

Resumo Executivo =
VAR Atual = [Receita Total]
VAR Variacao = [Var. YoY %]
VAR Direcao = IF( Variacao >= 0, "crescimento", "queda" )
VAR Meta = [Atingimento Proporcional %]
VAR StatusMeta =
    SWITCH( TRUE(),
        Meta >= 1, "acima da meta proporcional",
        Meta >= 0.9, "próximo da meta",
        "abaixo da meta"
    )
VAR MaiorCategoria =
    CALCULATE(
        SELECTEDVALUE( dProduto[Categoria] ),
        TOPN( 1, ALLSELECTED( dProduto[Categoria] ), [Receita Total], DESC )
    )
RETURN
"Receita de " & FORMAT( Atual, "R$ #,##0" ) &
", " & Direcao & " de " & FORMAT( ABS(Variacao), "0.0%" ) & " sobre o ano anterior, " &
StatusMeta & ". Maior contribuição: " & MaiorCategoria & "."

```

BOA PRÁTICA

Uma caixa de texto com essa medida no alto da página faz o diretor entender o cenário antes de olhar qualquer gráfico. É o recurso de maior impacto percebido por menor esforço técnico.

Anotações e referência

- Linha de meta no gráfico (Analytics → Linha constante ou Linha de medida).
- Faixa sombreada indicando o período de comparação.
- Rótulo apenas no primeiro e no último ponto da linha, em vez de todos.
- Nota de rodapé com a origem e a data da última atualização — credibilidade instantânea.

DAX

```

Última Atualização =
"Dados atualizados em " &
FORMAT( MAX( fVendas[Data] ), "dd/mm/yyyy" ) &
" • Fonte: ERP Vendas"

```

ATENÇÃO

NOW() é avaliado na hora da consulta, não na hora do refresh. Para mostrar quando os dados foram carregados, crie uma coluna com DateTime.LocalNow() no Power Query.

EXERCÍCIO PRÁTICO

Crie a medida de Resumo Executivo adaptada ao seu contexto e posicione-a no topo da página principal. Peça a um colega da área de negócio para ler só o resumo e dizer o que entendeu.

Quiz do módulo

1. Para comparar a receita de 15 categorias, o melhor visual é:

- Pizza
- ✓ **Barras horizontais ordenadas por valor**
- Gráfico 3D
- Medidor

Comparação de comprimento é precisa e comporta rótulos longos.

2. Por que não começar o eixo Y fora do zero em um gráfico de barras?

- Porque o Power BI não permite
- ✓ **Porque exagera visualmente as diferenças e distorce a leitura**
- Porque deixa o visual lento
- Porque quebra a formatação condicional

O comprimento da barra deixa de ser proporcional ao valor.

3. Qual é o uso mais eficaz do cinza em um painel?

- Cor de fundo apenas
- ✓ **Deixar as séries de contexto em cinza e destacar em cor só o que importa**
- Texto de título
- Bordas dos cartões

Contraste seletivo direciona a atenção sem poluir.

4. Verde e vermelho como único código de status é problemático porque:

- São cores muito fortes
- ✓ **É justamente o par que pessoas com daltonismo mais comum não distinguem**
- Não combinam com a marca
- Deixam a impressão ruim

Reforce sempre com ícone e texto além da cor.

5. Quantos visuais por página é uma boa referência?

- 1 a 2
- ✓ **4 a 6**
- 10 a 12

- Quantos couberem

Acima disso o usuário escaneia em vez de ler.

6. Para mostrar quando os dados foram carregados, o correto é:

- Medida com NOW()
- ✓ **Coluna criada no Power Query com DateTime.LocalNow()**
- Texto fixo atualizado manualmente
- TODAY() em uma coluna calculada

NOW() em medida mostra a hora da consulta, não a do refresh.

Visuais avançados e interatividade

Indicadores, favoritos, drill-through, tooltips de página, botões e navegação — tudo que transforma um relatório estático em uma aplicação.

Objetivos de aprendizagem

- Controlar a interação entre visuais
- Construir navegação com botões, indicadores e favoritos
- Criar drill-through e tooltips de página que dão contexto sem poluir
- Usar painéis retráteis e seletores para economizar espaço

Interações, drill-down e drill-through

Quem filtra quem, e como levar o usuário ao detalhe sem perder o contexto.

Editar interações

Por padrão, todo visual filtra todos os outros. Isso raramente é o desejado. Selecione um visual → Formatar → Editar interações e defina, para cada um dos demais, se ele deve Filtrar, Destacar ou Não interagir.

- Filtrar: remove as linhas que não pertencem à seleção. É o padrão em tabelas.
- Destacar: mantém o total e sombreia a parte selecionada. Excelente em colunas empilhadas.
- Não interagir: use em cartões de contexto geral (por exemplo, 'Meta do mês') que não devem mudar.

BOA PRÁTICA

Se o usuário clica em uma categoria e a linha do tempo some, ele perde a referência. Destacar preserva o todo e mostra a parte.

Drill-down

1. Arraste mais de um campo para o eixo, formando uma hierarquia (Ano → Trimestre → Mês).
2. Use os controles no canto do visual: seta para baixo (drill por item), setas duplas (expandir todo o nível).
3. 'Expandir todo o nível' mostra a concatenação (Jan 2026, Fev 2026) — útil em barras.
4. Ative 'Mostrar rótulo do nível de detalhamento' para o usuário saber onde está.

Drill-through

Drill-through leva o usuário a uma página de detalhe, passando o contexto do item clicado. É o recurso mais subutilizado do Power BI e o que mais reduz poluição de página.

1. Crie uma página nova (ex.: 'Detalhe do Cliente') e oculte-a da barra de guias.
2. No painel Visualizações, arraste o campo dCliente[Cliente] para 'Detalhamento'.
3. Mantenha ativado 'Manter todos os filtros' para preservar o período selecionado.

4. O Power BI adiciona o botão de voltar automaticamente — posicione-o no canto superior esquerdo.
5. Na página de origem, clique com o botão direito no item → Detalhar → Detalhe do Cliente.

DAX

```
Título Detalhe =  
VAR Cliente = SELECTEDVALUE( dCliente[Cliente], "Todos os clientes" )  
VAR Período =  
    FORMAT( MIN( dCalendario[Date] ), "dd/mm/yyyy" ) & " a " &  
    FORMAT( MAX( dCalendario[Date] ), "dd/mm/yyyy" )  
RETURN Cliente & " • " & Período
```

ATENÇÃO

Se você colocar Cliente e Produto como campos de detalhamento, o menu só habilita quando o contexto fornece os dois. Para opções independentes, crie duas páginas de detalhe.

EXERCÍCIO PRÁTICO

Crie uma página de detalhe por cliente com drill-through, título dinâmico e botão de retorno. Ajuste as interações da página principal para que o gráfico de tendência use 'Destacar'.

Indicadores (bookmarks), botões e navegação

Construir menus, alternar visuais e criar painéis retráteis.

Indicadores

Um indicador guarda o estado da página: filtros, seleção, visibilidade dos objetos e ordenação. Combinado com botões, vira navegação.

1. Exibição → Indicadores e Exibição → Seleção (mantenha os dois painéis abertos).
2. Organize a visibilidade dos objetos no painel Seleção (olho ligado/desligado).
3. Adicionar em Indicadores e renomeie de forma descritiva ('Ver como gráfico').
4. Clique nos três pontos do indicador e desmarque o que ele NÃO deve controlar — em geral, desmarque 'Dados' e deixe só 'Exibição' e 'Objetos atuais'.
5. Insira um botão → Ação → Tipo: Indicador → escolha o indicador.

ATENÇÃO

Deixar 'Dados' marcado faz o indicador congelar os filtros do momento em que foi criado. O usuário troca o mês, clica no botão e o painel volta para o mês antigo. Desmarque 'Dados' em indicadores de navegação.

BOA PRÁTICA

Selecione no painel Seleção apenas os objetos envolvidos e marque 'Objetos selecionados' antes de salvar o indicador. Isso evita que ele mexa em elementos de outras partes da página.

Alternar entre gráfico e tabela

1. Posicione o gráfico e a tabela exatamente sobrepostos, do mesmo tamanho.

2. Crie o indicador 'Ver gráfico' com a tabela oculta; e 'Ver tabela' com o gráfico oculto.
3. Adicione dois botões de ícone (gráfico e tabela), cada um apontando para seu indicador.
4. Formate o botão do estado ativo com cor de destaque para dar feedback visual.

Painel de filtros retrátil

1. Crie um retângulo lateral com as segmentações dentro, e agrupe tudo (Ctrl+G) como 'Painel Filtros'.
2. Indicador 'Filtros abertos': grupo visível. Indicador 'Filtros fechados': grupo oculto.
3. Botão de ícone de funil fora do painel abre; um X dentro do painel fecha.
4. Ganho típico: 20% a mais de área útil para os visuais.

Navegador de páginas e de indicadores

Inserir → Botões → Navegador → Navegador de página cria um menu automático com todas as páginas visíveis, e ele se atualiza sozinho quando você adiciona ou renomeia uma página. Use sempre no lugar de botões manuais.

- Formate os estados Padrão, Ao passar o mouse, Pressionado e Selecionado — o feedback é o que faz parecer um aplicativo.
- Oculte páginas de detalhe e de tooltip para que não apareçam no navegador.
- Mantenha o menu na mesma posição em todas as páginas; copie e cole entre páginas.

EXERCÍCIO PRÁTICO

Monte um menu lateral com navegador de páginas, um painel de filtros retrátil e um alternador gráfico/tabela em uma das páginas. Teste todos os estados dos botões.

Tooltips de página e visuais especiais

Contexto sob demanda e os visuais que resolvem problemas específicos.

Tooltip de página

1. Nova página → Formatar página → Tamanho da tela → Dica de ferramenta (320x240).
2. Formatar página → Informações da página → Permitir uso como dica de ferramenta: Ativado.
3. Monte dentro dela um mini-painel: uma linha de tendência, dois cartões e um título dinâmico.
4. Oculte a página da barra de guias.
5. No visual de origem: Formatar → Dica de ferramenta → Tipo: Página do relatório → escolha a página.

BOA PRÁTICA

Em uma matriz por vendedor, passar o mouse e ver a evolução dos últimos 12 meses daquele vendedor substitui uma página inteira de análise.

Visuais nativos subutilizados

Visual	Para quê	Cuidado
Cascata (waterfall)	Explicar a variação entre dois períodos por contribuição	Limite o número de categorias; use 'Detalhamento'
Influenciadores-chave	Descobrir o que mais impacta um resultado	Precisa de volume de linhas para dar resultado confiável
Árvore hierárquica (decomposition tree)	Exploração livre de causa raiz	Defina os níveis permitidos para evitar bagunça
Gráfico de dispersão com Play Axis	Evolução de duas variáveis no tempo	Excelente em apresentação, pouco útil em impressão
Small multiples	Comparar a mesma métrica entre muitas categorias	Mantenha a escala do eixo compartilhada
Narrativa inteligente	Texto automático explicando o visual	Sempre revise o texto gerado antes de publicar

Visuais do AppSource

- Prefira visuais certificados: eles não fazem chamadas externas e funcionam em exportação para PDF e e-mail.
- Cada visual personalizado adiciona peso ao relatório e vira uma dependência de terceiro.
- Antes de instalar, pergunte: dá para resolver com visual nativo + formatação condicional? Na maioria das vezes, dá.
- Úteis de verdade: Bullet Chart, Sankey, Gantt (certificado) e Tabela com barras (Charticulator).

ATENÇÃO

Antes de basear um projeto inteiro em um visual da loja, confirme com o time de TI se a organização permite visuais não-certificados. Muitos ambientes bloqueiam por política.

EXERCÍCIO PRÁTICO

Crie uma página de tooltip com tendência de 12 meses e aplique-a a uma matriz. Depois, construa um gráfico de cascata explicando a variação de receita entre o ano atual e o anterior por categoria.

Quiz do módulo

1. Em um gráfico de tendência, a interação recomendada ao clicar em outra categoria é:

- Filtrar
- ✓ **Destacar**
- Não interagir
- Tanto faz

Destacar preserva o total como referência e mostra a parte selecionada.

2. O erro mais comum ao criar indicadores de navegação é:

- Não renomear o indicador

- ✓ **Deixar 'Dados' marcado, congelando os filtros do momento da criação**
- Usar botão em vez de imagem
- Criar mais de três indicadores

Indicadores de navegação devem controlar apenas exibição e objetos.

3. Para uma página funcionar como tooltip, é obrigatório:

- Usar tamanho 1280x720
- ✓ **Ativar 'Permitir uso como dica de ferramenta' nas informações da página**
- Criar um indicador
- Publicar o relatório antes

Sem essa opção, a página não aparece na lista de tooltips.

4. Drill-through com dois campos de detalhamento:

- Funciona com qualquer um dos dois
- ✓ **Só habilita quando o contexto fornece ambos**
- Não é permitido
- Ignora o segundo campo

Para opções independentes, crie páginas de detalhe separadas.

5. Vantagem do navegador de páginas sobre botões manuais:

- É mais bonito
- ✓ **Atualiza-se sozinho quando páginas são adicionadas ou renomeadas**
- Carrega mais rápido
- Permite mais cores

Elimina manutenção manual do menu.

6. Antes de instalar um visual do AppSource, a primeira pergunta deve ser:

- Ele é gratuito?
- ✓ **Dá para resolver com visual nativo e formatação condicional?**
- Tem boa avaliação?
- Quem é o autor?

Visual personalizado adiciona peso e dependência externa.

Formatação condicional, RLS e temas JSON

Formatação condicional por medida, segurança em nível de linha estática e dinâmica, e temas JSON que padronizam todos os relatórios da empresa.

Objetivos de aprendizagem

- Dominar formatação condicional em todos os pontos onde ela existe
- Implementar RLS estática e dinâmica com tabela de permissões
- Testar segurança corretamente antes de publicar
- Criar e distribuir um tema JSON corporativo

Formatação condicional avançada

Cores, ícones, barras e URLs controlados por medida.

Os cinco pontos de formatação condicional

Onde	O que faz	Uso típico
Cor do plano de fundo	Pinta a célula	Mapa de calor em matriz
Cor da fonte	Pinta o texto	Variação negativa em vermelho
Barras de dados	Barra dentro da célula	Comparação de magnitude sem gráfico extra
Ícones	Semáforo, setas	Status de meta
URL da Web	Transforma o texto em link	Ir para o ERP no registro específico

DAX

```
Cor Margem =
VAR M = [Margem %]
RETURN
SWITCH( TRUE(),
    ISBLANK( M ), "#3A3F47",
    M < 0, "#D0453B",
    M < 0.15, "#E0A526",
    M < 0.30, "#7FB77E",
    "#2E9E6B"
)
```

BOA PRÁTICA

Escala automática muda de significado conforme o filtro. Uma medida de cor com faixas fixas dá leitura estável: vermelho é sempre prejuízo, independentemente da seleção.

Título e URLs dinâmicos

DAX

Link do Pedido =

```
"https://erp.empresa.com/pedidos/" & SELECTEDVALUE( fVendas[NumeroPedido] )
```

-- Depois: Modelagem → Categoria de dados → URL da Web

-- Na tabela: Formatar → Valores → URL da Web → Ativado

DAX

Título Gráfico =

