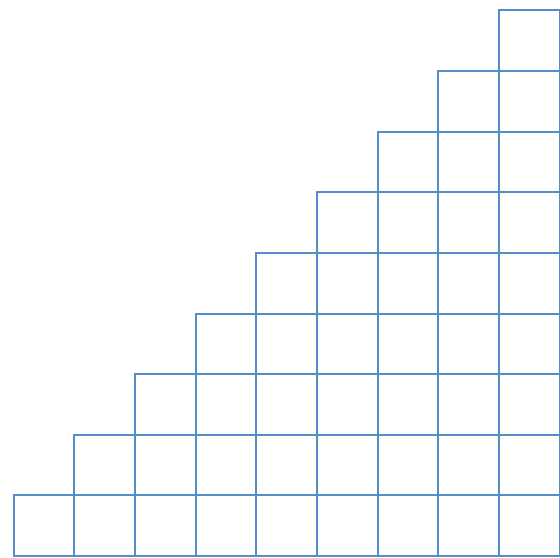




Curso

Power BI

Apostila



FABRIDATA

Rua Cosme Velho, 21 - Belo Horizonte-MG

CNPJ: 39.751.591/0001-74

www.fabridata.com.br

POWER BI:

prof. FABRICIO DOUGLAS VAZ

Sumário

Introdução	2
O que é o Power BI?	3
Características do Power BI Desktop	4
Conhecendo o menu do Power BI Desktop.	6
Um painel rápido em Power BI.	7
Detalhes da publicação de um Painel.	11
<i>Bussiness Intelligence</i> - Inteligência de Negócio.	14
Por que sim e por que não o gráfico de Setores (Pizza)?	16
Visualização de dados.	19
Obtenção de dados	30
Power Query (Consultas) e a linguagem M	39
Transformando dados.	40
Cruzando dados	41
Introdução a linguagem M	46
Criando medidas, colunas e tabelas no Power BI.	53
Introdução a linguagem Dax	55
Trabalhando com o DAX.	63

Introdução

Esta apostila tem o objetivo principal de servir como um guia para o curso de Power BI básico. Se trata de um compilado com adaptações e algumas anotações da documentação oficial do fabricante. Além de conter uma visão geral do Power BI, também pode servir como um ponto de referência para recordar os conteúdos discutidos em sala de aula.

No curso básico do Excel aprendemos os três pilares (como calcular - a aritmética, a base da lógica computacional e o endereçamento no Excel). Por sua vez, no curso intermediário deve-se focar em estruturas de dados e endereçamentos estruturados para compor cálculos mais elaborados e, com isso, ser mais produtivo no uso da ferramenta. Os cursos de Excel são passos anteriores que reputo importante para entender o Power BI em sua completude. Afinal, o PowerBI tem muitas ferramentas que estavam e estão presentes no Excel. Como o Power Query, Power Pivot etc.

Entretanto, há quem defenda que para estudar o Power BI não é necessário saber Excel. Isso é verdade! Mas, caso a pessoa tenha conhecimentos prévios em Excel, provavelmente, terá um entendimento acelerado no PowerBI. Visto que, muitos conceitos importantes usados no Power BI são discutidos nos cursos de Excel. Neste pensamento que este curso foi elaborado. A ideia é fazer um curso independente, que facilite o entendimento desta nova ferramenta criada em 2015 – Power BI – para ampliar os horizontes organizacionais na análise de dados, principalmente com grande quantidade de dados.

Mas, o que é uma grande quantidade de dados, ou o também falado Big Data? Como o poder computacional que temos em mãos atualmente e o Power BI é possível realizar análises com centenas de bilhões de dados no computador da sua casa, dependendo do licenciamento adquirido. O que hoje é considerado Big Data, no amanhã isso poderá ser feito no seu *smartphone*. O fato é que uma ferramenta para analisar esses dados é fundamental. Principalmente, é importante ter uma ferramenta que o não especialista em análise de dados possa utilizar com facilidade.

Importante recordar e precisamos ter em mente que o processamento de dados pode ter vários propósitos. Um dos propósitos do processamento de dados é a transformação dos dados para fins de **análise de dados**. Aprenderemos que o Power Query é uma boa ferramenta para isso e que faz parte do PowerBI. Recordando, a **analisar de dados** é o ato de se verificar, observar, interpretar ou entender os dados, por partes ou de modo generalizado, para se encontrar padrões novos e úteis. Alguns desses padrões nos dados estão mais visíveis a partir de como os dados são apresentados e resumidos.

Sendo assim, vamos ter a oportunidade de ver neste curso que os vários conceitos sobre a análise de dados. Utilizar métodos sistemáticos para tornar o Power BI uma ferramenta fácil e útil, que trará mais recursos para o desenrolar das análises de dados no dia a dia de cada um. Por exemplo, vamos recordar e aprender o conceito de tabela e a possibilidade de se construir gráficos rapidamente no Power BI. Muito provavelmente o Power BI seja a ferramenta número em visualização de dados, certamente da Microsoft, na atualidade.

O que é o Power BI?

O Power BI é uma plataforma de software, lançada em 2015, com aplicativos e conectores que trabalham juntos para transformar dados de várias fontes em insights para o negócio.

Os dados são carregados no Power BI de planilhas do Excel ou de uma coleção de Data Warehouses baseados em nuvem. O Power BI permite que você se conecte à diversas fontes de dados, crie visualizações e resumos de um modo bem intuitivo. Traz à tona o conceito de self-service BI, que trata da possibilidade de envolvidos no negócio, não especialistas em Tecnologia da Informação e Comunicações (TIC), realizarem suas próprias análises sem necessidade de pessoal técnico todo o tempo auxiliando.

No Power BI existem vários componentes que funcionam juntos, por exemplos os três básicos:

- a. **Power BI Desktop:** Um aplicativo de área de trabalho do Windows para construção das visualizações.
- b. **Power BI Service:** Um serviço SaaS (Software as a Service) online para visualizações.
- c. **Aplicativo móvel do Power BI** para dispositivos Windows, iOS e Android.

Esses três elementos – Power BI Desktop, o Power BI Service e os aplicativos móveis – foram projetados para permitir se criar, compartilhar e consumir informações para gerar insights para os negócios.

Além desses três, o Power BI também apresenta dois outros componentes:

- a. **Power BI Report Builder**, para criar relatórios paginados para compartilhar no serviço do Power BI.
- b. **Power BI Report Server**, um servidor de relatórios local onde você pode publicar seus relatórios do Power BI, depois de criá-los no Power BI Desktop (publicação de painéis on-premise).

O fluxo de trabalho comum no Power BI começa conectando-se a fontes de dados no Power BI Desktop e criando um relatório. Em seguida, você publica esse relatório do Power BI Desktop direto no Power BI Service e o compartilha para que os usuários do Power BI e em dispositivos móveis possam visualizar e interagir com o relatório.

Sendo assim, os caminhos comuns para análise de dados no Power BI Desktop:

1. Conectar-se aos dados.
2. Transformar e limpar os dados
3. Criar modelos de dados, como tabelas e seus relacionamentos.
4. Criar elementos visuais, ou gráficos, e resumos que forneçam representações visuais dos dados.

5. Criar painéis, que são compilações desses elementos visuais, em uma ou mais páginas de painéis.
6. Compartilhar os painéis com outras pessoas usando o Power BI Service.

Portanto, o Power BI é uma plataforma de serviços de análise de dados da Microsoft com o objetivo de fornecer visualizações interativas e recursos de *business intelligence*. Ele dá poder aos usuários finais de criarem os seus próprios relatórios e painéis.