```
VAR Cat = CONCATENATEX( VALUES( dProduto[Categoria] ), dProduto[Categoria], ", " )
```

```
VAR Qtd = COUNTROWS( VALUES( dProduto[Categoria] ) )
```

```
RETURN
```

```
"Receita mensal • " &
```

```
IF( Qtd > 3, Qtd & " categorias selecionadas", Cat )
```

ATENÇÃO

Uma matriz com 5 mil células e 3 regras condicionais executa 15 mil avaliações de medida.

Reduza linhas com Top N ou avalie usar um visual mais leve.

EXERCÍCIO PRÁTICO

Crie uma matriz por produto com cor de fundo por medida, barras de dados na receita, ícone de status e título dinâmico. Verifique o tempo de renderização no Analisador de Desempenho.

Segurança em nível de linha (RLS)

Estática, dinâmica e as armadilhas que causam vazamento de dados.

RLS estática

1. Modelagem → Gerenciar funções → Criar.
2. Nomeie a função (ex.: 'Região Sul') e selecione a tabela dRegiao.
3. Expressão DAX do filtro: [Regiao] = "Sul".
4. Salve e teste em Exibir como.
5. Após publicar, no Serviço: Conjunto de dados → Segurança → adicione os usuários à função.

ATENÇÃO

Com 40 regiões você teria 40 funções para manter. Use estática apenas para 2 ou 3 recortes estáveis; para o resto, use dinâmica.

RLS dinâmica

Uma única função filtra conforme o usuário conectado, usando uma tabela de permissões. É a abordagem correta para qualquer cenário real.

DAX

```
-- Tabela dPermissao: Email | Regiao
-- Na tabela dPermissao, expressão da função:
[Email] = USERPRINCIPALNAME()

-- A relação dPermissao[Regiao] → dRegiao[Regiao] propaga para a fato.
-- Se não houver relação física, use o padrão abaixo em dRegiao:
[Regiao] IN
    CALCULATETABLE(
        VALUES( dPermissao[Regiao] ),
        dPermissao[Email] = USERPRINCIPALNAME()
    )
```

BOA PRÁTICA

No Desktop, USERNAME() devolve DOMINIO\usuario; no Serviço, devolve o e-mail. USERPRINCIPALNAME() devolve o e-mail nos dois. Use sempre USERPRINCIPALNAME().

Hierarquia de gestão

DAX

```
-- Coluna calculada em dVendedor
Caminho = PATH( dVendedor[IdVendedor], dVendedor[IdGestor] )

-- Filtro da função RLS em dVendedor:
PATHCONTAINS(
    dVendedor[Caminho],
    LOOKUPVALUE(
        dVendedor[IdVendedor],
        dVendedor[Email], USERPRINCIPALNAME()
    )
)
```

Testando corretamente

1. No Desktop: Modelagem → Exibir como → marque a função e informe 'Outro usuário' com um e-mail real.
2. Percorra TODAS as páginas do relatório, inclusive tooltips e páginas de detalhe.
3. Confira os totais: se o total geral não mudou, a RLS não está filtrando.
4. No Serviço: Segurança → três pontos da função → Testar como função.
5. Peça a validação de um usuário real de cada perfil antes do go-live.

ATENÇÃO

1) Filtro bidirecional pode contornar a RLS — ative 'Aplicar filtro de segurança nos dois sentidos' quando necessário. 2) Membros com função de Administrador ou Membro no workspace ignoram RLS. 3) RLS não protege o download do arquivo .pbix. 4) Medidas com ALL/REMOVEFILTERS não removem RLS, mas removem o filtro visual — verifique o comportamento esperado.

Segurança em nível de objeto (OLS) esconde tabelas e colunas inteiras, mas exige ferramenta externa (Tabular Editor). Use quando a existência da coluna já é informação sensível, como salário.

EXERCÍCIO PRÁTICO

Crie a tabela dPermissao, implemente RLS dinâmica e teste com três perfis diferentes. Depois implemente a variação com hierarquia de gestão usando PATH e PATHCONTAINS.

Temas JSON corporativos

Padronizar cores, fontes e o estilo de cada tipo de visual em todos os relatórios.

Um tema JSON define a identidade visual de todos os relatórios de uma empresa. Sem ele, cada analista escolhe cores diferentes e o portfólio vira uma colcha de retalhos.

JSON

```
{
  "name": "Tema Corporativo 2026",
  "dataColors": [
    "#E0A526", "#3FA7C4", "#7FB77E", "#C4785A",
    "#8C7BC4", "#D0453B", "#4C8FA8", "#A8A29E"
  ],
  "background": "#14161A",
  "foreground": "#E8EAED",
  "tableAccent": "#E0A526",
  "good": "#2E9E6B",
  "neutral": "#E0A526",
  "bad": "#D0453B",
  "maximum": "#2E9E6B",
  "minimum": "#D0453B",
  "textClasses": {
    "title": { "fontFace": "Segoe UI Semibold", "fontSize": 16, "color": "#E8EAED" },
    "header": { "fontFace": "Segoe UI Semibold", "fontSize": 12, "color": "#E8EAED" },
    "label": { "fontFace": "Segoe UI", "fontSize": 10, "color": "#A8ADB5" },
    "callout": { "fontFace": "Segoe UI Light", "fontSize": 32, "color": "#E8EAED" }
  },
  "visualStyles": {
    "*": {
      "*": {
        "background": [{ "show": true, "color": { "solid": { "color": "#1B1E24" } } }, "transparenc
cy": 0 }],
        "border": [{ "show": true, "color": { "solid": { "color": "#2A2E36" } } }, "radius": 8
      ]},
        "visualHeader": [{ "show": false }],
        "title": [{ "show": true, "fontSize": 13, "fontColor": { "solid": { "color": "#E8EAED" }
      } } ]
    }
  },
  "page": {
    "*": {
      "background": [{ "color": { "solid": { "color": "#14161A" } } }, "transparency": 0 }],
      "outspace": [{ "color": { "solid": { "color": "#0F1115" } } } ]
    }
  },
  "card": {
    "*": {
      "labels": [{ "fontSize": 28, "color": { "solid": { "color": "#E8EAED" } } } ],
      "categoryLabels": [{ "fontSize": 10, "color": { "solid": { "color": "#A8ADB5" } } } ]
    }
  },
  "lineChart": {
    "*": {
      "lineStyles": [{ "strokeWidth": 3, "lineStyle": "solid" }],
      "categoryAxis": [{ "gridlineShow": false }],
      "valueAxis": [{ "gridlineColor": { "solid": { "color": "#2A2E36" } } }, "gridlineStyle": "
dotted" } ]
    }
  }
}
```

Como aplicar e distribuir

1. Salve o arquivo como tema-corporativo.json.

2. Exibição → Temas → Procurar temas → selecione o arquivo.
3. Teste em um relatório com todos os tipos de visual usados na empresa.
4. Versione o JSON em um repositório compartilhado (SharePoint, Git).
5. Crie um arquivo .pbix modelo com o tema já aplicado, páginas padrão e medidas base.

BOA PRÁTICA

A primeira cor é a que aparece em gráficos de série única. Coloque na posição 1 a cor de destaque da marca e deixe os tons neutros para o final.

ATENÇÃO

Se alguém pintou um visual manualmente, o tema não altera aquele visual. Use Formatar → Redefinir ao padrão para devolver o controle ao tema.

- Mantenha um tema claro e um escuro com a mesma estrutura; troque conforme o público.
- Documente no próprio JSON, com o campo 'name', a versão e a data.
- Ferramentas como o Theme Generator ajudam, mas sempre revise o JSON final à mão.

EXERCÍCIO PRÁTICO

Crie o tema JSON da sua empresa com paleta, tipografia e estilo de cartão, linha e tabela. Aplique em dois relatórios diferentes e verifique a consistência.

Quiz do módulo

1. Por que formatar por 'Valor do campo' é melhor que escala de cor automática?

- É mais rápido
- ✓ **As faixas ficam fixas e o significado da cor não muda conforme o filtro**
- Permite mais cores
- Funciona em mais visuais

Escala automática recalibra conforme a seleção e confunde a leitura.

2. A função correta para identificar o usuário tanto no Desktop quanto no Serviço é:

- USERNAME()
- ✓ **USERPRINCIPALNAME()**
- CUSTOMDATA()
- USEROBJECTID()

USERNAME() devolve DOMINIO\usuario no Desktop.

3. Um usuário com papel de Administrador no workspace:

- Também é filtrado pela RLS
- ✓ **Ignora a RLS e vê todos os dados**
- Não consegue abrir o relatório
- Vê apenas o resumo

Admin e Membro do workspace não são filtrados por RLS.

4. Para um gestor ver a si e a toda a sua cadeia de subordinados, use:

- FILTER com ALL
- ✓ **PATH e PATHCONTAINS**
- TREATAS
- USERRELATIONSHIP

PATH monta a hierarquia e PATHCONTAINS testa a pertinência.

5. Se um visual foi formatado manualmente e o tema é trocado:

- O tema sobrepõe tudo
- ✓ **A formatação manual prevalece; é preciso redefinir ao padrão**
- O visual quebra
- O tema não carrega

Formatação explícita tem precedência sobre o tema.

6. Para esconder completamente uma coluna sensível como salário, o recurso é:

- RLS
- ✓ **OLS (segurança em nível de objeto)**
- Ocultar no modelo
- Formatação condicional

Ocultar no modelo é cosmético; OLS impede o acesso de fato.

Performance e boas práticas

Analizador de Desempenho, DAX Studio, redução de cardinalidade, agregações e o conjunto de regras que separa um modelo amador de um profissional.

Objetivos de aprendizagem

- Medir onde o tempo é gasto, em vez de adivinhar
- Reduzir o tamanho do modelo com decisões de coluna e cardinalidade
- Reescrever DAX lento com padrões comprovados
- Aplicar um checklist de qualidade antes de publicar

Medindo antes de otimizar

Analizador de Desempenho, DAX Studio e a leitura de FE x SE.

ATENÇÃO

Na maioria dos relatórios lentos, 80% do tempo está em 1 ou 2 visuais. Otimizar os outros 20 não muda nada para o usuário.

Analizador de Desempenho

1. Exibição → Analizador de desempenho → Iniciar gravação.
2. Atualizar visuais (ou interaja normalmente com a página).
3. Ordene por Duração e olhe os três primeiros.
4. Cada linha se decompõe em: Consulta DAX, Exibição de visual, Outro.
5. Copie a consulta com 'Copiar consulta' para analisar no DAX Studio.

Componente lento	Significa	Caminho de correção
Consulta DAX	A medida ou o modelo é o gargalo	Reescreva a medida, revise relacionamentos e cardinalidade
Exibição de visual	O visual tem linhas ou elementos demais	Reduza pontos, use Top N, troque o tipo de visual
Outro	Espera por outros visuais e sincronização	Reduza a quantidade de visuais na página

DAX Studio: Server Timings

- SE (Storage Engine): rápido, multithread, faz varreduras e agregações simples. Quanto mais tempo aqui, melhor.
- FE (Formula Engine): single-thread, executa a lógica complexa. Tempo alto em FE indica medida mal escrita.
- Regra prática: se FE passa de 30% do total, há o que reescrever.
- Olhe também 'SE Queries': dezenas de consultas ao SE indicam materialização excessiva.

DAX

```
-- LENTO: IF linha a linha dentro do iterador sobre a fato
Receita Promo Lenta =
SUMX(
    fVendas,
    IF( RELATED( dProduto[Promocao] ) = "Sim",
        fVendas[Quantidade] * fVendas[PrecoUnitario], 0 )
)

-- RÁPIDO: o filtro vira trabalho do Storage Engine
Receita Promo =
CALCULATE( [Receita Total], dProduto[Promocao] = "Sim" )
```

BOA PRÁTICA

No DAX Studio, Advanced → View Metrics mostra o tamanho de cada coluna. Quase sempre as 3 maiores colunas respondem por metade do arquivo — e costumam ser desnecessárias.

EXERCÍCIO PRÁTICO

Grave o Analisador de Desempenho em um relatório real, identifique os três visuais mais lentos e classifique cada um como problema de DAX, de visual ou de quantidade de visuais.

Reduzindo o tamanho do modelo

Cardinalidade, colunas desnecessárias, tipos de dado e agregações.

O VertiPaq comprime por coluna. O que determina o tamanho não é o número de linhas, e sim a cardinalidade: quantos valores distintos a coluna tem.

Ação	Ganho típico	Custo
Remover colunas não usadas	20% a 60%	Nenhum, se realmente não são usadas
Separar data e hora em duas colunas	Alto em fatos grandes	Perde o carimbo exato combinado
Remover colunas de ID textual longo	Alto	Precisa manter uma chave numérica
Trocar decimal por moeda (4 casas fixas)	Médio	Arredondamento em 4 casas
Resumir a fato na granularidade necessária	Muito alto	Perde o detalhe linha a linha
Desativar 'Data/hora automática'	Médio	Exige tabela calendário própria (que você já deve ter)

ATENÇÃO

Uma coluna com data e hora até o segundo tem milhões de valores distintos. Separe em Data (baixa cardinalidade) e Hora (no máximo 1.440 valores por minuto). O ganho chega a 40% do arquivo.

M

let

```
Origem = Sql.Database("servidor", "vendas"),
Fato = Origem{[Schema="dbo", Item="fVendas"]}[Data],
ComData = Table.AddColumn(Fato, "Data", each DateTime.Date([DataHora]), type date),
ComHora = Table.AddColumn(ComData, "Hora",
    each Time.From(Text.Combine({
        Text.PadStart(Text.From(Time.Hour([DataHora])), 2, "0"), ":",
        Text.PadStart(Text.From(Time.Minute([DataHora])), 2, "0"), ":00"
    })), type time),
SemOriginal = Table.RemoveColumns(ComHora, {"DataHora"})
in
SemOriginal
```

Desativar a data/hora automática

1. Arquivo → Opções → Configurações atuais do arquivo → Inteligência de dados.
2. Desmarque 'Data/hora automática'.
3. Faça o mesmo em Configurações globais para novos arquivos.
4. Motivo: o Power BI cria uma tabela de calendário oculta para CADA coluna de data do modelo.

Agregações

Em modelos muito grandes, crie uma tabela agregada (por exemplo, por dia + produto) em modo Importação sobre uma fato detalhada em DirectQuery. O Power BI usa a agregação quando ela responde à consulta e só desce ao detalhe quando precisa.

1. Crie a tabela agregada na fonte ou no Power Query, com as chaves das dimensões.
2. Relacione-a às mesmas dimensões da fato detalhada.
3. Clique com o botão direito na tabela → Gerenciar agregações.
4. Mapeie cada coluna: Contagem, Soma, Mín, Máx, Agrupar por.
5. Oculte a tabela agregada — o usuário nunca deve vê-la.

BOA PRÁTICA

Colunas mantidas 'por via das dúvidas' são a principal causa de modelos inchados. Remova. Se precisar, você adiciona em dois minutos no Power Query.

EXERCÍCIO PRÁTICO

Analise seu modelo com o VertiPaq Analyzer, liste as 5 maiores colunas e elimine ou reduza pelo menos três. Registre o tamanho do arquivo antes e depois.

Checklist profissional antes de publicar

As regras que evitam 90% dos problemas em produção.

Modelo

- Esquema estrela; sem relacionamentos entre duas tabelas fato.
- Tabela calendário marcada como tabela de datas, contínua e completa.

- Nenhum relacionamento bidirecional sem justificativa documentada.
- Colunas de chave ocultas; colunas não usadas removidas.
- Nomes em linguagem de negócio: 'Receita Líquida', não 'vl_rec_liq'.
- Medidas organizadas em uma tabela dedicada '_Medidas' com pastas de exibição.
- Formatação definida na medida, não no visual.
- Data/hora automática desativada.

DAX

- DIVIDE em toda divisão, nunca o operador /.
- Variáveis para qualquer expressão usada mais de uma vez.
- CALCULATE com filtro de coluna em vez de FILTER sobre tabela inteira, sempre que possível.
- Sem colunas calculadas que poderiam ser medidas.
- Sem IF dentro de iterador sobre a fato quando um CALCULATE resolve.
- Comentários explicando a regra de negócio, não a sintaxe.

Relatório

- Máximo de 6 visuais por página relevante.
- Tema aplicado; nenhuma cor escolhida manualmente sem motivo.
- Texto alternativo preenchido em todos os visuais.
- Ordem de tabulação definida.
- Interações revisadas visual a visual.
- Páginas de tooltip e de detalhe ocultas.
- Carimbo de atualização e fonte visíveis no rodapé.

Publicação

- RLS testada com pelo menos um usuário real de cada perfil.
- Credenciais da fonte configuradas no Serviço.
- Agendamento de atualização definido, com notificação de falha para uma lista, não para uma pessoa.
- Workspace separado para Desenvolvimento, Homologação e Produção.
- Arquivo .pbix versionado (OneDrive com histórico, SharePoint ou Git com PBIP).

BOA PRÁTICA

O Tabular Editor traz o Best Practice Analyzer, com dezenas de regras da comunidade. Rode antes de cada publicação: ele encontra em segundos o que levaria uma revisão manual de uma hora.

ATENÇÃO

Publicar sem documentar de onde vem cada número. Seis meses depois, ninguém — nem você — lembra por que a receita exclui devoluções. Mantenha uma página 'Dicionário de indicadores' dentro do próprio relatório.

Formato PBIP e versionamento

Salve como Projeto do Power BI (.pbip): o modelo e o relatório viram arquivos de texto (TMDL e JSON), versionáveis em Git com diff legível, revisão por pull request e CI/CD. É o padrão para times.

EXERCÍCIO PRÁTICO

Rode o checklist completo em um relatório seu e registre cada item não conforme. Corrija os cinco mais críticos e documente os demais como dívida técnica.

Quiz do módulo

1. No Analisador de Desempenho, tempo alto em 'Exibição de visual' indica:

- Medida mal escrita
- **✓ Visual com pontos ou elementos demais para renderizar**
- Problema de rede
- Falta de RLS

O DAX já respondeu; o custo está na renderização.

2. Se o Formula Engine consome mais de 30% do tempo da consulta:

- É normal e esperado
- **✓ A medida provavelmente precisa ser reescrita**
- O servidor está lento
- Falta memória

FE é single-thread; o ideal é empurrar trabalho para o Storage Engine.

3. O que mais determina o tamanho de uma coluna no VertiPaq?

- Número de linhas
- **✓ Cardinalidade (valores distintos)**
- Tipo de dado apenas
- Nome da coluna

A compressão por dicionário depende dos valores distintos.

4. Separar DataHora em Data e Hora:

- Não muda nada
- **✓ Reduz muito o tamanho, porque diminui drasticamente a cardinalidade**
- Aumenta o modelo
- Só funciona em DirectQuery

Uma coluna de data e hora ao segundo tem milhões de valores distintos.

5. Por que desativar 'Data/hora automática'?

- Porque desativa a inteligência temporal
- **✓ Porque o Power BI cria uma tabela de calendário oculta para cada coluna de data**

- Porque impede o uso de filtros
- Porque quebra o DirectQuery

São tabelas ocultas que inflam o modelo sem benefício quando existe dCalendario.

6. A principal vantagem do formato .pbip é:

- Abrir mais rápido
- **✓ Modelo e relatório viram texto, permitindo versionamento em Git com diff legível**
- Ocupar menos espaço
- Dispensar o Power BI Desktop

Habilita revisão por pull request e CI/CD para times.

Publicação, colaboração e governança

Workspaces, aplicativos, atualização agendada, gateway, fluxos de dados, ciclo de vida de implantação e a estrutura de um Centro de Excelência de BI.

Objetivos de aprendizagem

- Publicar e distribuir relatórios com o modelo correto de permissão
- Configurar atualização, gateway e alertas de falha
- Estruturar Dev, Homologação e Produção com pipelines
- Definir regras mínimas de governança para escalar

Workspaces, aplicativos e permissões

Onde o relatório mora e como as pessoas certas acessam.

Papéis no workspace

Papel	Pode	Usar para
Administrador	Tudo, inclusive excluir o workspace e gerenciar acessos	1 ou 2 responsáveis
Membro	Publicar, editar, compartilhar e publicar o app	Time de desenvolvimento
Colaborador	Publicar e editar, mas não gerencia acessos nem publica o app	Analistas
Visualizador	Apenas ver o conteúdo; é filtrado pela RLS	Usuários finais

ATENÇÃO

Se o usuário precisa ser filtrado por segurança em nível de linha, ele tem que ser Visualizador — ou, melhor ainda, acessar pelo aplicativo e nunca pelo workspace.

Aplicativos (Apps)

- O app é a embalagem para o consumidor: você escolhe quais relatórios entram e em que ordem.
- Permite navegação com seções, imagem de capa e descrição.
- Alterações no workspace só chegam ao usuário quando você clica em 'Atualizar app' — isso é uma vantagem, não um obstáculo.
- Nunca dê acesso direto ao workspace para o usuário final: distribua pelo app.

Modelos semânticos compartilhados

Publique o modelo uma vez e conecte vários relatórios a ele (Obter dados → Modelos semânticos do Power BI). Uma definição de receita, uma atualização, muitos relatórios. É a base de qualquer operação de BI madura.

1. Publique o arquivo que contém apenas o modelo (sem páginas, ou com uma página de documentação).
2. Marque o modelo como 'Promovido' ou peça a 'Certificação' ao time de governança.
3. Conceda a permissão 'Build' aos analistas que criarão relatórios.
4. Cada relatório novo se conecta em modo de conexão dinâmica (Live Connection).

BOA PRÁTICA

Modelos Promovidos e Certificados aparecem primeiro na busca e ganham selo. É o mecanismo mais simples para combater a proliferação de versões não-oficiais da mesma informação.

EXERCÍCIO PRÁTICO

Estruture um workspace com papéis definidos, publique um modelo compartilhado e crie um relatório em conexão dinâmica sobre ele. Depois publique o app para um grupo de usuários.

Atualização, gateway e fluxos de dados

Manter os dados frescos sem depender de alguém clicar em atualizar.

Gateway de dados

O gateway é o ponte entre a nuvem e as fontes locais (SQL Server on-premises, arquivos em rede, ERPs internos). Sem ele, o Serviço não consegue atualizar esses dados.

- Modo padrão (standard): instalado em um servidor, compartilhado pela organização. É o correto para produção.
- Modo pessoal: só funciona para o seu usuário, não é compartilhável. Use apenas para testes.
- Instale em uma máquina que fique sempre ligada, próxima da fonte de dados.
- Configure um cluster com dois gateways para alta disponibilidade.
- Fontes exclusivamente em nuvem (SharePoint Online, Dataverse, bancos em nuvem) geralmente dispensam gateway.

Atualização agendada

1. No Serviço: Modelo semântico → Configurações → Credenciais da fonte de dados → Editar credenciais.
2. Atualização agendada → Ativar → defina fuso horário e horários.
3. Free e Pro: até 8 atualizações diárias. PPU e Capacidade: até 48.
4. Ative 'Enviar notificação de falha' e aponte para um grupo de e-mail, nunca para uma pessoa só.
5. Verifique o histórico de atualização semanalmente nos primeiros meses.

ATENÇÃO

Se o ETL da origem termina às 7h50 e atrasa um dia, seu relatório mostra dado velho sem avisar ninguém. Exiba sempre a data do último registro (não a hora do refresh) no rodapé e crie um alerta quando ela ficar atrasada.

Atualização incremental

Em vez de recarregar 5 anos de histórico todo dia, carregue só os últimos dias. Configure com os parâmetros RangeStart e RangeEnd no Power Query e a política de incremento no modelo. Reduz janelas de atualização de horas para minutos.

Fluxos de dados (Dataflows)

- É o Power Query executado na nuvem, salvando o resultado em armazenamento gerenciado.
- Use quando várias pessoas repetem a mesma transformação: uma dCalendario, um cadastro de produto tratado.
- Centraliza a lógica de ETL e evita que cada analista limpe os dados do seu jeito.
- O relatório passa a consumir o fluxo, e a atualização vira: fluxo primeiro, modelo depois.

BOA PRÁTICA

Fonte → Fluxo de dados (limpeza padronizada) → Modelo semântico compartilhado (relacionamentos e medidas) → Relatórios (visualização). Cada camada tem um dono claro e uma responsabilidade.

EXERCÍCIO PRÁTICO

Configure a atualização agendada de um modelo com notificação de falha para um grupo, e crie um fluxo de dados com a tabela calendário para reuso em outros projetos.

Ciclo de vida, pipelines e governança

Do desenvolvimento à produção sem quebrar o relatório do diretor.

Pipelines de implantação

Pipeline conecta três workspaces — Desenvolvimento, Teste e Produção — e promove conteúdo entre eles com um clique, preservando as configurações de cada estágio.

1. Crie três workspaces: Vendas [Dev], Vendas [Teste], Vendas [Prod].
2. Crie o pipeline e atribua cada workspace ao seu estágio.
3. Configure regras de parâmetro para trocar a fonte de dados por estágio (base de teste x base de produção).
4. Desenvolva sempre no Dev; promova para Teste; valide com usuários-chave; promova para Prod.
5. Publique o app apenas a partir do workspace de Produção.

ATENÇÃO

Uma medida alterada às pressas na sexta-feira à tarde vira um número errado na reunião de segunda. Pipeline não é burocracia: é o que protege a credibilidade da área.

Governança mínima viável

Área	Regra mínima	Quem responde
Nomenclatura	Padrão para workspaces, relatórios e medidas	Time de BI
Endosso	Só modelos certificados podem embasar decisão oficial	Governança de dados
Acesso	Usuário final sempre via app; nunca no workspace	Admin do workspace
Documentação	Dicionário de indicadores dentro do relatório	Autor do relatório
Ciclo de vida	Nada vai a produção sem passar por teste	Time de BI
Monitoramento	Revisão mensal de uso e de falhas de atualização	Centro de Excelência

Monitoramento

- Hub de monitoramento: veja atualizações em andamento e falhas recentes.
- Métricas de uso por relatório: quem abre, com que frequência, em qual dispositivo.
- Relatórios sem acesso há 90 dias são candidatos a arquivamento — reduzem custo e confusão.
- Portal de administração: log de auditoria, configurações de locatário, exportação e compartilhamento externo.

Centro de Excelência (CoE)

- Define padrões: tema, nomenclatura, arquitetura em camadas.
- Mantém os ativos compartilhados: fluxos de dados, modelos certificados, arquivo modelo.
- Capacita: treinamento interno, revisão de código DAX, comunidade de prática.
- Mede: adoção, qualidade, tempo de entrega, incidentes.

BOA PRÁTICA

Não espere um CoE formal para começar a governar. Três regras escritas e seguidas valem mais do que um manual de 60 páginas que ninguém lê. Escolha nomenclatura, ciclo de vida e endosso — e cumpra.

Com isso você encerra a trilha técnica: sabe extrair, transformar, modelar, calcular, visualizar, publicar e governar. Os estudos de caso a seguir aplicam tudo isso em sete áreas de negócio reais.

EXERCÍCIO PRÁTICO

Desenhe a estrutura de workspaces e o pipeline para um projeto seu, escreva as três regras de governança que você conseguiria fazer cumprir na prática, e defina quem responde por cada uma.

Quiz do módulo

1. Para que um usuário seja filtrado pela RLS, ele deve ter o papel de:

- Administrador
- Membro
- Colaborador
- **✓ Visualizador**

Admin, Membro e Colaborador não são filtrados pela RLS.

2. A principal vantagem de distribuir por aplicativo em vez de dar acesso ao workspace é:

- Carrega mais rápido
- **✓ Você controla o que e quando o usuário recebe, publicando a atualização quando quiser**
- Reduz o custo de licença
- Dispensa RLS

O app desacopla o desenvolvimento do consumo.

3. Um modelo semântico compartilhado serve para:

- Reduzir o tamanho do arquivo
- **✓ Ter uma definição única de indicador alimentando vários relatórios**
- Permitir mais atualizações diárias
- Substituir o gateway

Uma verdade, uma atualização, muitos relatórios.

4. Gateway em modo pessoal:

- É o recomendado para produção
- **✓ Funciona apenas para o seu usuário e não é compartilhável**
- Dispensa instalação
- Só funciona com Excel

Produção exige o gateway em modo padrão, idealmente em cluster.

5. Um fluxo de dados (dataflow) é indicado quando:

- O relatório está lento
- **✓ Várias pessoas repetem a mesma transformação e ela deve ser centralizada**
- É preciso aplicar RLS
- O modelo tem muitas medidas

Ele padroniza a camada de ETL na nuvem e permite reuso.

6. A função do pipeline de implantação é:

- Acelerar a atualização
- **✓ Promover conteúdo entre Dev, Teste e Produção preservando as configurações de cada estágio**
- Criar backups
- Converter relatórios em PDF

Evita edição direta em produção, a maior causa de incidentes.

Estudos de caso reais

Sete projetos completos: contexto, perguntas de negócio, indicadores com metas, modelo de dados, medidas DAX e layout do dashboard. Reproduza cada um do zero para consolidar o aprendizado.

Painel Comercial de uma rede varejista com 42 lojas

Contexto

Rede de varejo alimentar com 42 lojas em 6 estados, 18 mil SKUs e 210 vendedores. A diretoria comercial recebia hoje três planilhas diferentes por e-mail toda segunda-feira, com números que não fechavam entre si. O objetivo é uma fonte única, atualizada diariamente, que responda em 10 segundos se o mês está no ritmo da meta e onde está o problema.

Perguntas que o painel responde

- Estamos no ritmo da meta do mês, por loja e por vendedor?
- O crescimento vem de mais clientes, mais itens ou preço maior?
- Quais produtos estão puxando a margem para baixo?
- Quais lojas caíram em relação ao mesmo período do ano passado, com dias comparáveis?
- Quais clientes de atacado reduziram compra nos últimos 90 dias?

Indicadores e metas

Indicador	Meta	Por que importa
Receita Líquida	R\$ 48 mi/mês	Indicador-mestre; exclui devoluções e impostos sobre venda
Atingimento Proporcional	$\geq 100\%$	Mede o ritmo dentro do mês, sem falso alarme no dia 5
Margem Bruta %	$\geq 24\%$	Crescer vendendo com prejuízo não é crescimento
Ticket Médio	R\$ 87	Mostra se o crescimento vem de mais clientes ou de compras maiores
Itens por Cupom	$\geq 5,2$	Mede a eficácia de cross-sell e da exposição em loja
Ruptura de Gôndola %	$\leq 3\%$	Venda perdida não aparece no faturamento — só aqui
Curva ABC de Produtos	$A \leq 20\%$ dos SKUs	Concentra atenção no que realmente gera resultado

Modelo de dados

Esquema estrela clássico: uma fato de vendas na granularidade item do cupom, e dimensões de calendário, loja, produto, vendedor e cliente. A tabela de metas fica separada, em granularidade mês × loja × vendedor.

Tabela	Tipo	Granularidade	Volume
fVendas	Fato	Item do cupom	~ 62 mi linhas/ano
fMetas	Fato	Mês × Loja × Vendedor	~ 2,5 mil linhas/mês
fEstoque	Fato	Dia × Loja × SKU	~ 9 mi linhas/mês
dCalendario	Dimensão	Dia	1.826 linhas (5 anos)
dLoja	Dimensão	Loja	42
dProduto	Dimensão	SKU	18.000
dVendedor	Dimensão	Vendedor	210
dCliente	Dimensão	Cliente (atacado)	3.400

ATENÇÃO

fVendas e fEstoque têm granularidades diferentes e NÃO devem ser relacionadas entre si. Ambas se conectam às mesmas dimensões (dCalendario, dLoja, dProduto), e a comparação acontece na medida — este é o padrão de múltiplas fatos.

- Data e hora do cupom separadas em duas colunas (reduz ~35% do modelo).
- Coluna CustoUnitario trazida para fVendas na carga (custo vigente na data da venda).
- IDs textuais convertidos em chaves inteiras no Power Query.
- Atualização incremental: últimos 7 dias recarregados diariamente, histórico congelado.

Medidas DAX do projeto

Receita Líquida

Nasce linha a linha porque o desconto é percentual por item; devoluções entram como medida separada para poder ser analisada sozinha.

DAX