Fluxo de Funcionamento do Power Bi

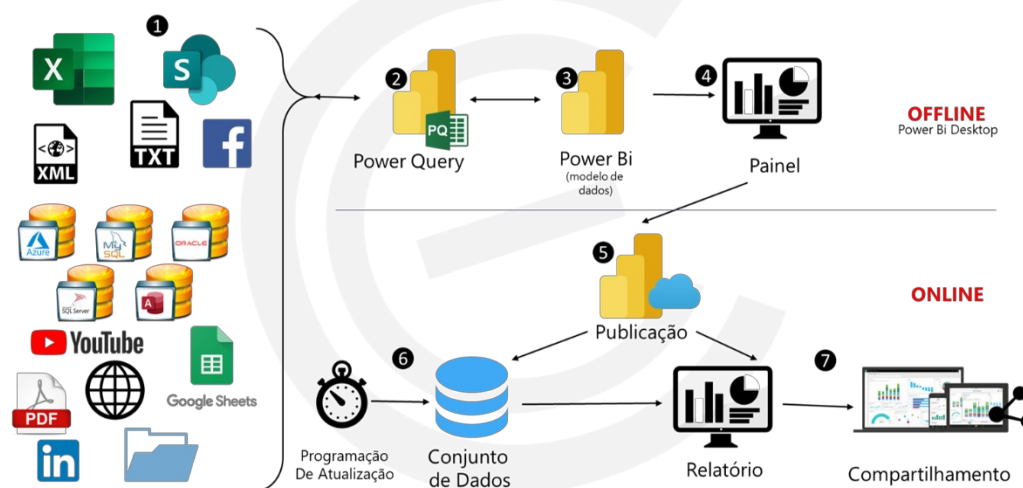


Figura 1¹

Características do Power BI Desktop

O que um usuário pode fazer no serviço do Power BI depende do tipo de *licença por usuário* que ele tem (gratuita, Pro, Premium por Usuário etc.). O nível de acesso é fornecido pela licença do usuário. Todos os usuários do serviço do Power BI precisam ter uma licença.

	Power BI Pro	Power BI Premium Por usuário	Power BI Premium Por capacidade
Colaboração e análise			
Acesso a aplicativos móveis	Incluído	Incluído	Incluído
Publique relatórios para compartilhar e colaborar	Incluído	Incluído	Incluído

¹ <https://www.consultoriaexcelencia.com.br/dashboards.html> (acesso em 26/09/2022)

Power BI

RDL (relatórios paginados)		Incluído	Incluído
Consuma conteúdo sem uma licença por usuário			Incluído
Relatórios locais com o Servidor de Relatórios do Power BI			Incluído
Preparação, modelagem e visualização de dados			
Limite de tamanho de modelo	1 GB	100 GB	400 GB
Taxa de atualização	8/dia	48/dia	48/dia
Conecte-se a mais de 100 fontes de dados	Incluído	Incluído	Incluído
Criar relatórios e visualizações com o Power BI Desktop	Incluído	Incluído	Incluído
Insira APIs e controles	Incluído	Incluído	Incluído
Visuais de IA	Incluído	Incluído	Incluído
IA avançada (análise de texto, detecção de imagem, aprendizado de máquina automatizado)		Incluído	Incluído
Fluxos de dados (consulta direta, entidades vinculadas e computadas, mecanismo de computação aprimorado)		Incluído	Incluído
Criação de Data Mart		Incluído	Incluído
Governança e administração			
Segurança e criptografia dos dados	Incluído	Incluído	Incluído
Métricas para criação, consumo e publicação de conteúdo	Incluído	Incluído	Incluído
Gerenciamento do ciclo de vida de aplicativos		Incluído	Incluído
Gerenciamento de implantação em várias regiões			Incluído
Disponibilidade do complemento de dimensionamento automático			Incluído
Armazenamento máximo	10 GB/usuário	100 TB	100 TB

Figura 2²

Conhecendo o menu do Power BI Desktop.

O Power BI Desktop é um software baseado em desktop do Windows. A principal funcionalidade dele é projetar painéis interativos e publicar relatórios. Seu menu principal é composto basicamente pelo menu Arquivo, Página Inicial, Inserir, Modelagem, Exibição e Ajuda, como mostra a Figura 3.

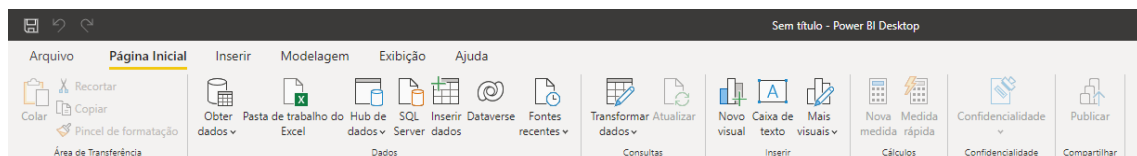


Figura 3

O menu Página Inicial traz uma coletânea básica das funcionalidades que são utilizadas com maior frequência no Power BI. Por exemplo obter dados ou adicionar um novo gráfico na tela, visto na Figura 3.

O menu Arquivos apresenta as opções para manipulação dos arquivos que armazenam os projetos e painéis do Power BI.



Figura 4

O menu inserir, por sua vez, aglutina as funcionalidades de inclusão de objetos nos painéis do Power BI. Por exemplo, pode-se adicionar uma nova página no painel ou uma imagem.

² <https://powerbi.microsoft.com/pt-br/pricing/> (acesso em 02/08/2022)



Figura 5

O menu Modelagem permite a criação de novas medidas, tabelas, parâmetros e funções. Muito útil no desenvolvimento dos cálculos e consolidações.

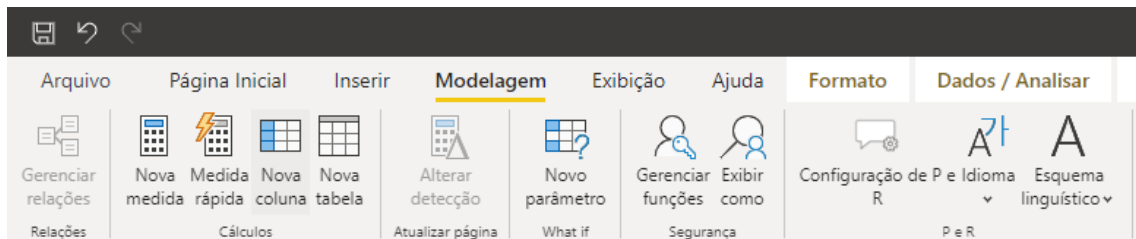


Figura 6

O menu Exibição permite alterar a apresentação das visualizações. Por exemplo cores e filtros, vide Figura 7.

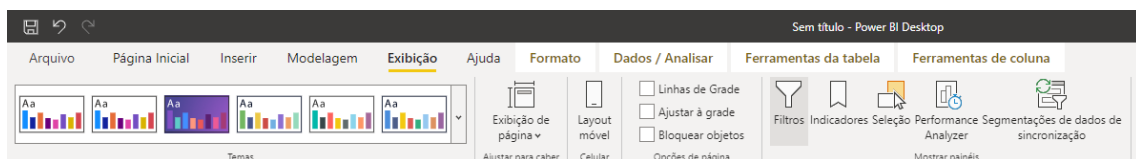


Figura 7

O menu Ajuda traz informações e possibilidades de acesso a conhecimentos relacionados ao Power BI na internet e outras funcionalidades para se interagir com a comunidade do Power BI.

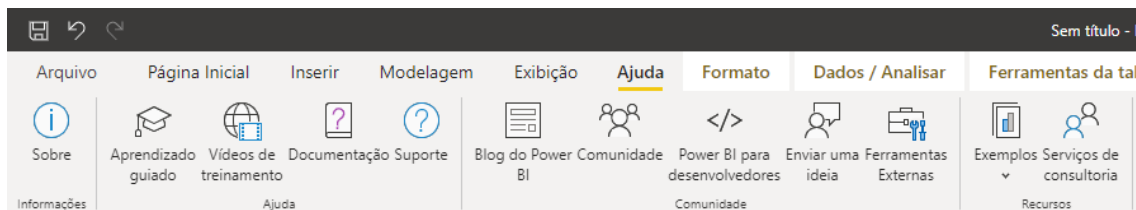


Figura 8

Um painel rápido em Power BI.

Para começar a usar o Power BI Desktop, a primeira etapa é conectar-se aos dados. Há muitas fontes de dados diferentes às quais você pode se conectar no Power BI Desktop. Na faixa inicial, selecione Obter Dados.