```
Receita Líquida =
SUMX(
    fVendas,
    fVendas[Quantidade] * fVendas[PrecoUnitario] * ( 1 - fVendas[PercDesconto] )
) - [Devoluções]
```

Margem Bruta %

Usa o custo gravado na venda, não o custo atual do cadastro — isso evita reescrever o histórico a cada mudança de preço de compra.

DAX

```
Margem Bruta =
SUMX(
    fVendas,
    fVendas[Quantidade] * ( fVendas[PrecoUnitario] * ( 1 - fVendas[PercDesconto] ) - fVendas[CustoUnitario] )
)

Margem Bruta % = DIVIDE( [Margem Bruta], [Receita Líquida] )
```

Ticket Médio e Itens por Cupom

DISTINCTCOUNT sobre cupom é a operação mais cara do painel; em modelos muito grandes, substitua por uma coluna de cupom já deduplicada em uma tabela auxiliar.

DAX

```
Qtd Cupons = DISTINCTCOUNT( fVendas[NumeroCupom] )

Ticket Médio = DIVIDE( [Receita Líquida], [Qtd Cupons] )

Itens por Cupom = DIVIDE( SUM( fVendas[Quantidade] ), [Qtd Cupons] )
```

Comparação justa com o ano anterior

A medida comparável é a que evita a reunião mensal mais desgastante do varejo: explicar por que o mês 'caiu 60%' no dia 12.

DAX

```
Receita LY =
CALCULATE( [Receita Líquida], SAMEPERIODYLASTYEAR( dCalendario[Date] ) )

Receita LY Comparável =
-- Só os mesmos dias já decorridos, para não comparar 12 dias com 30
VAR UltimoDiaComVenda =
    CALCULATE( MAX( fVendas[Data] ), REMOVEFILTERS( dCalendario ) )
VAR DiaCorte = MIN( MAX( dCalendario[Date] ), UltimoDiaComVenda )
RETURN
CALCULATE(
    [Receita Líquida],
    SAMEPERIODYLASTYEAR(
        FILTER( VALUES( dCalendario[Date] ), dCalendario[Date] <= DiaCorte )
    )
)

Var. YoY % = DIVIDE( [Receita Líquida] - [Receita LY Comparável], [Receita LY Comparável] )
```

Meta proporcional e semáforo

TREATAS conecta metas sem relacionamento físico, evitando ambiguidade no modelo quando a meta existe em granularidades diferentes.

DAX

```
Meta =
CALCULATE(
    SUM( fMetas[ValorMeta] ),
    TREATAS( VALUES( dCalendario[AnoMes] ), fMetas[AnoMes] ),
    TREATAS( VALUES( dLoja[Loja] ), fMetas[Loja] )
)

Meta Proporcional =
VAR DiasDecorridos =
    CALCULATE( COUNTROWS( dCalendario ), dCalendario[Date] <= TODAY() )
VAR DiasTotais = COUNTROWS( dCalendario )
RETURN [Meta] * DIVIDE( DiasDecorridos, DiasTotais )

Atingimento % = DIVIDE( [Receita Líquida], [Meta Proporcional] )

Cor Atingimento =
SWITCH( TRUE(),
    ISBLANK( [Meta] ), "#8A8F98",
    [Atingimento %] >= 1, "#2E9E6B",
    [Atingimento %] >= 0.9, "#E0A526",
    "#D0453B"
)
```

Ruptura de gôndola

Ruptura é o indicador que mais impacta o resultado e menos aparece nos painéis: a venda que não aconteceu não está em nenhuma tabela de faturamento.

DAX

```
SKUs Ativos por Loja =
CALCULATE( DISTINCTCOUNT( fEstoque[IdProduto] ), REMOVEFILTERS( fEstoque[Saldo] ) )

SKUs em Ruptura =
CALCULATE(
    DISTINCTCOUNT( fEstoque[IdProduto] ),
    fEstoque[Saldo] <= 0,
    dProduto[CurvaABC] IN { "A", "B" }
)

Ruptura % = DIVIDE( [SKUs em Ruptura], [SKUs Ativos por Loja] )

Venda Perdida Estimada =
SUMX(
    FILTER( SUMMARIZE( fEstoque, dProduto[Produto], dLoja[Loja] ), [SKUs em Ruptura] > 0 ),
    [Venda Média Diária do SKU]
)
```

Curva ABC dinâmica

Classe calculada por medida respeita o filtro do usuário: a curva ABC de uma loja específica é diferente da curva da rede.

DAX

```
Receita Acumulada % =  
VAR Base =  
    ADDCOLUMNS( ALLSELECTED( dProduto[Produto] ), "@Rec", [Receita Líquida] )  
VAR RecAtual = [Receita Líquida]  
VAR Acumulado = SUMX( FILTER( Base, [@Rec] >= RecAtual ), [@Rec] )  
RETURN DIVIDE( Acumulado, SUMX( Base, [@Rec] ) )  
  
Classe ABC Dinâmica =  
SWITCH( TRUE(),  
    [Receita Acumulada %] <= 0.8, "A",  
    [Receita Acumulada %] <= 0.95, "B",  
    "C"  
)
```

Clientes de atacado em risco

Queda de 30% em 90 dias é o gatilho combinado com o time comercial — ajuste o limiar ao seu ciclo de compra.

DAX

```
Clientes em Queda =  
VAR Base =  
    ADDCOLUMNS(  
        VALUES( dCliente[Cliente] ),  
        "@Atual", CALCULATE( [Receita Líquida], DATESINPERIOD( dCalendario[Date], TODAY(), -90, DAY ) ),  
        "@Anterior", CALCULATE( [Receita Líquida], DATESINPERIOD( dCalendario[Date], TODAY() - 90, -90, DAY ) )  
    )  
RETURN  
COUNTROWS(  
    FILTER( Base, [@Anterior] > 0 && DIVIDE( [@Atual], [@Anterior] ) < 0.7 )  
)
```

Layout do dashboard

- Faixa superior: título, seletor de período, filtros ativos em texto dinâmico e carimbo de atualização.
- Linha de KPIs (6 cartões): Receita Líquida, Atingimento Proporcional, Margem %, Ticket Médio, Itens/Cupom, Ruptura % — todos com variação YoY e cor semântica.
- Bloco central esquerdo: linha de receita diária acumulada do mês contra a linha de meta acumulada (mostra o ritmo, não só o total).
- Bloco central direito: cascata explicando a variação YoY por categoria.
- Bloco inferior esquerdo: matriz por loja com barras de dados na receita, ícone de status e cor por margem.
- Bloco inferior direito: Top 10 produtos por contribuição de margem e Bottom 10 por margem negativa.
- Tooltip de página em toda a matriz: tendência de 12 meses da loja ou do produto.
- Drill-through por loja: página com detalhe de vendedores, ruptura e mix.
- Página oculta 'Dicionário': definição escrita de cada indicador e a origem do dado.

DESAFIO

Implemente o painel completo e, depois, responda com ele: se a rede cresceu 8% no ano, quanto desse crescimento veio de mais cupons, quanto de ticket maior e quanto de inflação de preço? Crie a decomposição em cascata que separa efeito volume, efeito mix e efeito preço.

DRE gerencial, fluxo de caixa e orçado x realizado

Contexto

Indústria de médio porte, faturamento de R\$ 320 milhões por ano, três unidades de negócio e cinco centros de custo por unidade. O fechamento gerencial levava 9 dias úteis e era montado no Excel por duas pessoas. A meta é publicar a DRE no terceiro dia útil, com abertura por unidade, comparação com orçamento e explicação automática das maiores variações.

Perguntas que o painel responde

- Qual foi o resultado do mês por unidade de negócio e por centro de custo?
- Quais linhas da DRE mais se afastaram do orçamento e em quanto?
- Temos caixa para os próximos 60 dias, considerando contas a pagar e a receber?
- A margem caiu por preço, por custo de matéria-prima ou por mix de produto?
- Quais clientes concentram a inadimplência?

Indicadores e metas

Indicador	Meta	Por que importa
Receita Bruta e Líquida	Conforme orçamento	Base de tudo; a diferença entre elas revela a carga tributária
Margem de Contribuição	$\geq 38\%$	Separa o que é custo variável do que é estrutura
EBITDA e Margem EBITDA	$\geq 15\%$	Geração de caixa operacional, comparável entre empresas
Variação Orçamentária	$\pm 5\%$	Desvio é o que exige ação, não o valor absoluto
Saldo e Projeção de Caixa	≥ 45 dias de cobertura	Lucro não paga fornecedor; caixa paga
Prazo Médio de Recebimento	≤ 42 dias	Cada dia a mais é capital de giro imobilizado
Inadimplência > 30 dias	$\leq 2,5\%$	Antecipa perda que ainda não virou provisão

Modelo de dados

O coração do modelo financeiro é o plano de contas hierárquico com sinal contábil e ordem de exibição. Sem isso, a DRE nunca sai na ordem certa nem soma corretamente.

Tabela	Papel	Observação
fLancamentos	Fato	Um lançamento contábil por linha, com data, conta, centro de custo e valor
fOrcamento	Fato	Mês × Conta × Centro de custo
fContasReceber	Fato	Título em aberto, com vencimento e data de baixa
fContasPagar	Fato	Título a pagar, com vencimento previsto
dPlanoContas	Dimensão	Conta, grupo, subgrupo, nível, sinal, ordem
dCentroCusto	Dimensão	Centro, área, unidade de negócio
dCalendario	Dimensão	Com ano fiscal, se diferente do civil

SQL

```
-- A coluna Sinal resolve o maior problema da DRE em Power BI:
-- despesas gravadas como positivo precisam ser subtraídas.
SELECT
    IdConta,
    Codigo,          -- 3.01.02
    Descricao,       -- Despesas com Pessoal
    Grupo,           -- Despesas Operacionais
    Subgrupo,        -- Administrativas
    Nivel,           -- 3
    Sinal,           -- 1 para receita, -1 para custo e despesa
    OrdemExibicao,    -- 310102 (numérico, controla a ordem na DRE)
    ContaDRE         -- Marca quais contas entram no demonstrativo
FROM dbo.PlanoContas;
```

ATENÇÃO

Use a coluna OrdemExibicao e configure 'Classificar por coluna' em dPlanoContas[Descricao]. Sem isso, 'Receita Bruta' aparece depois de 'Despesas' e o demonstrativo fica ilegível.

Medidas DAX do projeto

Valor com sinal contábil

Aplicar o sinal na medida — e não na carga — permite exibir tanto a visão gerencial (despesa negativa) quanto a analítica (valor absoluto) do mesmo dado.

DAX

```
Valor =
SUMX( fLancamentos, fLancamentos[Valor] * RELATED( dPlanoContas[Sinal] ) )
```

Linhas da DRE

Com o sinal já aplicado, as linhas se somam naturalmente. É por isso que a coluna Sinal é a decisão de modelagem mais importante da controladoria.

DAX

```
Receita Bruta      = CALCULATE( [Valor], dPlanoContas[Grupo] = "Receita Bruta" )
Deduções          = CALCULATE( [Valor], dPlanoContas[Grupo] = "Deduções" )
Receita Líquida    = [Receita Bruta] + [Deduções]
CPV               = CALCULATE( [Valor], dPlanoContas[Grupo] = "Custo dos Produtos Vendidos" )
Lucro Bruto        = [Receita Líquida] + [CPV]
Despesas Operacionais = CALCULATE( [Valor], dPlanoContas[Grupo] = "Despesas Operacionais" )
EBITDA            = [Lucro Bruto] + [Despesas Operacionais]
Margem EBITDA %    = DIVIDE( [EBITDA], [Receita Líquida] )
```

DRE em matriz única

Uma tabela auxiliar dLinhaDRE, desconectada, permite intercalar subtotais calculados entre as contas analíticas na mesma matriz.

DAX

```
Resultado DRE =
VAR Grupo = SELECTEDVALUE( dPlanoContas[Grupo] )
VAR Subtotal = SELECTEDVALUE( dLinhaDRE[Linha] )
RETURN
SWITCH( Subtotal,
    "Receita Líquida", [Receita Líquida],
    "Lucro Bruto",    [Lucro Bruto],
    "EBITDA",        [EBITDA],
    [Valor]
)
```

Orçado x Realizado

Gastar menos que o orçado é favorável; faturar menos é desfavorável. O status precisa conhecer a natureza da conta — é o erro mais comum em painéis financeiros.

DAX

```
Orçado =
CALCULATE(
    SUM( fOrçamento[Valor] ),
    TREATAS( VALUES( dCalendario[AnoMes] ), fOrçamento[AnoMes] ),
    TREATAS( VALUES( dPlanoContas[IdConta] ), fOrçamento[IdConta] ),
    TREATAS( VALUES( dCentroCusto[IdCentro] ), fOrçamento[IdCentro] )
)

Variação $ = [Valor] - [Orçado]

Variação % = DIVIDE( [Variação $], ABS( [Orçado] ) )

Status Orçamento =
VAR Ehreceita = SELECTEDVALUE( dPlanoContas[Sinal] ) = 1
VAR V = [Variação %]
RETURN
SWITCH( TRUE(),
    ISBLANK( [Orçado] ), "Sem orçamento",
    Ehreceita && V >= 0.00, "Favorável",
    Ehreceita && V >= -0.05, "Atenção",
    Ehreceita, "Desfavorável",
    NOT Ehreceita && V <= 0.00, "Favorável",
    NOT Ehreceita && V <= 0.05, "Atenção",
    "Desfavorável"
)
```

Fluxo de caixa projetado

USERELATIONSHIP é essencial: os títulos têm data de emissão (relacionamento ativo) e de vencimento (inativo). O fluxo de caixa olha o vencimento.

DAX