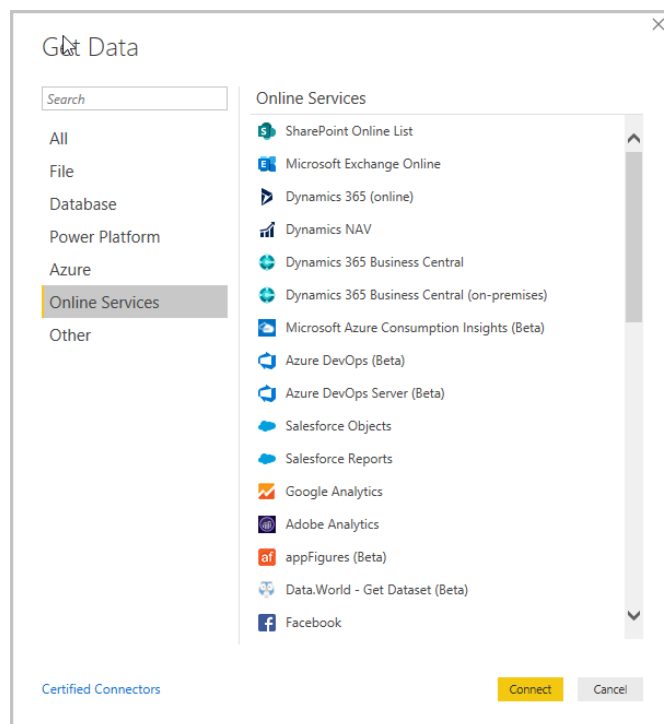


Figura 9

A janela Obter Dados é exibida, como da Figura 9, mostrando as várias categorias às quais o Power BI Desktop pode se conectar.

Quando você seleciona um tipo de dados, são solicitadas informações, como o caminho do arquivo no computador, a URL ou as credenciais, necessárias para que o Power BI Desktop se conecte à fonte de dados remota.

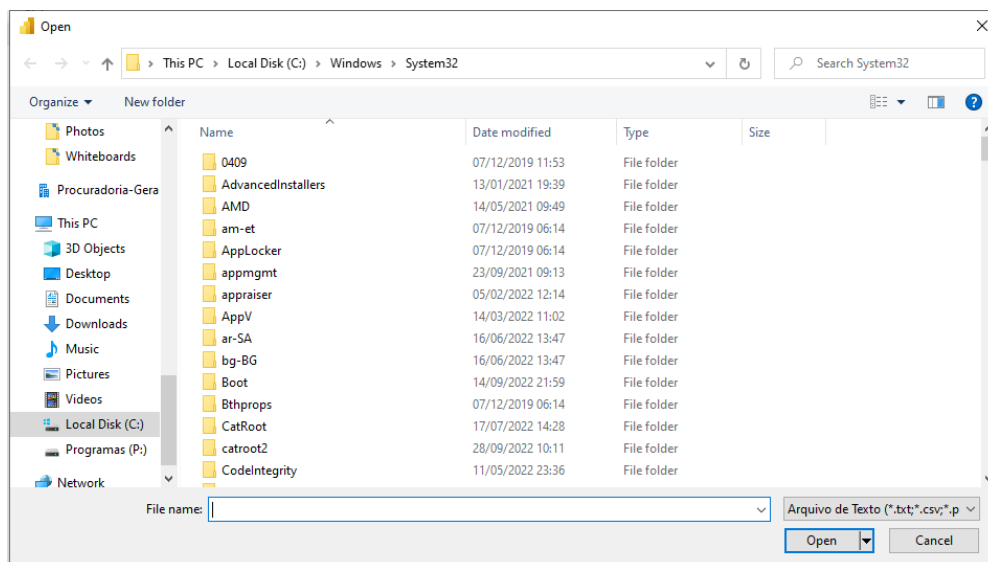


Figura 10

Após carregar os dados no Power BI Desktop, você pode limpar e transformar dados usando o Editor do Power Query integrado. Com o Power Query Editor, você faz alterações em seus dados, por exemplo alterar um tipo de dados, remover colunas ou combinar dados de várias fontes.

Sem título - Editor do Power Query

File | Página Inicial | Transformar | Adicionar Coluna | Exibição | Ferramentas | Ajuda

Fechar e Aplicar | Nova Fonte | Fontes Recentes | Inserir Dados | Configurações da fonte de dados | Gerenciar Parâmetros | Atualizar Visualização | Editor Avançado | Gerenciar | Escolher Colunas | Remover Colunas | Manter Linhas | Remover Linhas | Dividir Coluna | Agrupar por

Consultas [1]

20200206_Expulsoes

= Table.TransformColumnTypes(#"Cabeçalhos Promovidos",{{"NOME", type text}, {"CPF", type text}, {"MATRÍCULA", type text}, {"ÓRGÃO DE LOTAÇÃO", type text}})

	A ^B NOME	A ^B CPF	A ^B MATRÍCULA	A ^B ÓRGÃO DE LOTAÇÃO
1	FRANCISCO GUIMARAES NASCIMENTO	***.990.657-**	075****	INSS
2	OSVALDO LUCIANO DOS SANTOS LIMA	***.024.809-**	128****	UFPR
3	ROGERIO EMILIO DE ANDRADE	***.684.141-**	131****	AGU
4	JOAO BOSCO SANTANA SOARES	***.264.521-**	204****	INSS
5	NILENE CLEMENTE BARROS ALVES DE OLIVEI...	***.538.974-**	248****	UFCA
6	PAULO HENRIQUE GUERREIRO SCHAU	***.324.047-**	091****	SRF
7	HANDERSON LOUREIRO CAMELLO	***.979.147-**	010****	ME
8	ALEXANDRE ALBUQUERQUE GOMES DA SILVA	***.379.752-**	116****	UFAM
9	ANTONIO CARLOS CORREA	***.293.937-**	016****	ANTT
10	JOAO FRANCISCO NOGUEIRA EISENMANN	***.632.568-**	009****	SRF
11	JOSE SONCAO DOS SANTOS	***.770.251-**	017****	PR
12	MARCEL ANTONIO DE SALES	***.732.354-**	137****	ME
13	JULIANA GONCALVES ALEGRE	***.786.418-**	205****	UNIFESP
14	ELIZETE CABRAL NERI	***.762.524-**	008****	SRF
15	SANTO APARECIDO BARBIERI	***.939.029-**	012****	ME

Figura 11

Por fim é necessário verificar a estratégia de atualização dos dados. Basicamente o Power BI tem três modos de acesso ao armazenamento:

1. O modo mais comum é o modo Importa Data, quando os dados são lidos da fonte e são carregados dentro do arquivo .pbix. Se o dados precisam ser atualizados, por exemplo, se utiliza o Gateway de dados.
2. O segundo modo é o Direct Query, quando o dado é armazenado diretamente na fonte. Inclusive, os processamentos das consultas aos dados são feitos na origem. Por exemplo, um banco de dados SQL Server.
3. O próximo modo de armazenamento é o Live Connection que se conecta a cubos OLAP.

Figura 12³

³ <https://www.andersonmachado.eng.br/post/import-vs-direct-query-vs-live-connection?lang=en> (acesso em 10/11/2021)

Depois de obter os dados, transformar e carregar no PowerBI, pode-se arrastar visualizações para a tela do relatório para criar um belo relatório. Um relatório é uma representação gráfica dos dados combinado de modo a fazer sentido dentro de um contexto de negócio.

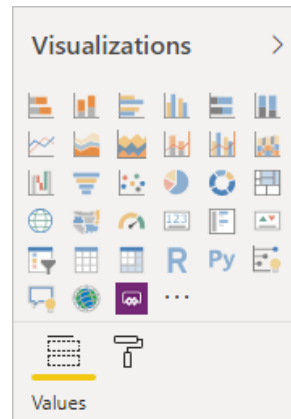


Figura 13

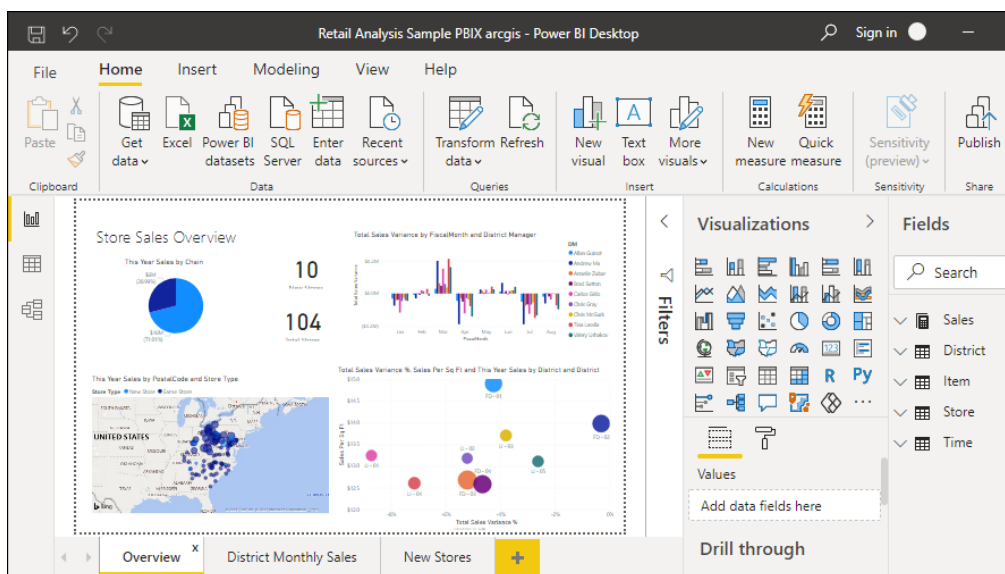


Figura 14

Depois que um relatório estiver pronto para ser compartilhado se poderá publicá-lo no Power BI.

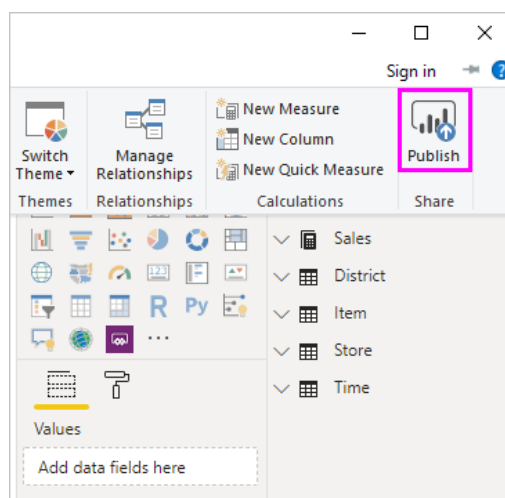


Figura 15

Detalhes da publicação de um Pannel.

No Power BI Desktop, na guia **Página Inicial**, clique ou toque em **Publicar**.

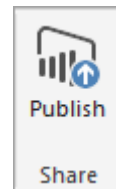


Figura 16

Se você ainda não tiver entrado no serviço do Power BI, insira uma conta e clique ou toque em Sign in.

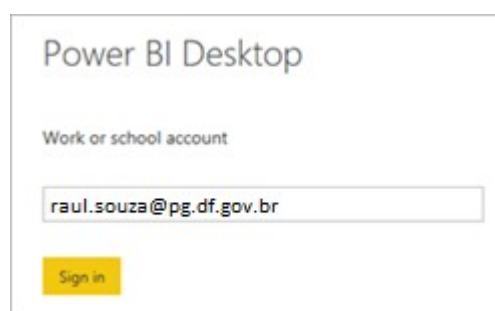


Figura 17

Insira uma senha e, em seguida, clique ou toque em Sign in.

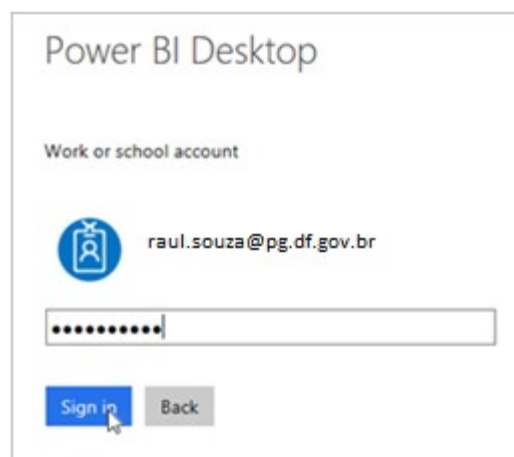


Figura 18

Escolha um destino para o relatório e, em seguida, clique ou toque em Select.

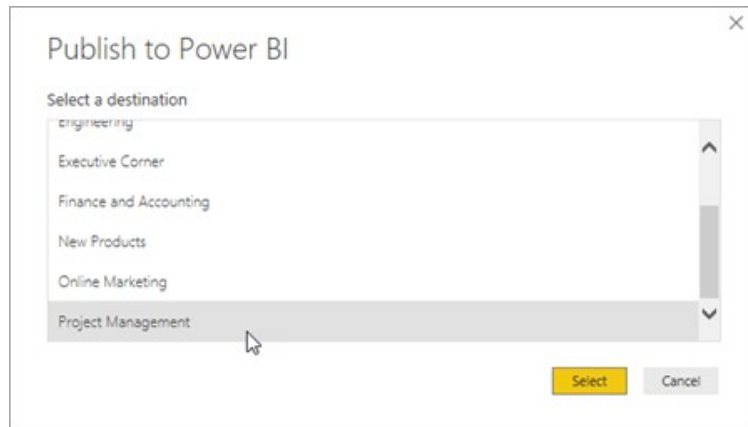


Figura 19

É recomendável publicar em um workspace para simplificar o acesso ao relatório no SharePoint.

Após a publicação, ser concluída, deve-se clicar em **Abrir 'nome_do_painel.pbix' no Power BI**.



Figura 20

O Power BI Service carrega o relatório em um navegador. Também é possível expandir o painel de navegação à esquerda que traz opções de navegação dos recursos disponíveis para o usuário no Power BI Service.

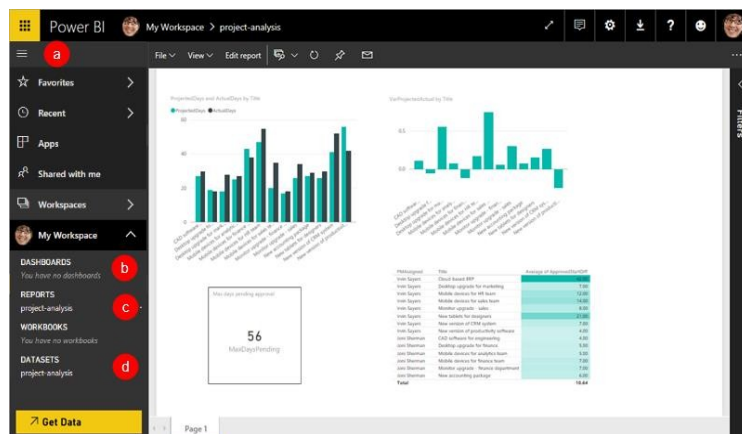



Figura 21

Pode-se observar que ao publicar no Power BI carregou-se um conjunto de dados e um relatório. No serviço, clicando em  no canto superior direito e, em seguida, clicando em Configurações, pode-se selecionar o Dataset – Conjunto de Dados – e lá se observar o Gateway connection e Schedule refresh, por exemplo. Neste momento pode-se utilizar um Gateway de Dados para atualizar as fontes de dados automaticamente ou uma conexão direta ao SharePoint.

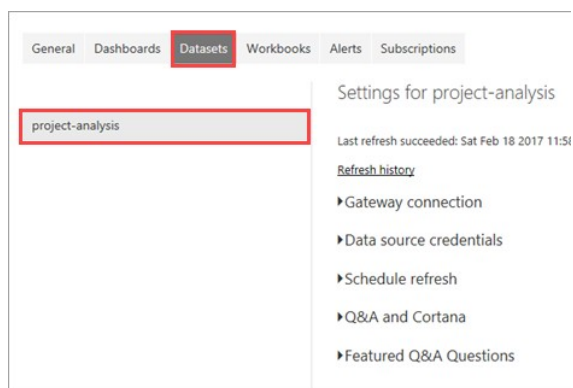


Figura 22

Bussiness Intelligence - Inteligência de Negócio.

Vamos dar um passo atrás para entender o que é o BI (Bussiness Intelligence) ou Inteligência de Negócio, que se trata da área de inteligência das organizações. Ou seja, o BI apoia a tomada de decisões baseadas em dados, gerando informações coletadas e exibidas por meio dos computadores para gerar *insight* para tomada de decisão.