```
Saldo Inicial =  
CALCULATE(  
    [Movimento de Caixa],  
    dCalendario[Date] < MIN( dCalendario[Date] ),  
    REMOVEFILTERS( dCalendario )  
)  
  
A Receber no Período =  
CALCULATE(  
    SUM( fContasReceber[Valor] ),  
    USERELATIONSHIP( fContasReceber[DataVencimento], dCalendario[Date] ),  
    ISBLANK( fContasReceber[DataBaixa] )  
)  
  
A Pagar no Período =  
CALCULATE(  
    SUM( fContasPagar[Valor] ),  
    USERELATIONSHIP( fContasPagar[DataVencimento], dCalendario[Date] ),  
    ISBLANK( fContasPagar[DataBaixa] )  
)  
  
Saldo Projetado =  
[Saldo Inicial] +  
SUMX(  
    FILTER( ALLSELECTED( dCalendario[Date] ), dCalendario[Date] <= MAX( dCalendario[Date] ) ),  
    [A Receber no Período] - [A Pagar no Período]  
)  
  
Dias de Cobertura =  
DIVIDE( [Saldo Atual], DIVIDE( [Despesa Operacional 90d], 90 ) )
```

Aging de recebíveis

A tabela de faixas desconectada é o padrão para aging: as faixas ficam configuráveis sem alterar o modelo.

DAX

```
-- dFaixaAging: tabela desconectada com Faixa, Min, Max, Ordem
Valor na Faixa =
VAR MinD = SELECTEDVALUE( dFaixaAging[Min] )
VAR MaxD = SELECTEDVALUE( dFaixaAging[Max] )
RETURN
CALCULATE(
    SUM( fContasReceber[Valor] ),
    FILTER(
        fContasReceber,
        ISBLANK( fContasReceber[DataBaixa] ) &&
        DATEDIFF( fContasReceber[DataVencimento], TODAY(), DAY ) >= MinD &&
        DATEDIFF( fContasReceber[DataVencimento], TODAY(), DAY ) < MaxD
    )
)

Inadimplência % =
DIVIDE(
    CALCULATE( [Valor na Faixa], dFaixaAging[Min] >= 30 ),
    [Total a Receber]
)

PMR =
VAR TitulosBaixados =
    FILTER( fContasReceber, NOT ISBLANK( fContasReceber[DataBaixa] ) )
RETURN
AVERAGEX(
    TitulosBaixados,
    DATEDIFF( fContasReceber[DataEmissao], fContasReceber[DataBaixa], DAY )
)
```

Explicação automática da variação

Essa medida em uma caixa de texto elimina a pergunta 'o que explica a diferença?' antes que ela seja feita.

DAX

```
Maiores Desvios =
VAR Base =
    ADDCOLUMNS(
        VALUES( dPlanoContas[Descricao] ),
        "@Desvio", [Variação $]
    )
VAR Top3 = TOPN( 3, Base, ABS( [@Desvio] ), DESC )
RETURN
CONCATENATEX(
    Top3,
    dPlanoContas[Descricao] & " (" & FORMAT( [@Desvio], "R$ #,##0;(R$ #,##0)" ) & ")",
    "; ",
    ABS( [@Desvio] ), DESC
)
```

Layout do dashboard

- Página 1 — DRE: matriz com o plano de contas hierárquico nas linhas, e colunas Realizado, Orçado, Variação R\$ e Variação %, com cor por Status Orçamento e ícone de tendência.

- Cabeçalho da página 1: cartões de Receita Líquida, Lucro Bruto, EBITDA, Margem EBITDA, todos com variação contra orçamento.
- Caixa de texto no topo com a medida Maiores Desvios.
- Página 2 — Caixa: linha de saldo projetado nos próximos 90 dias, com faixa sombreada indicando o mínimo de segurança; colunas de entradas e saídas por semana.
- Página 3 — Recebíveis: aging em barras por faixa, matriz de clientes ordenada por valor vencido, PMR em linha histórica.
- Página 4 — Centro de Custo: árvore hierárquica de decomposição da despesa, permitindo ao gestor achar a origem do estouro.
- Drill-through em qualquer conta: lista de lançamentos com número do documento e link para o ERP.

DESAFIO

Construa a DRE completa com subtotais intercalados e a análise de variação orçamentária com status sensível à natureza da conta. Depois adicione uma quarta coluna com o acumulado do ano contra o orçamento anual e o percentual de consumo do orçamento.

Painel de People Analytics: headcount, turnover e clima

Contexto

Empresa de serviços com 2.400 colaboradores em 9 filiais. O RH reportava headcount por e-mail no dia 10 e turnover apenas no fechamento do trimestre — sempre tarde demais para agir. O objetivo é um painel que mostre movimentação diária, turnover por área com causa provável e risco de saída por perfil.

Perguntas que o painel responde

- Quantas pessoas eu tinha em cada mês e como esse número se moveu?
- Quais áreas têm turnover acima da média e desde quando?
- Existe relação entre tempo desde a última promoção e pedido de demissão?
- Quanto custou a rotatividade no último ano, por área?
- Qual é o perfil de quem sai nos primeiros 90 dias?

Indicadores e metas

Indicador	Meta	Por que importa
Headcount Ativo	Conforme quadro aprovado	Base de todos os outros indicadores; precisa ser uma foto de data, não um total
Turnover Mensal e 12M	$\leq 1,8\%$ / $\leq 18\%$	Mede a perda de conhecimento e o custo de reposição
Turnover Voluntário x Involuntário	Voluntário $\leq 60\%$	Saída voluntária de talento é falha de gestão; involuntária é falha de seleção
Tempo Médio de Casa	$\geq 3,2$ anos	Indicador de maturidade e retenção real
Tempo Médio de Preenchimento de Vaga	≤ 38 dias	Vaga aberta é entrega que não acontece
Custo de Reposição	Reduzir 15% a.a.	Traduz turnover em dinheiro e patrocina o investimento em retenção
Absenteísmo %	$\leq 2,2\%$	Sinal precoce de clima, saúde e sobrecarga
eNPS	≥ 40	A pergunta mais preditiva de saída voluntária

Modelo de dados

O desafio central de RH em Power BI é que o headcount é um saldo em uma data, não uma soma. A solução é uma fato de movimentação (admissões e desligamentos) e uma medida de saldo acumulado — ou, em bases grandes, um snapshot periódico.

Tabela	Papel	Observação
fMovimentacao	Fato	Uma linha por evento: admissão, promoção, transferência, desligamento
fSnapshotMensal	Fato	Foto do quadro no último dia de cada mês (opcional, para bases grandes)
fAbsenteismo	Fato	Dia × Colaborador × Tipo de ausência
fPesquisaClima	Fato	Resposta por colaborador e por onda de pesquisa
dColaborador	Dimensão	Matrícula, cargo, área, gestor, data de admissão, gênero, faixa etária
dCargo	Dimensão	Cargo, nível, faixa salarial
dCalendario	Dimensão	Dia
dMotivoSaida	Dimensão	Motivo, tipo (voluntário / involuntário)

ATENÇÃO

RLS não é opcional aqui. Cada gestor deve ver apenas a própria cadeia (PATH + PATHCONTAINS), e colunas de salário exigem OLS. Antes de publicar, valide com o jurídico e com a LGPD da empresa quais campos podem aparecer e para quem.

Medidas DAX do projeto

Headcount em uma data

Saldo em uma data, não soma de linhas. Esta é a medida que sustenta todo o painel — e a que mais costuma vir errada em planilhas.

DAX

```
Headcount =
VAR DataRef = MAX( dCalendario[Date] )
RETURN
CALCULATE(
    DISTINCTCOUNT( dColaborador[Matricula] ),
    FILTER(
        ALL( dColaborador ),
        dColaborador[DataAdmissao] <= DataRef &&
        ( ISBLANK( dColaborador[DataDesligamento] ) || dColaborador[DataDesligamento] > DataRef
    )
)
```

Admissões, desligamentos e headcount médio

Duas datas na mesma dimensão exigem relacionamentos inativos ativados sob demanda com USERELATIONSHIP.

DAX

```
Admissões =
CALCULATE(
    COUNTROWS( dColaborador ),
    USERELATIONSHIP( dColaborador[DataAdmissao], dCalendario[Date] )
)

Desligamentos =
CALCULATE(
    COUNTROWS( dColaborador ),
    USERELATIONSHIP( dColaborador[DataDesligamento], dCalendario[Date] )
)

Headcount Médio =
VAR Inicial =
    CALCULATE( [Headcount], dCalendario[Date] = MIN( dCalendario[Date] ) - 1, REMOVEFILTERS( dCalendario ) )
VAR Final = [Headcount]
RETURN DIVIDE( Inicial + Final, 2 )
```

Turnover

Turnover precoce é o indicador que mais rápido paga o painel: ele aponta falha em seleção, onboarding ou expectativa de vaga.

DAX

```
Turnover Mensal % = DIVIDE( [Desligamentos], [Headcount Médio] )

Turnover 12M % =
VAR Deslig12M =
    CALCULATE( [Desligamentos], DATESINPERIOD( dCalendario[Date], MAX( dCalendario[Date] ), -12, MONTH ) )
VAR HC12M =
    AVERAGEX(
        VALUES( dCalendario[AnoMes] ),
        CALCULATE( [Headcount], LASTDATE( dCalendario[Date] ) )
    )
RETURN DIVIDE( Deslig12M, HC12M )

Turnover Voluntário % =
DIVIDE(
    CALCULATE( [Desligamentos], dMotivoSaida[Tipo] = "Voluntário" ),
    [Headcount Médio]
)

Turnover Precoce (< 90 dias) =
CALCULATE(
    [Desligamentos],
    FILTER(
        dColaborador,
        DATEDIFF( dColaborador[DataAdmissao], dColaborador[DataDesligamento], DAY ) <= 90
    )
)
```

Custo da rotatividade

Traduzir turnover em reais é o que transforma um indicador de RH em pauta de diretoria. Calibre as constantes com o time de RH da empresa.

DAX

```
Custo de Reposição =  
VAR CustoRecrutamento = [Desligamentos] * 2800  
VAR CustoTreinamento = [Desligamentos] * 4200  
VAR PerdaProdutividade =  
    SUMX(  
        FILTER( dColaborador, NOT ISBLANK( dColaborador[DataDesligamento] ) ),  
        RELATED( dCargo[SalarioMedio] ) * 3 -- 3 meses de curva de aprendizado  
    )  
RETURN CustoRecrutamento + CustoTreinamento + PerdaProdutividade
```

Risco de saída (score simples)

Não é machine learning: é uma regra de negócio explícita, auditável e discutível com o RH. Em People Analytics, transparência vale mais que sofisticação.

DAX

```
Score de Risco =  
VAR MesesSemPromocao =  
    DATEDIFF( MAX( dColaborador[DataUltimaPromocao] ), TODAY(), MONTH )  
VAR AbsenteismoRel = DIVIDE( [Horas Ausentes], [Horas Previstas] )  
VAR NotaClima = [eNPS Individual]  
VAR TempoCasa = DATEDIFF( MAX( dColaborador[DataAdmissao] ), TODAY(), MONTH )  
RETURN  
    IF( MesesSemPromocao > 30, 3, IF( MesesSemPromocao > 18, 2, 0 ) ) +  
    IF( AbsenteismoRel > 0.05, 2, 0 ) +  
    IF( NotaClima <= 6, 3, IF( NotaClima <= 8, 1, 0 ) ) +  
    IF( TempoCasa BETWEEN 12 && 24, 2, 0 )  
  
Faixa de Risco =  
SWITCH( TRUE(),  
    [Score de Risco] >= 7, "Alto",  
    [Score de Risco] >= 4, "Médio",  
    "Baixo"  
)
```

Absenteísmo e clima

Sempre exiba a taxa de resposta ao lado do eNPS: um eNPS de 60 com 12% de resposta não significa nada.

DAX

```
Absenteísmo % = DIVIDE( SUM( fAbsenteismo[HorasAusentes] ), [Horas Previstas] )  
  
eNPS =  
VAR Promotores = CALCULATE( COUNTROWS( fPesquisaClima ), fPesquisaClima[Nota] >= 9 )  
VAR Detratores = CALCULATE( COUNTROWS( fPesquisaClima ), fPesquisaClima[Nota] <= 6 )  
VAR Respondentes = COUNTROWS( fPesquisaClima )  
RETURN DIVIDE( Promotores - Detratores, Respondentes ) * 100  
  
Taxa de Resposta % =  
DIVIDE( DISTINCTCOUNT( fPesquisaClima[Matricula] ), [Headcount] )
```

Diversidade

Compare sempre dentro do mesmo cargo e nível: a razão bruta mistura composição de quadro com diferença salarial real.

DAX

Participação por Gênero % =

```
DIVIDE( [Headcount], CALCULATE( [Headcount], REMOVEFILTERS( dColaborador[Genero] ) ) )
```

Razão Salarial por Gênero =

```
VAR MediaF = CALCULATE( AVERAGE( fRemuneracao[Salario] ), dColaborador[Genero] = "F" )
```

```
VAR MediaM = CALCULATE( AVERAGE( fRemuneracao[Salario] ), dColaborador[Genero] = "M" )
```