O termo Business Intelligence (BI) apareceu pela primeira vez na literatura em 1865 utilizado por Richard Millar Devenís para descrever a história de um banqueiro Sir Henry Fernese que se valeu de informações de seus concorrentes para lucrar.

Em meados do ano de 1958 Hans Peter Luhn, pesquisador da IBM, publicou um artigo com sua ideia de BI. Já na década de 70 os chamados Sistemas de Apoio à Decisão (DSS – Decision Support System) chegaram para serem sistemas de informação interativos para ajudar os tomadores de decisões a utilizar dados e modelos para a tomada de decisão⁵ estratégica.

Na década de 80 surgiu a ideia da área de tecnologia juntamente com as áreas de negócio trabalharem juntas para construir informações para a alta gestão tomar suas decisões. Os sistemas de Planejamento de Recursos Empresariais (ERP – Enterprise Resource Planning) surgiram nesta mesma época, integrando os dados transacionais ou operacionais e estruturando os dados em Sistemas de Gerenciamento de Banco de Dados Relacional (RDBM – Relation Database Management), gerando relatórios mais interligados e holísticos. Nesta época, o analista Howard Dresner, trouxe a definição de BI como um conjunto de aplicativos, tecnologias e processos para coletar, armazenar, acessar e analisar dados, auxiliando e melhorando a tomada de decisão⁵.

Nos anos 80-90 surge o conceito de Data Warehouse (DW), por Bill Inmon, que se trata de uma camada intermediária, geralmente em SGBDs relacionais, utilizada como repositório de dados e invariavelmente tratamento como fonte única de dados para os relatórios negociais. Época que o famoso livro de Ralph Kimball foi publicado, introduzindo a modelagem dimensional. Um modelo que representa o DW está na Figura 23, onde os dados são coletados de diversas fontes empresariais e por meio de um processo ETL (*Extract, Transform e Load*) se constrói o DW.

Nos anos 2000 os sistemas de apoio à decisão baseados em DW começaram efetivamente a apoiar o BI. As áreas de dados começaram a surgir dentro das organizações, agregando mais atividades que o gerenciamento de bancos de dados. Termos como Data Marts, Data Mining, OLTP, OLAP, Planilhas Eletrônicas, dentre outras, se tornaram cotidianos dentro das organizações.

Quando chegou a década de 2010 o termo Big Data ganhou força e os serviços em nuvem começaram a galgar competitividade nas organizações. Os usuários começaram a trabalhar diretamente com os dados na fonte e os metadados começaram

⁴ <https://www.linkedin.com/pulse/business-intelligence-hist%C3%B3ria-rodri-go-textor-de-textor/?originalSubdomain=pt> (acessado em 10/08/2022).

⁵ [https://www.dbccompany.com.br/o-bi-ao-longo-da-historia/#:~:text=O%20termo%20Business%20Intelligence%20\(BI,lucrar%20diante%20de%20s-eus%20concorrentes](https://www.dbccompany.com.br/o-bi-ao-longo-da-historia/#:~:text=O%20termo%20Business%20Intelligence%20(BI,lucrar%20diante%20de%20s-eus%20concorrentes) (acessado em 10/08/2022).

a serem desenvolvidos para o negócio como um todo. Houve uma quebra de paradigma no modo como os dados são capturados e utilizados. As mídias para redes sociais e os serviços baseados em dados começaram a demonstrar que os dados seriam o “novo petróleo”. As empresas focadas em dados começam a alcançar patamares de empresas mais valiosas do mundo, por exemplo: Google, Facebook, Amazon etc.

Sendo assim, o BI é composto, basicamente, por três componentes: ETL, DW e ferramentas de visualizações. As ferramentas de visualizações são a camada final e muitas vezes os usuários as confundem com o BI em si. Porém, abaixo da linha da água existe uma grande complexidade. O Power BI, por sua vez, é uma ferramenta de visualização. No qual os relatórios são criados e o usuário pode, de modo facilitado, construir suas análises e seus próprios *dashboards*, *cockpits* e relatórios.

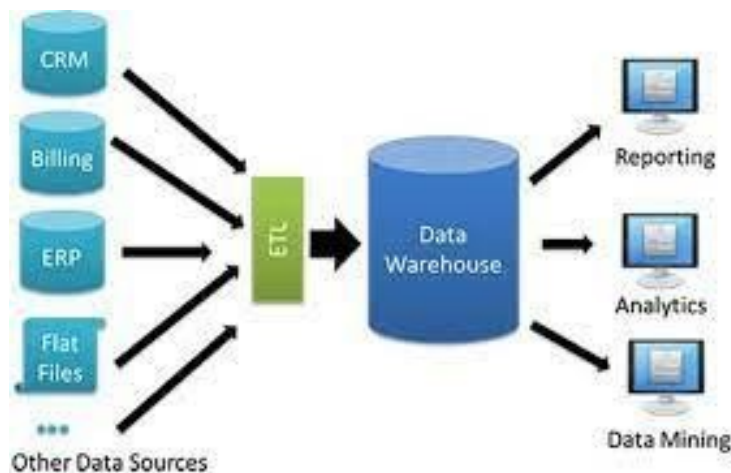


Figura 23⁶

⁶ <https://www.astera.com/pt/tipo/blog/defini%C3%A7%C3%A3o-de-data-warehouse/> (acesso em 26/09/2022)

Por que sim e por que não o gráfico de Setores (Pizza)?

TechCrunch Coverage: 2005 - 2011

A slightly better pie?

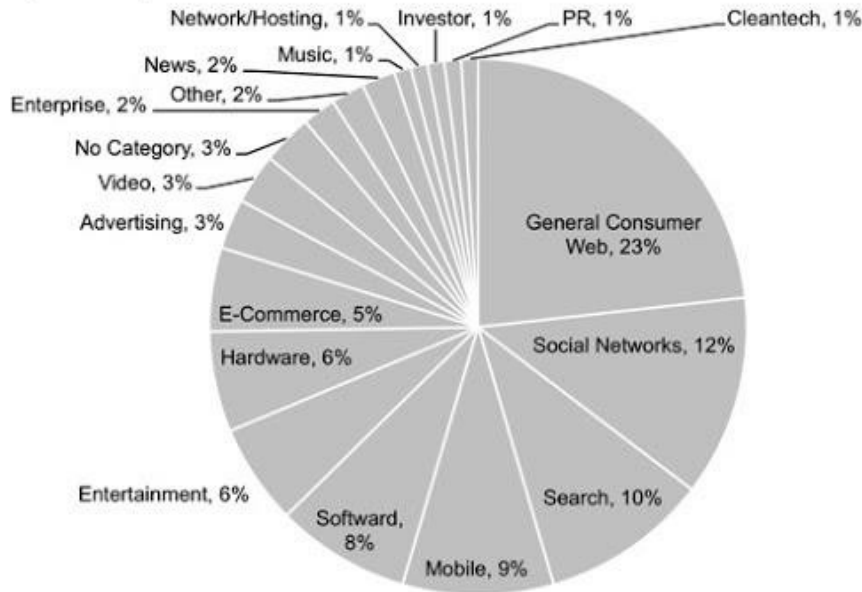


Figura 24⁷

Por que sim e por que não o gráfico de Setores (Pizza)? Para responder essa pergunta iremos explorar o artigo de Cole Nussbaumer Knaflitz. Sendo assim, inicia-se observando a Figura 24, onde vemos que quando se retira as cores de um gráfico de pizza ele perde muito o sentido, apesar de que as cores em gráficos de pizza geralmente não representam muita coisa no contexto de negócio. Geralmente as cores em gráficos de pizza são usadas para diferenciar as fatias ou setores. Para Colpeum efeito colateral não intencional em gráficos de pizza é a ilusão de ótica que se obtém com as fatias de cor mais escura, parecem maiores que as fatias do mesmo tamanho de uma cor mais clara.

Ainda Cole⁷ afirma que as cores devem ser sempre uma escolha explícita em gráficos para comunicação com o negócio e devem ser usadas estrategicamente para se chamar a atenção do público-alvo da visualização. Ou seja, o poder das cores não pode ser desperdiçado.

A pizza ou torta da Figura 24 é o que Excel retornaria, geralmente, quando se coloca os dados para gerar um gráfico automaticamente. Importante registrar que Cole orienta utilizar o Excel em primeira linha para representar graficamente os dados, o Excel seria a ferramenta inicial para o processo de design. Obviamente que o Power BI Desktop pode ser utilizado para essa prototipação, inclusive por conseguir operar com Big Data e com visualizações com mais facilidade.

⁷ <https://www.storytellingwithdata.com/blog/2011/07/death-to-pie-charts> (acesso em 10/09/2022)

Ao observar a Figura 24, outro problema encontrado, seria que os rótulos são confusos, pela quantidade e tamanho dos textos. Apenas algumas coisas são imediatamente aparentes: General Consumer Web é a maior parte. Há muitas pequenas fatias e isso é um dos grandes problemas do gráfico de setores/pizza.

Portanto, fica claro, analisando a Figura 24, que o principal problema com gráficos de pizza, em geral, é que os olhos não são bons para atribuir valor quantitativo a espaços bidimensionais. Quando os setores são próximos em tamanho fica difícil (se não impossível) dizer qual é o maior. Quando os setores não são de tamanho próximo, o melhor que você pode fazer é determinar que um é maior que o outro, mas você não pode julgar o tanto. Para superar isso, você pode adicionar rótulos de dados, mas ainda há um visual não claro quanto ao espaço que cada setor ocupa.

Uma solução para isso seria substituir um gráfico de setores (pizza) por um gráfico de barras horizontais, como o observado na Figura 25, organizando as grandezas da maior para a menor ou vice-versa. Com gráficos de barras, os olhos comparam grandezas melhor. Como eles estão alinhados em uma linha de base comum, é muito fácil avaliar o tamanho relativo. Isso facilita ver não apenas qual segmento é o maior (por exemplo), mas também quão incrementalmente ele é maior do que os outros segmentos.

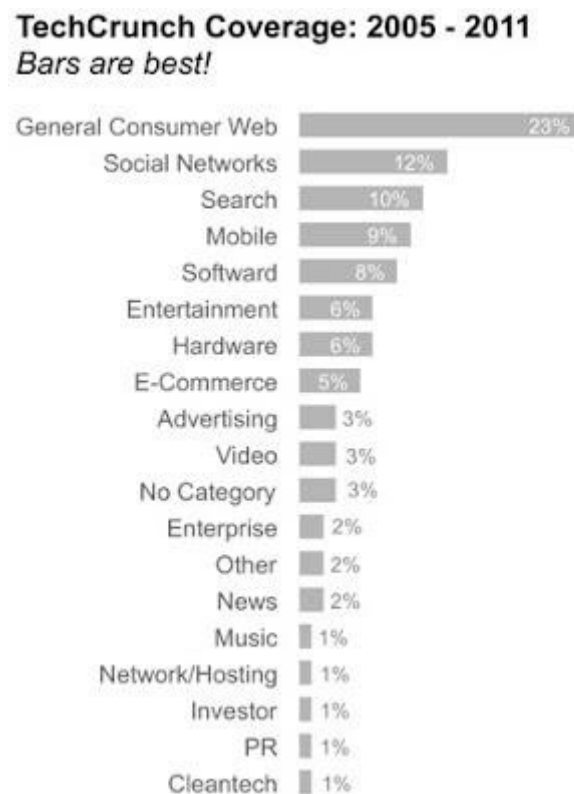


Figura 25

A única coisa que você obtém com um gráfico de pizza que está ausente em um gráfico de barras é o conceito de haver um todo e, portanto, partes de um todo. Mas, não é abominar o gráfico de setores, a ideia principal é demonstrar que não se pode utilizar um tipo de gráfico para tudo.

Como observa Cole, o gráfico de pizza nas organizações é um exemplo do fato de que alguns elegem um gráfico como o preferido e seguem suas análises apenas com

um tipo de gráfico. Cada gráfico tem seu propósito e o designer do painel deve ficar atento para qual informação deve ser entregue. Qual a pergunta de negócio o painel deve responder? O hábito de se utilizar o mesmo gráfico para tudo gera uma falsa impressão de que todos os tipos de gráficos trazem a mesma informação. Porém, dependendo do público-alvo um tipo de gráfico não pode ser usado e em outro público sim, até mesmo o gráfico de setores(pizza). O que importa é uma preocupação com a arquitetura da informação. Ter em mente que os dados são como objetos e que organizados de uma determinada forma trazem um artefato de informação específico.

Em frente a

uma bela casa,

uma bela árvore.

Ninguém viu nada.

Ninguém soube de nada.

Mas ela se foi...

Em frente a

uma bela árvore,

uma bela casa.

Ninguém viu nada.

Ninguém soube de nada.

Mas ela se foi...

Visualização de dados.

Sendo uma ferramenta de visualização, os gráficos e resumos são a principal funcionalidade que o Power BI disponibiliza. O layout do Power BI é focado para se criar relatórios e painéis. A seguir iremos verificar sucintamente algumas utilidades dos gráficos que o Power BI disponibiliza para se montar as visualizações.

Proporção

Gráficos de Setores (Pizza ou Torta)

Os gráficos de Setores (Pizza ou Torta), Figura 26, deve ser o gráfico número um nas apresentações empresariais na atualidade. Ele mostra a relação das partes com um todo.

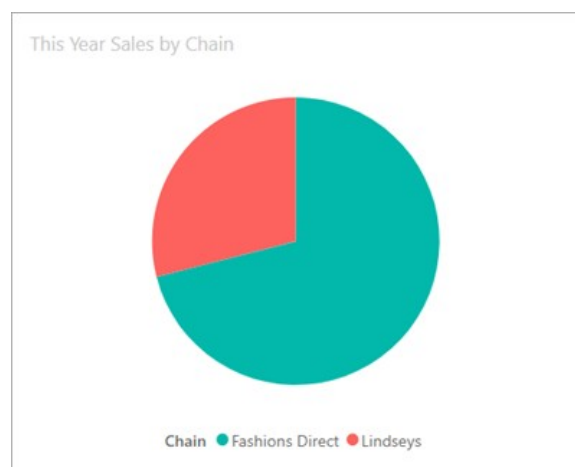


Figura 26

Gráficos de rosca

Os gráficos de rosca são semelhantes aos gráficos de pizza. Eles mostram a relação das partes com um todo. A única diferença com o gráfico de pizza é que o centro está em branco e permite mais espaço para rótulos.



Figura 27

Treemaps

Os gráficos de Treemaps ainda são gráficos de proporção, são formados por retângulos coloridos, com um tamanho que representa o valor proporcional. Eles podem ser hierárquicos, com retângulos aninhados nos retângulos principais. O espaço dentro de cada retângulo é alocado com base no valor que está sendo medido da parte para o todo. E os retângulos são organizados no tamanho da parte superior esquerda (maior) à parte inferior direita (menor).

Os treemaps são uma ótima opção:

- Para exibir grandes quantidades de dados hierárquicos.
- Quando um gráfico de barras não puder lidar efetivamente com grande número de valores.
- Para mostrar as proporções entre cada parte e o todo.
- Para mostrar o padrão da distribuição da medida em cada nível das categorias na hierarquia.
- Para mostrar atributos usando a codificação de cor e tamanho.
- Para identificar padrões, exceções, colaboradores mais importantes e exceções.



Figura 28

Comparar grandezas

Gráficos de barras e colunas

Os gráficos de barras ou colunas, Figura 29 e Figura 30, servem para se apresentar a variação de um valor específico entre entes diferentes medições.