```
RETURN DIVIDE( MediaF, MediaM )
```

Layout do dashboard

- Página 1 — Visão Geral: cartões de Headcount, Admissões, Desligamentos, Turnover 12M, Tempo de Casa e Absenteísmo; gráfico de colunas de movimentação mensal (admissões positivas, desligamentos negativos) com linha de headcount sobreposta.
- Página 1, bloco inferior: matriz por área com turnover, headcount e cor semântica; small multiples de turnover por filial.
- Página 2 — Turnover: cascata de movimentação do ano (saldo inicial, admissões, desligamentos, saldo final); pizza (4 fatias) voluntário x involuntário; barras por motivo de saída; curva de sobrevivência por tempo de casa.
- Página 3 — Risco e Clima: matriz de colaboradores por faixa de risco (respeitando a RLS de gestor), eNPS por área com taxa de resposta, correlação entre absenteísmo e nota de clima em dispersão.
- Página 4 — Recrutamento: funil de vagas (abertas, triagem, entrevista, proposta, admitido), tempo médio por etapa, custo por contratação.
- RLS obrigatória: cada gestor vê apenas a própria cadeia via PATHCONTAINS; salários visíveis apenas para o perfil RH.

DESAFIO

Implemente a medida de Headcount por data e valide contra a folha de pagamento de três meses diferentes. Depois construa a curva de sobrevivência: percentual de colaboradores admitidos em cada ano que permanecem na empresa após 6, 12, 24 e 36 meses.

Torre de controle logística: OTIF, estoque e custo de frete

Contexto

Operador logístico com 4 centros de distribuição, 180 rotas semanais e 12 mil SKUs armazenados para 30 embarcadores. Os indicadores existiam, mas cada CD calculava OTIF de um jeito. O objetivo é uma definição única, um painel diário para a operação e um painel mensal para a diretoria e para os clientes.

Perguntas que o painel responde

- Quais clientes estão abaixo do OTIF contratado e por qual causa?
- Quais SKUs estão em ruptura e quais estão obsoletos?
- Qual rota tem o maior custo por entrega e por quê?
- O atraso é do CD, da transportadora ou do cliente que não recebeu?
- Quanto de estoque parado eu tenho hoje, em reais?

Indicadores e metas

Indicador	Meta	Por que importa
OTIF (On Time In Full)	$\geq 95\%$	O indicador que o cliente enxerga; combina prazo e completude
On Time %	$\geq 97\%$	Separa o problema de prazo do problema de disponibilidade
In Full %	$\geq 98\%$	Mede ruptura e erro de separação
Giro de Estoque	$\geq 9x/\text{ano}$	Estoque parado é capital imobilizado e risco de obsolescência
Cobertura em Dias	30 a 45 dias	Traduz o giro em linguagem de planejamento
Acuracidade de Inventário	$\geq 99,5\%$	Sem acuracidade, todo o resto do painel é ficção
Custo de Frete sobre Receita	$\leq 6,5\%$	Principal linha de custo variável da operação
Ocupação de Veículo	$\geq 85\%$	Frete pago por espaço vazio é perda pura

Modelo de dados

Logística tem múltiplas fatos com granularidades diferentes: pedido, item do pedido, embarque e posição de estoque. Cada uma se relaciona com as dimensões compartilhadas, nunca entre si.

Tabela	Granularidade	Datas relevantes
fPedidos	Pedido	Data do pedido, prometida, expedição, entrega
fltensPedido	Item do pedido	Quantidade pedida x separada x entregue
fEmbarques	Conhecimento de transporte	Coleta, entrega, peso, cubagem, valor do frete
fEstoquePosicao	Dia x CD x SKU	Saldo, custo médio, data da última movimentação
fInventario	Contagem x SKU	Saldo sistema x saldo contado
dCliente / dCD / dProduto / dTransportadora / dRota	Dimensões	—
dCalendario	Dia	Com marcação de dia útil e feriado por estado

BOA PRÁTICA

fPedidos tem quatro datas relevantes. Crie um relacionamento ativo com a data do pedido e três inativos, acionados por USERELATIONSHIP em medidas específicas. Não crie quatro tabelas de calendário — isso destrói a capacidade de comparar.

Medidas DAX do projeto

OTIF e seus componentes

OTIF não é On Time x In Full. É o pedido que cumpriu as duas condições simultaneamente — por isso é sempre menor ou igual ao menor dos dois.

DAX

```
Pedidos Entregues = DISTINCTCOUNT( fPedidos[NumeroPedido] )

On Time =
CALCULATE(
    [Pedidos Entregues],
    FILTER( fPedidos, fPedidos[DataEntrega] <= fPedidos[DataPrometida] )
)

In Full =
CALCULATE(
    [Pedidos Entregues],
    FILTER(
        fPedidos,
        NOT EXISTS(
            FILTER(
                RELATEDTABLE( fItensPedido ),
                fItensPedido[QtdEntregue] < fItensPedido[QtdPedida]
            )
        )
    )
)

OTIF =
CALCULATE(
    [Pedidos Entregues],
    FILTER(
        fPedidos,
        fPedidos[DataEntrega] <= fPedidos[DataPrometida] &&
        fPedidos[QtdItensDivergentes] = 0
    )
)

OTIF % = DIVIDE( [OTIF], [Pedidos Entregues] )
On Time % = DIVIDE( [On Time], [Pedidos Entregues] )
In Full % = DIVIDE( [In Full], [Pedidos Entregues] )
```

Atraso e sua causa

Separar a causa é o que transforma o OTIF de indicador de cobrança em ferramenta de melhoria: o CD e a transportadora param de culpar um ao outro.

DAX

```
Dias de Atraso Médio =
AVERAGEX(
    FILTER( fPedidos, fPedidos[DataEntrega] > fPedidos[DataPrometida] ),
    DATEDIFF( fPedidos[DataPrometida], fPedidos[DataEntrega], DAY )
)

Causa do Atraso =
VAR AtrasoSeparacao =
    CALCULATE( COUNTROWS( fPedidos ), FILTER( fPedidos, fPedidos[DataExpedicao] > fPedidos[DataPrometidaExpedicao] ) )
VAR AtrasoTransporte =
    CALCULATE(
        COUNTROWS( fPedidos ),
        FILTER( fPedidos,
            fPedidos[DataExpedicao] <= fPedidos[DataPrometidaExpedicao] &&
            fPedidos[DataEntrega] > fPedidos[DataPrometida] )
    )
RETURN
SWITCH( TRUE(),
    AtrasoSeparacao > AtrasoTransporte, "CD / Separação",
    AtrasoTransporte > 0, "Transporte",
    "Sem atraso"
)
```

Giro, cobertura e estoque parado

Giro precisa ser anualizado para ser comparável entre períodos de tamanhos diferentes. Cobertura é o mesmo número em linguagem de planejador.

DAX

```
Estoque Médio =
AVERAGEX( VALUES( dCalendario[Date] ), CALCULATE( SUM( fEstoquePosicao[ValorEstoque] ) ) )

CMV Período = CALCULATE( SUM( fItensPedido[CustoTotal] ) )

Giro de Estoque =
VAR DiasPeriodo = COUNTROWS( dCalendario )
RETURN DIVIDE( [CMV Período], [Estoque Médio] ) * DIVIDE( 365, DiasPeriodo )

Cobertura em Dias = DIVIDE( 365, [Giro de Estoque] )

Estoque Parado =
CALCULATE(
    SUM( fEstoquePosicao[ValorEstoque] ),
    FILTER(
        fEstoquePosicao,
        DATEDIFF( fEstoquePosicao[DataUltimaSaida], TODAY(), DAY ) > 120
    )
)

Estoque Parado % = DIVIDE( [Estoque Parado], [Valor Total em Estoque] )
```

Acuracidade de inventário

Acuracidade por SKU e acuracidade financeira contam histórias diferentes: errar 200 parafusos não é o mesmo que errar 2 motores.

DAX

```
Acuracidade % =  
VAR SKUsContados = DISTINCTCOUNT( fInventario[IdProduto] )  
VAR SKUsCorretos =  
    CALCULATE(  
        DISTINCTCOUNT( fInventario[IdProduto] ),  
        FILTER( fInventario, fInventario[SaldoSistema] = fInventario[SaldoContado] )  
    )  
RETURN DIVIDE( SKUsCorretos, SKUsContados )  
  
Acuracidade Financeira % =  
VAR DivergenciaAbs =  
    SUMX( fInventario, ABS( fInventario[SaldoSistema] - fInventario[SaldoContado] ) * fInventari  
o[CustoUnitario] )  
RETURN 1 - DIVIDE( DivergenciaAbs, [Valor Total em Estoque] )
```

Custo de frete e ocupação

Ocupação é o maior entre peso e cubagem: carga leve e volumosa lota o caminhão sem atingir o peso máximo.

DAX

```
Custo de Frete = SUM( fEmbarques[ValorFrete] )  
  
Frete sobre Receita % = DIVIDE( [Custo de Frete], [Receita de Vendas] )  
  
Custo por Entrega = DIVIDE( [Custo de Frete], DISTINCTCOUNT( fEmbarques[IdEntrega] ) )  
  
Custo por Kg = DIVIDE( [Custo de Frete], SUM( fEmbarques[PesoKg] ) )  
  
Ocupação de Veículo % =  
AVERAGEX(  
    VALUES( fEmbarques[IdViagem] ),  
    VAR PesoUtil = CALCULATE( SUM( fEmbarques[PesoKg] ) )  
    VAR Capacidade = CALCULATE( MAX( dVeiculo[CapacidadeKg] ) )  
    VAR CubUtil = CALCULATE( SUM( fEmbarques[CubagemM3] ) )  
    VAR CapCub = CALCULATE( MAX( dVeiculo[CapacidadeM3] ) )  
    RETURN MAX( DIVIDE( PesoUtil, Capacidade ), DIVIDE( CubUtil, CapCub ) )  
)
```

Classificação XYZ (previsibilidade de demanda)

Cruzar ABC (valor) com XYZ (previsibilidade) gera 9 estratégias de estoque distintas — é a análise de maior retorno em supply chain.

DAX

```
Coef. Variação Demanda =  
VAR Serie =  
    ADDCOLUMNS( VALUES( dCalendario[AnoMes] ), "@Qtd", [Quantidade Vendida] )  
VAR Media = AVERAGEX( Serie, [@Qtd] )  
VAR Desvio =  
    SQRT( DIVIDE( SUMX( Serie, ( [@Qtd] - Media ) ^ 2 ), COUNTROWS( Serie ) - 1 ) )  
RETURN DIVIDE( Desvio, Media )  
  
Classe XYZ =  
SWITCH( TRUE(),  
    [Coef. Variação Demanda] <= 0.25, "X - Previsível",  
    [Coef. Variação Demanda] <= 0.60, "Y - Variável",  
    "Z - Errático"  
)
```

Layout do dashboard

- Página 1 — Torre de Controle (operação, atualizada de hora em hora): cartões de OTIF do dia, pedidos em atraso, pedidos a expedir, veículos em rota; tabela de pedidos críticos com semáforo e link para o WMS.
- Página 2 — OTIF Gerencial: linha de OTIF mensal com meta; decomposição em On Time e In Full; barras por causa de atraso; matriz por cliente com OTIF contratado x realizado.
- Página 3 — Estoque: matriz ABC x XYZ com valor em estoque em cada célula; barras de estoque parado por faixa de dias; giro e cobertura por CD; acuracidade por contagem.
- Página 4 — Custo: custo por entrega e por kg por rota; ocupação média por transportadora; dispersão distância x custo revelando rotas fora da curva.
- Tooltip de página em toda matriz de cliente: evolução de 12 meses do OTIF daquele cliente.
- Drill-through por rota: detalhe de cada viagem com peso, cubagem, ocupação e custo.

DESAFIO

Implemente OTIF com a separação de causa e valide a definição com a operação dos 4 CDs. Depois construa a matriz ABC x XYZ e proponha uma política de estoque distinta para cada uma das 9 células.

Funil completo: da mídia paga ao LTV do cliente

Contexto

E-commerce de moda com investimento mensal de R\$ 1,2 milhão em mídia distribuído em cinco canais. Cada plataforma reportava suas próprias conversões, e a soma das conversões declaradas era 2,3 vezes maior que o número real de pedidos. O objetivo é uma fonte única baseada no pedido real, com atribuição consistente, CAC por canal e LTV por coorte.

Perguntas que o painel responde

- Qual canal traz clientes que continuam comprando, e não apenas o primeiro pedido?
- Onde o funil perde mais gente: tráfego, produto, carrinho ou pagamento?
- Qual o CAC real por canal, considerando o pedido efetivado e não a conversão declarada?
- Quanto uma coorte de clientes gera em 3, 6 e 12 meses?
- Que campanhas estão queimando verba sem gerar cliente novo?

Indicadores e metas

Indicador	Meta	Por que importa
ROAS (Retorno sobre Investimento em Mídia)	$\geq 4,5x$	Eficiência imediata da mídia paga
CAC (Custo de Aquisição)	$\leq R\$ 78$	Quanto custa trazer um cliente novo, não um pedido
LTV	$\geq R\$ 320$	Sem LTV, o CAC não tem referência de julgamento
Razão LTV/CAC	$\geq 3,0$	Abaixo de 3, o crescimento consome caixa
Taxa de Conversão do Funil	$\geq 2,4\%$	Aponta onde o visitante desiste
Ticket Médio	R\$ 264	Alavanca mais barata de crescimento de receita
Taxa de Recompra 90d	$\geq 28\%$	Retenção é o que separa negócio sustentável de queima de caixa
Participação de Receita Orgânica	$\geq 35\%$	Dependência de mídia paga é risco estratégico

Modelo de dados

A regra de ouro do marketing analytics: a fonte da verdade é o pedido do seu sistema, nunca a conversão declarada pela plataforma de mídia. As plataformas entram no modelo apenas como custo e como origem atribuída.

Tabela	Origem	Granularidade
fPedidos	E-commerce (fonte da verdade)	Pedido, com UTM de origem
fInvestimento	APIs das plataformas	Dia × Campanha × Canal
fSessoes	Analytics	Dia × Canal × Dispositivo
fFunil	Analytics (eventos)	Dia × Etapa × Canal
dCliente	E-commerce	Cliente, com data da primeira compra
dCampanha	Plataformas	Campanha, canal, público, criativo
dCalendario	Gerada	Dia

ATENÇÃO

Cada plataforma reivindica a conversão se teve qualquer contato dentro da janela de atribuição. Um cliente que viu anúncio no Instagram, pesquisou no Google e clicou no e-mail gera três conversões declaradas e um pedido real. Somar é matematicamente errado.

M

```
let
    Origem = Sql.Database("servidor", "ecommerce"),
    Pedidos = Origem{[Schema="dbo", Item="Pedidos"]}[Data],
    // UTMs chegam sujos: maiúsculas, espaços, variações de nome
    LimpaSource = Table.TransformColumns(Pedidos, {"utm_source", each Text.Lower(Text.Trim(_ ?? "direto")), type text}),
    LimpaMedium = Table.TransformColumns(LimpaSource, {"utm_medium", each Text.Lower(Text.Trim(_ ?? "none")), type text}),
    // Agrupa variações conhecidas em um canal padronizado
    Canal = Table.AddColumn(LimpaMedium, "Canal", each
        if List.Contains({"google", "googleads", "adwords"}, [utm_source]) and [utm_medium] = "cpc"
        then "Google Ads"
        else if List.Contains({"facebook", "instagram", "meta", "fb", "ig"}, [utm_source]) then "Meta Ads"
        else if [utm_medium] = "email" then "E-mail"
        else if [utm_medium] = "organic" then "Orgânico"
        else if [utm_source] = "direto" then "Direto"
        else "Outros", type text)
in
    Canal
```

Medidas DAX do projeto

Investimento, ROAS e CPA

ROAS ignora a margem. Um ROAS de 4x em um produto de 20% de margem destrói valor. A última medida é a que realmente importa.

DAX

```
Investimento = SUM( fInvestimento[Custo] )
```

```
Receita = SUM( fPedidos[ValorTotal] )
```

```
ROAS = DIVIDE( [Receita], [Investimento] )
```

```
CPA = DIVIDE( [Investimento], [Qtd Pedidos] )
```

```
Margem de Contribuição após Mídia =  
[Receita] * [Margem Bruta %] - [Investimento]
```

CAC real (cliente novo, não pedido)

A diferença entre CPA e CAC é a mais cara de ignorar: você pode estar pagando caro para trazer de volta quem já compraria de qualquer jeito.

DAX

```
Clientes Novos =
```

```
CALCULATE(  
    DISTINCTCOUNT( fPedidos[IdCliente] ),  
    FILTER(  
        fPedidos,  
        fPedidos[DataPedido] = RELATED( dCliente[DataPrimeiraCompra] )  
    )  
)
```

```
CAC = DIVIDE( [Investimento], [Clientes Novos] )
```

```
Participação de Clientes Novos % =  
DIVIDE( [Clientes Novos], DISTINCTCOUNT( fPedidos[IdCliente] ) )
```

LTV por coorte

A matriz de coorte (linhas = mês de aquisição, colunas = meses desde a primeira compra) é o visual mais valioso de marketing: revela se a qualidade do cliente adquirido está melhorando ou piorando.

DAX

```
Coorte = FORMAT( MAX( dCliente[DataPrimeiraCompra] ), "yyyy-mm" )