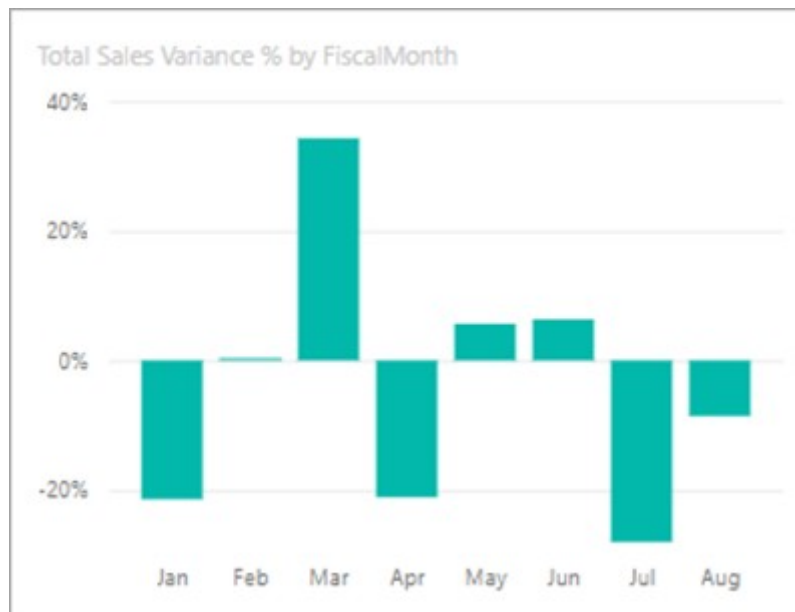


Figura 29

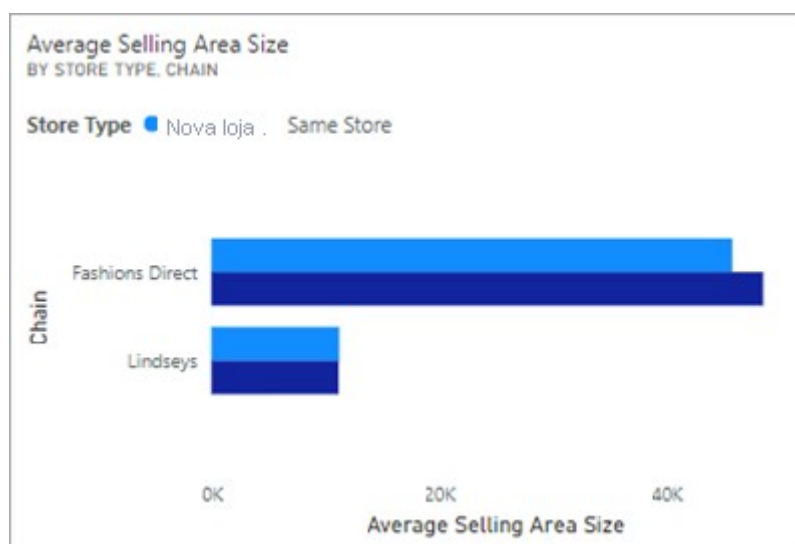


Figura 30

Tendências

Gráficos de linhas

Os gráficos de linhas enfatizam o formato geral de uma série inteira de valores, geralmente ao longo do tempo.



Figura 31

Gráficos de área ou integral

Um gráfico de área básico baseia-se no gráfico de linhas com a área entre o eixo e a linha preenchida. Os gráficos de área enfatizam a magnitude da alteração ao longo do tempo e pode ser usado para chamar a atenção para o valor total entre uma tendência ou uma distribuição de frequência. Por exemplo, os dados que representam os ganhos ao longo do tempo podem ser apresentados para enfatizar o montante final.

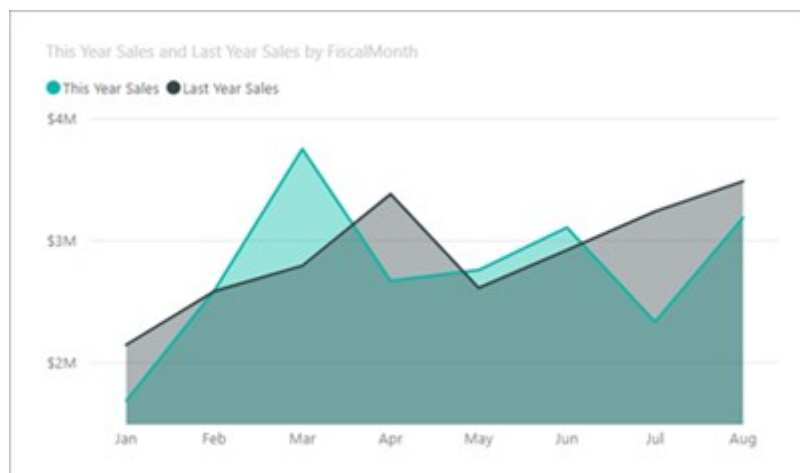


Figura 32

Gráfico combinado

Um gráfico combinado é um gráfico de colunas e um gráfico de linhas juntos. Combinar os dois gráficos permite que se faça uma comparação rápida dos dados. Gráficos de combinação podem ter um ou dois eixos Y, portanto, não deixe de examiná-los cuidadosamente.



Figura 33

Os gráficos de combinação são uma ótima opção:

- Quando você tem um gráfico de linhas e um gráfico de colunas com o mesmo eixo X.
- Para comparar várias medidas com intervalos de valores diferentes.
- Para ilustrar a correlação entre duas medidas em um visual.
- Para verificar se uma medida atende ao alvo definido pela outra medida.
- Para conservar o espaço de tela.

Medidas

Gráficos de medidor



Figura 34

Um gráfico de medidor radial tem um arco circular e exibe um único valor que acompanha o progresso em relação a um objetivo. A meta, ou o valor de destino, é representada pela linha (agulha). Geralmente, o progresso em relação a esse objetivo é representado pela parte mais clara e o valor que representa o progresso é representado com a parte mais escura dentro do arco. Todos os valores possíveis são distribuídos uniformemente ao longo do arco, do mínimo (valor mais à esquerda) para o máximo (valor mais à direita).

Os medidores radiais são uma ótima opção para:

- Mostrar o progresso para atingir uma meta.
- Mostrar a integridade de uma única medida.
- Exiba informações que podem ser examinadas e compreendidas rapidamente.

KPIs

Um KPI (Indicador Chave de Desempenho) é uma indicação visual que comunica a quantidade de progresso feito em relação a uma meta mensurável.

Os KPIs são uma ótima opção:

- Para medir o progresso (no que estou adiantado ou atrasado?).
- Para medir a distância para uma métrica (o quão adiantado ou atrasado eu estou?).



Figura 35

Cartões múltiplas linhas

Cartões de múltiplas linhas exibem um ou mais pontos de dados, um por linha.



Figura 36

Cartões número único

Cartões de número único exibem um único fato, um único ponto de dados. Às vezes, um único número é a coisa mais importante que você deseja acompanhar no seu painel ou relatório do Power BI.



Figura 37

Tabelas

Uma tabela é uma grade que contém dados relacionados em linhas e colunas. Ela também pode conter cabeçalhos e linhas de totais. As tabelas funcionam bem com comparações quantitativas em que você observa muitos valores de uma única categoria. As tabelas são uma ótima opção:

- Para ver e comparar dados detalhados e valores exatos (em vez de representações visuais).
- Para exibir dados em um formato tabular.
- Para exibir dados numéricos por categorias.

Category	This Year Sales Status	Average Unit Price	Last Year Sales	This Year Sales	This Year Sales Goal
080-Accessories	●	\$4.84	\$1,273,096	\$1,379,259	\$1,273,096
090-Home	●	\$3.93	\$2,913,647	\$3,053,326	\$2,913,647
100-Groceries	●	\$1.47	\$810,176	\$829,776	\$810,176
020-Mens	●	\$7.12	\$4,453,133	\$4,452,421	\$4,453,133
030-Kids	●	\$5.30	\$2,726,892	\$2,705,490	\$2,726,892
050-Shoes	●	\$13.84	\$3,640,471	\$3,574,900	\$3,640,471
010-Womens	●	\$7.30	\$2,680,662	\$1,787,958	\$2,680,662
040-Juniors	●	\$7.00	\$3,105,550	\$2,930,385	\$3,105,550
060-Intimate	●	\$4.28	\$955,370	\$852,329	\$955,370
070-Hosiery	●	\$3.69	\$573,604	\$486,106	\$573,604
Total	●	\$5.49	\$23,132,601	\$22,051,952	\$23,132,601

Figura 38

Matrizes

O visual de matriz é um tipo de visual de tabela que dá suporte a um layout em etapas. Uma tabela dá suporte a duas dimensões, mas uma matriz facilita significativamente a exibição dos dados entre várias dimensões. A matriz agrega automaticamente os dados e permite fazer drill down nos dados.