Receita Acumulada da Coorte =
VAR MesesDesdePrimeira = SELECTEDVALUE( dMesesDesde[Meses] )
RETURN
CALCULATE(
    [Receita],
    FILTER(
        fPedidos,
        DATEDIFF( RELATED( dCliente[DataPrimeiraCompra] ), fPedidos[DataPedido], MONTH ) <= MesesDesdePrimeira
    )
)

LTV por Cliente =
DIVIDE( [Receita Acumulada da Coorte], [Clientes da Coorte] )

Razão LTV/CAC = DIVIDE( [LTV por Cliente], [CAC] )

LTV Preditivo =
-- Modelo simples e defensável: frequência x ticket x margem x horizonte
VAR ComprasPorAno = DIVIDE( [Qtd Pedidos], [Clientes Ativos] ) * 12 / [Meses do Período]
VAR Ticket = [Ticket Médio]
VAR MargemPct = [Margem Bruta %]
VAR VidaMediaAnos = 2.5
RETURN ComprasPorAno * Ticket * MargemPct * VidaMediaAnos
```

Funil e taxas de conversão

Abandono de checkout acima de 30% quase sempre é frete, prazo ou meio de pagamento — não é problema de mídia.

DAX

```
Sessões = SUM( fSessoes[Sessoes] )
Visualizações = CALCULATE( SUM( fFunil[Eventos] ), fFunil[Etapas] = "Produto" )
Carrinhos = CALCULATE( SUM( fFunil[Eventos] ), fFunil[Etapas] = "Carrinho" )
Checkouts = CALCULATE( SUM( fFunil[Eventos] ), fFunil[Etapas] = "Checkout" )

Conversão Geral % = DIVIDE( [Qtd Pedidos], [Sessões] )
Conv. Sessão→Produto % = DIVIDE( [Visualizações], [Sessões] )
Conv. Produto→Carrinho % = DIVIDE( [Carrinhos], [Visualizações] )
Conv. Carrinho→Checkout % = DIVIDE( [Checkouts], [Carrinhos] )
Conv. Checkout→Pedido % = DIVIDE( [Qtd Pedidos], [Checkouts] )

Maior Gargalo do Funil =
VAR Etapas =
    DATATABLE( "Etapas", STRING, "Ordem", INTEGER, { { "Sessão→Produto", 1 }, { "Produto→Carrinho", 2 }, { "Carrinho→Checkout", 3 }, { "Checkout→Pedido", 4 } } )
VAR ComTaxa =
    ADDCOLUMNS( Etapas, "@Taxa",
        SWITCH( [Ordem],
            1, [Conv. Sessão→Produto %],
            2, [Conv. Produto→Carrinho %],
            3, [Conv. Carrinho→Checkout %],
            4, [Conv. Checkout→Pedido %] ) )
RETURN
CONCATENATEX( TOPN( 1, ComTaxa, [@Taxa], ASC ), [Etapas] )
```

Atribuição multicanal

Canais de descoberta (social, display) brilham em first click; canais de captura (busca de marca, e-mail) brilham em last click. Mostrar os dois lado a lado encerra a disputa interna por orçamento.

DAX

```
Receita Last Click =  
CALCULATE( [Receita], USERRELATIONSHIP( fPedidos[CanalUltimoClique], dCampanha[Canal] ) )  
  
Receita First Click =  
CALCULATE( [Receita], USERRELATIONSHIP( fPedidos[CanalPrimeiroClique], dCampanha[Canal] ) )  
  
Diferença de Atribuição % =  
DIVIDE( [Receita Last Click] - [Receita First Click], [Receita First Click] )
```

Recompra e retenção

Recompra é o indicador que a mídia paga não consegue comprar. É a prova de que o produto e a experiência funcionam.

DAX

```
Taxa de Recompra 90d =  
VAR ClientesBase =  
    CALCULATE( VALUES( fPedidos[IdCliente] ), DATESINPERIOD( dCalendario[Date], TODAY() - 90, -90, DAY ) )  
VAR Compraram =  
    CALCULATE(  
        DISTINCTCOUNT( fPedidos[IdCliente] ),  
        DATESINPERIOD( dCalendario[Date], TODAY(), -90, DAY ),  
        ClientesBase  
    )  
RETURN DIVIDE( Compraram, COUNTROWS( ClientesBase ) )
```

Layout do dashboard

- Página 1 — Visão de Growth: cartões de Receita, Investimento, ROAS, CAC, LTV/CAC e Clientes Novos; linha de receita e investimento em eixo duplo justificado (ambos em reais); barras de ROAS por canal ordenadas.
- Página 2 — Funil: visual de funil com as 5 etapas e taxas entre elas; comparação de funil por dispositivo (mobile x desktop, quase sempre reveladora); caixa de texto com a medida Maior Gargalo do Funil.
- Página 3 — Coortes: matriz de coorte com cor de fundo por medida (mapa de calor de retenção); curva de LTV acumulado por coorte em linhas sobrepostas; LTV/CAC por canal de aquisição.
- Página 4 — Campanhas: tabela detalhada com investimento, pedidos, CAC, ROAS e margem após mídia, ordenada por margem; destaque em vermelho para campanhas com margem negativa.
- Página 5 — Atribuição: comparação first click x last click por canal, em barras opostas.
- Rodapé em todas as páginas: 'Fonte da verdade: pedidos do e-commerce. Conversões declaradas pelas plataformas não são somadas.'

DESAFIO

Monte a matriz de coorte com mapa de calor de retenção e responda: a qualidade dos clientes adquiridos nos últimos 6 meses está melhor ou pior que a do ano anterior? Depois calcule o LTV/CAC por canal e defina para qual canal você realocaria 20% do orçamento.

Gestão hospitalar: ocupação, tempo de permanência e qualidade assistencial

Contexto

Hospital geral de 240 leitos, pronto-socorro com 320 atendimentos por dia e 11 especialidades cirúrgicas. A diretoria clínica e a diretoria financeira discutiam com números diferentes: uma olhava desfecho, a outra olhava custo. O objetivo é um painel único que mostre eficiência operacional sem perder de vista a qualidade assistencial.

Perguntas que o painel responde

- Quais unidades estão acima da ocupação de segurança neste momento?
- Quais especialidades têm permanência acima da média sem justificativa de gravidade?
- Onde está a fila do pronto-socorro e qual etapa a causa?
- Quais convênios mais glosam e em quais procedimentos?
- A reinternação está concentrada em algum diagnóstico ou em alguma equipe?

Indicadores e metas

Indicador	Meta	Por que importa
Taxa de Ocupação	85% a 90%	Acima de 90% o hospital perde capacidade de resposta; abaixo de 80%, ociosidade custosa
Tempo Médio de Permanência	$\leq 4,2$ dias	Cada dia a mais é leito indisponível e risco de infecção
Giro de Leito	$\geq 6,5$ /mês	Traduz ocupação e permanência em capacidade de atendimento
Tempo de Porta-Médico (PS)	≤ 18 min	Indicador de experiência e de segurança no pronto-socorro
Taxa de Reinternação em 30 dias	$\leq 7\%$	Principal indicador de qualidade do desfecho e da alta
Taxa de Infecção Hospitalar	$\leq 3,5\%$	Indicador crítico de segurança do paciente
Taxa de Cancelamento Cirúrgico	$\leq 5\%$	Desperdício de sala, equipe e expectativa do paciente
Glosa sobre Faturamento	$\leq 4\%$	Receita reconhecida que nunca entra no caixa

Modelo de dados

Saúde combina eventos (internação, atendimento, cirurgia) com posições diárias (censo de leitos). A fato de censo é o que permite calcular ocupação corretamente — não tente derivar ocupação apenas das internações.

Tabela	Granularidade	Uso
fInternacoes	Internação	Entrada, alta, diagnóstico, desfecho, equipe
fCensoDiario	Dia × Leito	Ocupado, vago, bloqueado — base da ocupação
fAtendimentosPS	Atendimento	Chegada, triagem, médico, alta, classificação de risco
fCirurgias	Procedimento	Agendada, realizada, cancelada, motivo
fFaturamento	Conta × Item	Valor apresentado, glosado, recebido
dPaciente	Paciente	Sem nome no modelo publicado; use código
dLeito / dUnidade / dEspecialidade / dCID / dConvenio / dProfissional	Dimensões	—

ATENÇÃO

Nome, CPF e diagnóstico identificável não devem estar no modelo publicado. Use código de paciente, agregue por faixa etária e aplique RLS por unidade e por especialidade. Envie o DPO da instituição antes de publicar — isso não é formalidade.

Medidas DAX do projeto

Ocupação e giro de leito

Leitos bloqueados (manutenção, isolamento) devem sair do denominador: incluí-los mascara a ocupação real e faz o hospital parecer folgado quando está no limite.

DAX

```
Leitos Operacionais =  
CALCULATE( DISTINCTCOUNT( fCensoDiario[IdLeito] ), fCensoDiario[Status] <> "Bloqueado" )  
  
Leitos-Dia Disponíveis =  
SUMX( VALUES( dCalendario[Date] ), [Leitos Operacionais] )  
  
Leitos-Dia Ocupados =  
CALCULATE( COUNTROWS( fCensoDiario ), fCensoDiario[Status] = "Ocupado" )  
  
Taxa de Ocupação % =  
DIVIDE( [Leitos-Dia Ocupados], [Leitos-Dia Disponíveis] )  
  
Giro de Leito =  
DIVIDE( [Altas], AVERAGEX( VALUES( dCalendario[Date] ), [Leitos Operacionais] ) )  
  
Intervalo de Substituição =  
-- Horas que o leito fica vago entre uma alta e a próxima internação  
DIVIDE(  
    ( [Leitos-Dia Disponíveis] - [Leitos-Dia Ocupados] ) * 24,  
    [Altas]  
)
```

Tempo de permanência

Comparar permanência bruta entre especialidades é injusto: UTI neurológica e maternidade não são comparáveis. O índice ajustado (observado/esperado) é a métrica defensável clinicamente.

DAX

```
Altas = CALCULATE( COUNTROWS( fInternacoes ), NOT ISBLANK( fInternacoes[DataAlta] ) )  
  
Tempo Médio de Permanência =  
AVERAGEX(  
    FILTER( fInternacoes, NOT ISBLANK( fInternacoes[DataAlta] ) ),  
    DATEDIFF( fInternacoes[DataEntrada], fInternacoes[DataAlta], DAY )  
)  
  
TMP Ajustado pela Gravidade =  
-- Compara a permanência observada com a esperada para o mesmo CID e faixa etária  
VAR Observado = [Tempo Médio de Permanência]  
VAR Esperado =  
    AVERAGEX(  
        VALUES( dCID[Codigo] ),  
        CALCULATE( AVERAGE( dCID[PermanenciaEsperadaDias] ) )  
    )  
RETURN DIVIDE( Observado, Esperado )  
  
Permanência Prolongada =  
CALCULATE(  
    COUNTROWS( fInternacoes ),  
    FILTER(  
        fInternacoes,  
        DATEDIFF( fInternacoes[DataEntrada], COALESCE( fInternacoes[DataAlta], TODAY() ), DAY )  
    )  
>  
    RELATED( dCID[PermanenciaEsperadaDias] ) * 1.5  
)
```

Pronto-socorro: tempos por etapa

A meta de tempo muda conforme a classificação de risco (protocolo de Manchester). Uma média única esconde exatamente o caso que não podia esperar.

DAX

```
Tempo Porta-Triagem =
AVERAGEX( fAtendimentosPS, DATEDIFF( fAtendimentosPS[HoraChegada], fAtendimentosPS[HoraTriagem],
    MINUTE ) )

Tempo Porta-Médico =
AVERAGEX( fAtendimentosPS, DATEDIFF( fAtendimentosPS[HoraChegada], fAtendimentosPS[HoraMedico],
    MINUTE ) )

Tempo Total de Permanência no PS =
AVERAGEX( fAtendimentosPS, DATEDIFF( fAtendimentosPS[HoraChegada], fAtendimentosPS[HoraSaida], M
    INUTE ) )

Dentro da Meta por Classificação % =
VAR MetaMinutos =
    SWITCH( SELECTEDVALUE( fAtendimentosPS[ClassificacaoRisco] ),
        "Vermelho", 0,
        "Laranja", 10,
        "Amarelo", 60,
        "Verde", 120,
        240 )
RETURN
DIVIDE(
    CALCULATE(
        COUNTROWS( fAtendimentosPS ),
        FILTER( fAtendimentosPS,
            DATEDIFF( fAtendimentosPS[HoraChegada], fAtendimentosPS[HoraMedico], MINUTE ) <= Met
aMinutos )
    ),
    COUNTROWS( fAtendimentosPS )
)

Evasão do PS % =
DIVIDE(
    CALCULATE( COUNTROWS( fAtendimentosPS ), fAtendimentosPS[Desfecho] = "Evasão" ),
    COUNTROWS( fAtendimentosPS )
)
```

Qualidade assistencial

Mortalidade institucional exclui óbitos nas primeiras 24 horas, que refletem a gravidade na chegada e não a qualidade do cuidado prestado. Esse tipo de precisão é o que dá credibilidade ao painel junto ao corpo clínico.

DAX

```
Reinternação 30 dias =
CALCULATE(
    COUNTROWS( fInternacoes ),
    FILTER(
        fInternacoes,
        VAR Pac = fInternacoes[IdPaciente]
        VAR Entrada = fInternacoes[DataEntrada]
        VAR AltaAnterior =
            CALCULATE(
                MAX( fInternacoes[DataAlta] ),
                ALLEXCEPT( fInternacoes, fInternacoes[IdPaciente] ),
                fInternacoes[DataAlta] < Entrada
            )
        RETURN
            NOT ISBLANK( AltaAnterior ) &&
            DATEDIFF( AltaAnterior, Entrada, DAY ) <= 30
    )
)

Taxa de Reinternação % = DIVIDE( [Reinternação 30 dias], [Altas] )

Taxa de Infecção Hospitalar % =
DIVIDE(
    CALCULATE( COUNTROWS( fInternacoes ), fInternacoes[InfeccaoHospitalar] = TRUE() ),
    [Altas]
)

Densidade de Incidência de IRAS =
-- Padrão internacional: casos por 1.000 pacientes-dia
DIVIDE( [Casos de IRAS], [Leitos-Dia Ocupados] ) * 1000

Taxa de Mortalidade Institucional % =
DIVIDE(
    CALCULATE( COUNTROWS( fInternacoes ), fInternacoes[Desfecho] = "Óbito", fInternacoes[HorasInternado] > 24 ),
    [Altas]
)
```

Centro cirúrgico

O atraso do primeiro horário é o indicador de maior efeito cascata do hospital: 30 minutos de atraso às 7h viram duas horas às 17h e um cancelamento no fim do dia.

DAX

```
Cirurgias Realizadas = CALCULATE( COUNTROWS( fCirurgias ), fCirurgias[Status] = "Realizada" )
```

```
Taxa de Cancelamento % =
```

```
DIVIDE(  
    CALCULATE( COUNTROWS( fCirurgias ), fCirurgias[Status] = "Cancelada" ),  
    COUNTROWS( fCirurgias )  
)
```

```
Taxa de Ocupação de Sala % =
```

```
DIVIDE(  
    SUMX( fCirurgias, DATEDIFF( fCirurgias[HoraInicio], fCirurgias[HoraFim], MINUTE ) ),  
    [Minutos de Sala Disponíveis]  
)
```

```
Atraso do Primeiro Horário =
```

```
AVERAGEX(  
    FILTER( fCirurgias, fCirurgias[EhPrimeiroHorario] = TRUE() ),  
    DATEDIFF( fCirurgias[HoraAgendada], fCirurgias[HoraInicio], MINUTE )  
)
```

Faturamento e glosas

Glosa é dinheiro já trabalhado e não recebido. Abrir por convênio, procedimento e motivo transforma o indicador em plano de ação para o faturamento.

DAX

```
Valor Apresentado = SUM( fFaturamento[ValorApresentado] )
```

```
Valor Glosado = SUM( fFaturamento[ValorGlosado] )
```

```
Valor Recebido = SUM( fFaturamento[ValorRecebido] )
```

```
Taxa de Glosa % = DIVIDE( [Valor Glosado], [Valor Apresentado] )
```

```
Glosa Recuperada % =
```

```
DIVIDE( SUM( fFaturamento[ValorRecuperado] ), [Valor Glosado] )
```

```
Principal Motivo de Glosa =
```

```
CONCATENATEX(  
    TOPN( 1, VALUES( dMotivoGlosa[Motivo] ), [Valor Glosado], DESC ),  
    dMotivoGlosa[Motivo]  
)
```

Layout do dashboard

- Página 1 — Painel do Dia (atualização de 15 em 15 minutos): mapa de leitos por unidade com cor por status, cartões de ocupação atual, pacientes aguardando leito no PS, cirurgias do dia e altas previstas.
- Página 2 — Eficiência: linha de ocupação com faixas de referência (80% e 90%); TMP por especialidade com índice observado/esperado; giro de leito e intervalo de substituição.
- Página 3 — Pronto-Socorro: funil de tempos por etapa; percentual dentro da meta por classificação de risco; mapa de calor de atendimentos por hora e dia da semana (dimensiona escala de equipe); evasão.
- Página 4 — Qualidade: reinternação por especialidade e por CID; densidade de IRAS por 1.000 pacientes-dia com carta de controle; mortalidade institucional; todos com linha de

referência da meta institucional.

- Página 5 — Centro Cirúrgico: ocupação de sala por hora; cancelamentos por motivo em Pareto; atraso do primeiro horário por equipe.
- Página 6 — Faturamento: glosa por convênio e motivo em Pareto; percentual recuperado; prazo médio de recebimento por convênio.
- RLS por unidade e especialidade. Nenhum dado identificável de paciente no modelo publicado.

DESAFIO

Implemente a taxa de reinternação em 30 dias — é o cálculo mais difícil deste caso, porque exige comparar cada internação com a alta anterior do mesmo paciente. Depois construa a carta de controle de IRAS com média e limites de 3 desvios, e identifique se houve algum ponto fora de controle nos últimos 24 meses.

Chão de fábrica: OEE, manutenção e aderência ao plano

Contexto

Indústria de autopeças com 3 plantas, 46 máquinas críticas e produção em três turnos. Cada planta calculava OEE de uma forma, o que impedia qualquer comparação. A manutenção era reativa e o PCP não conseguia enxergar o impacto das paradas no plano. O objetivo é um padrão único de OEE, visão de perdas por causa e ligação direta entre disponibilidade e aderência ao plano de produção.

Perguntas que o painel responde

- Qual máquina tem o pior OEE e qual das três perdas explica isso?
- Quais são as 5 causas de parada que respondem por 80% do tempo perdido?
- A manutenção preventiva está reduzindo as quebras ou apenas consumindo horas?
- Quantas horas de produção eu perdi por falta de material em vez de falha de máquina?
- O turno da noite produz com a mesma eficiência do turno da manhã?

Indicadores e metas

Indicador	Meta	Por que importa
OEE (Eficiência Global do Equipamento)	$\geq 75\%$ (classe mundial: 85%)	O indicador mestre de manufatura; multiplica disponibilidade, performance e qualidade
Disponibilidade	$\geq 90\%$	Isola perdas de parada (setup, quebra, falta de material)
Performance	$\geq 95\%$	Isola perdas de velocidade e microparadas
Qualidade	$\geq 99\%$	Isola refugo e retrabalho
MTBF (Tempo Médio entre Falhas)	≥ 180 h	Mede a confiabilidade do ativo
MTTR (Tempo Médio de Reparo)	$\leq 2,5$ h	Mede a capacidade de resposta da manutenção
Aderência ao Plano de Produção	$\geq 95\%$	Produzir muito do item errado não atende o cliente
Custo de Manutenção sobre Valor do Ativo	$\leq 3,5\%$ a.a.	Equilibra confiabilidade e gasto
Scrap (Refugo) %	$\leq 1,2\%$	Perda direta de matéria-prima, energia e capacidade

Modelo de dados

O modelo de manufatura gira em torno de duas fatos: apontamento de produção (o que foi produzido) e apontamento de parada (o tempo que não produziu e por quê). A soma dos dois deve fechar exatamente com o tempo de calendário da máquina — essa conciliação é o teste de qualidade do dado.

Tabela	Granularidade	Campos-chave
fProducao	Ordem × Máquina × Turno	Qtd boa, qtd refugo, tempo de ciclo real
fParadas	Evento de parada	Início, fim, máquina, motivo, tipo (planejada/não planejada)
fPlanoProducao	Dia × Máquina × Produto	Quantidade planejada
fManutencao	Ordem de manutenção	Tipo (preventiva/corretiva/preditiva), custo, horas
dMquina	Máquina	Centro de trabalho, planta, capacidade nominal, valor do ativo
dMotivoParada	Motivo	Categoria das 6 grandes perdas
dTurno	Turno	Início, fim, horas úteis, intervalos
dProduto / dCalendario	Dimensões	Tempo de ciclo padrão por produto e máquina

BOA PRÁTICA

Classifique todo motivo de parada em uma das seis: quebra, setup e ajuste (afetam Disponibilidade); microparada, redução de velocidade (afetam Performance); refugo de início, refugo de processo (afetam Qualidade). Sem essa classificação, o OEE vira um número sem plano de ação.

Medidas DAX do projeto

OEE e seus três componentes

Performance limitada a 100% com MIN protege contra tempo de ciclo padrão desatualizado — um clássico que gera OEE de 130% e destrói a confiança no painel.

DAX

```
Tempo Calendário =  
SUMX( VALUES( dMaquina[IdMaquina] ), [Horas do Período] ) * 60 -- em minutos  
  
Paradas Planejadas =  
CALCULATE( SUM( fParadas[DuracaoMinutos] ), dMotivoParada[Tipo] = "Planejada" )  
  
Tempo Programado = [Tempo Calendário] - [Paradas Planejadas]  
  
Paradas Não Planejadas =  
CALCULATE( SUM( fParadas[DuracaoMinutos] ), dMotivoParada[Tipo] = "Não planejada" )  
  
Tempo Operacional = [Tempo Programado] - [Paradas Não Planejadas]  
  
Disponibilidade % = DIVIDE( [Tempo Operacional], [Tempo Programado] )  
  
Tempo Teórico de Produção =  
SUMX( fProducao, fProducao[QtdTotalProduzida] * RELATED( dProduto[TempoCicloPadraoMin] ) )  
  
Performance % =  
MIN( 1, DIVIDE( [Tempo Teórico de Produção], [Tempo Operacional] ) )  
  
Qualidade % =  
DIVIDE( SUM( fProducao[QtdBoa] ), SUM( fProducao[QtdTotalProduzida] ) )  
  
OEE % = [Disponibilidade %] * [Performance %] * [Qualidade %]
```

Perdas em minutos e em peças

Converter perda de OEE em peças e em reais é o que faz a diretoria aprovar investimento em manutenção. Percentual não move orçamento; dinheiro move.

DAX

```
Perda por Disponibilidade (min) = [Paradas Não Planejadas]  
  
Perda por Performance (min) = [Tempo Operacional] - [Tempo Teórico de Produção]  
  
Perda por Qualidade (min) =  
SUMX( fProducao, fProducao[QtdRefugo] * RELATED( dProduto[TempoCicloPadraoMin] ) )  
  
Peças Perdidas Equivalentes =  
DIVIDE(  
    [Perda por Disponibilidade (min)] + [Perda por Performance (min)] + [Perda por Qualidade (min)],  
    AVERAGEX( VALUES( dProduto[Produto] ), dProduto[TempoCicloPadraoMin] )  
)  
  
Receita Potencial Perda =  
[Peças Perdidas Equivalentes] * AVERAGEX( VALUES( dProduto[Produto] ), dProduto[PrecoMedio] )
```

MTBF, MTTR e disponibilidade inerente

MTBF em queda com MTTR estável indica degradação do ativo. MTTR em alta com MTBF estável indica problema de processo de manutenção (peça, equipe ou procedimento) — são diagnósticos e ações completamente diferentes.

DAX

```
Qtd Falhas =  
CALCULATE( COUNTROWS( fParadas ), dMotivoParada[Categoria] = "Quebra" )  
  
Horas em Operação = DIVIDE( [Tempo Operacional], 60 )  
  
MTBF = DIVIDE( [Horas em Operação], [Qtd Falhas] )  
  
Tempo Total de Reparo =  
DIVIDE(  
    CALCULATE( SUM( fParadas[DuracaoMinutos] ), dMotivoParada[Categoria] = "Quebra" ),  
    60  
)  
  
MTTR = DIVIDE( [Tempo Total de Reparo], [Qtd Falhas] )  
  
Disponibilidade Inerente % = DIVIDE( [MTBF], [MTBF] + [MTTR] )  
  
Máquinas Críticas em Risco =  
COUNTROWS(  
    FILTER(  
        VALUES( dMaquina[Maquina] ),  
        [MTBF] < 100 && dMaquina[Críticidade] = "A"  
    )  
)
```

Pareto de paradas

O Pareto de paradas é o visual que mais gera ação imediata no chão de fábrica: em geral 4 ou 5 causas explicam 80% do tempo perdido, e três delas são resolvíveis em semanas.

DAX

```
Tempo de Parada = SUM( fParadas[DuracaoMinutos] )  
  
Parada Acumulada % =  
VAR Base =  
    ADDCOLUMNS( ALLSELECTED( dMotivoParada[Motivo] ), "@Tempo", [Tempo de Parada] )  
VAR Atual = [Tempo de Parada]  
VAR Acumulado = SUMX( FILTER( Base, [@Tempo] >= Atual ), [@Tempo] )  
RETURN DIVIDE( Acumulado, SUMX( Base, [@Tempo] ) )  
  
Está nos 80% Vitais =  
IF( [Parada Acumulada %] <= 0.8, "Vital", "Trivial" )
```

Manutenção: preventiva x corretiva

Referência de classe mundial: pelo menos 80% das horas em manutenção planejada. Abaixo de 50%, a equipe está apagando incêndio e o MTBF só piora.

DAX

```
Horas Preventiva = CALCULATE( SUM( fManutencao[Horas] ), fManutencao[Tipo] = "Preventiva" )
Horas Corretiva = CALCULATE( SUM( fManutencao[Horas] ), fManutencao[Tipo] = "Corretiva" )

Índice de Manutenção Planejada % =
DIVIDE( [Horas Preventiva], [Horas Preventiva] + [Horas Corretiva] )

Cumprimento do Plano de Manutenção % =
DIVIDE(
    CALCULATE( COUNTROWS( fManutencao ), fManutencao[Status] = "Concluída", fManutencao[DentroDo
Prazo] = TRUE() ),
    CALCULATE( COUNTROWS( fManutencao ), fManutencao[Tipo] = "Preventiva" )
)

Custo de Manutenção sobre Ativo % =
DIVIDE( SUM( fManutencao[CustoTotal] ), SUM( dMaquina[ValorAtivo] ) )
```

Aderência ao plano (PCP)

Aderência calculada com MIN(planejado, realizado) é o padrão correto: produzir 200% de um item e 0% de outro dá 100% em uma média ingênua e 50% aqui — que é a verdade.

DAX

```
Qtd Planejada = SUM( fPlanoProducao[QtdPlanejada] )
Qtd Produzida = SUM( fProducao[QtdBoa] )

Aderência ao Plano % =
-- Penaliza tanto produzir a menos quanto produzir a mais do item errado
VAR Base =
    ADDCOLUMNS(
        SUMMARIZE( fPlanoProducao, dProduto[Produto], dCalendario[Date] ),
        "@Plan", [Qtd Planejada],
        "@Real", [Qtd Produzida]
    )
VAR AderenteTotal = SUMX( Base, MIN( [@Plan], [@Real] ) )
VAR PlanTotal = SUMX( Base, [@Plan] )
RETURN DIVIDE( AderenteTotal, PlanTotal )

Desvio de Mix =
SUMX(
    SUMMARIZE( fPlanoProducao, dProduto[Produto] ),
    ABS( [Qtd Produzida] - [Qtd Planejada] )
)
```

Comparação entre turnos

A última medida responde à pergunta mais poderosa da manufatura: quantas horas de produção eu ganharia se todos os turnos operassem como o melhor deles? Costuma ser a maior oportunidade da planta, e não exige investimento em ativo.

DAX

OEE por Turno = [OEE %] -- basta colocar dTurno no eixo

Diferença do Melhor Turno =

```
VAR MelhorOEE = MAXX( ALLSELECTED( dTurno[Turno] ), [OEE %] )
```

```
RETURN [OEE %] - MelhorOEE
```

Ganho Potencial se Todos no Melhor Turno =

```
VAR MelhorOEE = MAXX( ALLSELECTED( dTurno[Turno] ), [OEE %] )
```

```
RETURN
```

```
SUMX(
```

```
    VALUES( dTurno[Turno] ),
```

```
    ( MelhorOEE - [OEE %] ) * [Tempo Programado]
```

```
) / 60
```

Layout do dashboard

- Página 1 — OEE da Planta: cartão grande de OEE com os três componentes ao lado; gráfico de cascata mostrando de 100% até o OEE final, com cada perda nomeada; linha de OEE diário com meta e faixa de classe mundial.
- Página 2 — Por Máquina: matriz com OEE, Disponibilidade, Performance e Qualidade por máquina, ordenada por OEE crescente e com cor semântica; small multiples de tendência por centro de trabalho.
- Página 3 — Paradas: Pareto de motivos com linha acumulada e marcação dos 80% vitais; mapa de calor de paradas por hora e dia da semana; tempo de parada por turno.
- Página 4 — Manutenção: MTBF e MTTR por máquina em dispersão (quadrante inferior direito = ativos críticos); índice de manutenção planejada em linha histórica; cumprimento do plano preventivo; custo sobre ativo.
- Página 5 — PCP: aderência ao plano por produto e por dia; desvio de mix; ganho potencial de nivelamento entre turnos.
- Página 6 — Financeiro da Perda: receita potencial perdida por causa de parada, em reais, acumulada no ano — a página que patrocina o investimento.
- Tooltip de página em toda matriz de máquina: últimas 10 paradas com motivo e duração.

DESAFIO

Implemente o OEE completo e faça a conciliação: a soma de tempo operacional, paradas planejadas e não planejadas deve fechar exatamente com o tempo de calendário de cada máquina. Crie uma medida de auditoria que mostre a diferença — se ela não for zero, o apontamento do chão de fábrica tem falha, e esse é o primeiro problema a resolver antes de confiar em qualquer indicador.