Drill on Rows													
Region	Centre			East		West		Total					
Sales Stage	Opportunity Count	Revenue		Opportunity Count	Revenue	Opportunity Count	Revenue	Opportunity Count	Revenue				
Lead	102	\$307,574,417		114	\$473,887,837	52	\$236,159,114	268	\$1,237,621,368				
Qualified	29	\$111,715,461		30	\$186,692,134	13	\$52,442,343	74	\$350,849,938				
Solution	29	\$100,743,788		39	\$134,347,170	13	\$33,441,301	74	\$268,532,259				
Proposal	14	\$46,722,869		13	\$39,970,804	10	\$43,032,669	37	\$149,726,342				
Finalize	5	\$13,332,248		5	\$30,696,425	4	\$21,176,183	14	\$75,174,859				
Total	179	\$790,058,782		212	\$884,594,513	96	\$426,251,832	487	\$2,110,905,127				

Figura 39

Segmentações



Figura 40

Uma segmentação de dados é um gráfico autônomo que pode ser usado para filtrar os outros visuais na página. As segmentações de dados são fornecidas em vários formatos diferentes (categoria, intervalo, data etc.) e podem ser formatadas para permitir a seleção de apenas um, muitos ou todos os valores disponíveis.

As segmentações de dados são uma ótima opção para:

- Exibir os filtros mais usados ou importantes na tela do relatório para facilitar o acesso.
- Facilitar a exibição do estado atual filtrado sem precisar abrir uma lista suspensa.
- Filtrar por colunas desnecessárias e ocultas nas tabelas de dados.
- Criar relatórios mais direcionados colocando a segmentação de dados ao lado de visuais importantes.

Dispersão

Gráfico de dispersão, de bolhas e de pontos



Figura 41

Um gráfico de dispersão sempre tem dois eixos de valor para mostrar um conjunto de dados numéricos em um eixo horizontal e outro conjunto de valores numéricos em um eixo vertical. O gráfico exibe pontos na interseção de um valor numérico de x e y, combinando esses valores em pontos de dados individuais. Esses pontos de dados podem ser distribuídos de maneira uniforme ou não pelo eixo horizontal, dependendo dos dados.

Um gráfico de bolhas substitui os pontos de dados por bolhas, com o tamanho de bolha representando uma dimensão adicional dos dados.

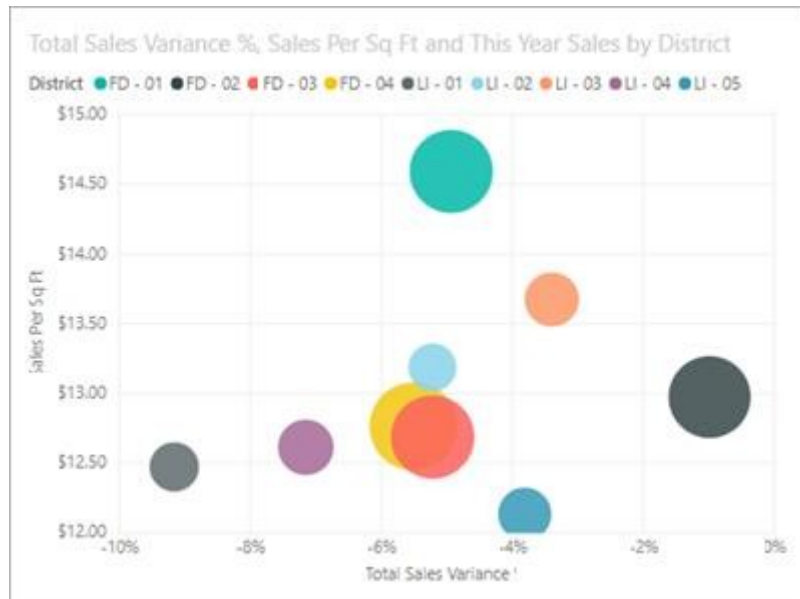


Figura 42

Um gráfico de pontos é semelhante a um gráfico de bolhas e um gráfico de dispersão, mas pode plotar dados numéricos ou categóricos ao longo do eixo X. Este exemplo usa quadrados, em vez de círculos, e plota vendas ao longo do eixo X.

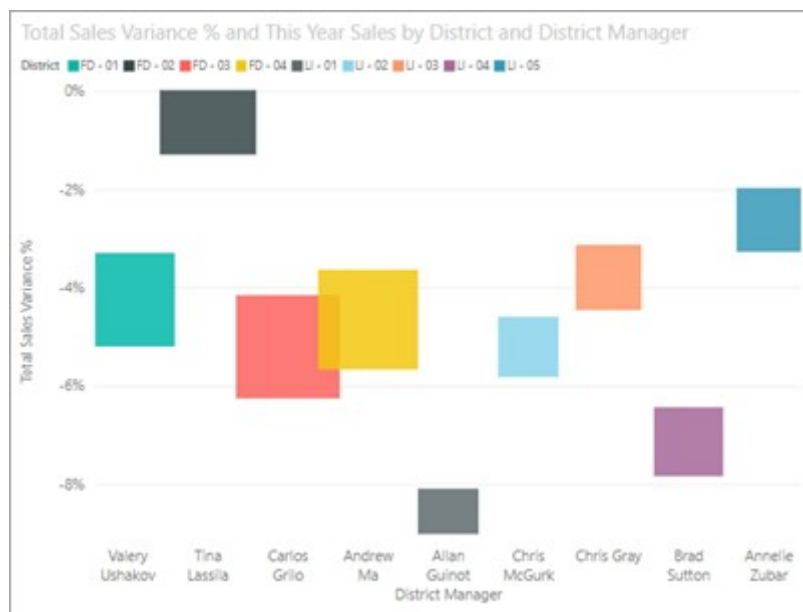


Figura 43

Dispersão de alta densidade

Por definição, os dados de alta densidade são amostrados para criar com uma rapidez razoável visuais que atendam à interatividade. A amostragem de alta densidade usa um algoritmo que elimina os pontos sobrepostos e garante que todos os pontos no conjunto de dados sejam representados no visual. Ele não apenas plota uma amostra representativa dos dados.



Figura 44

Isso assegura a melhor combinação de capacidade de resposta, representação e preservação clara de pontos importantes no conjunto de dados geral.

Georreferenciamento

Mapa básico

Use um mapa básico para associar informações categóricas e quantitativas a locais espaciais.

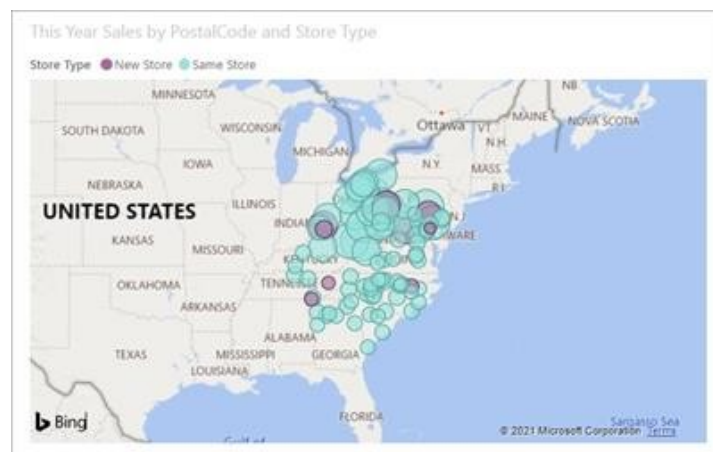


Figura 45

Mapa preenchido (coroplético)



Figura 46

Um mapa coroplético usa sombreamento ou tonalidade ou padrões para exibir como um valor na proporção em uma localização geográfica ou região. Exibe rapidamente diferenças relativas com sombreamento que varia de claro (menos frequente/inferior) para escuro (mais frequente/mais). Quanto mais intensa a cor, maior o valor.

Mapa de formas

Mapas de formas comparam regiões em um mapa usando cores. Um mapa de formas não pode mostrar locais geográficos precisos dos pontos de dados em um mapa. Em vez disso, sua finalidade principal é mostrar comparações relativas de regiões em um mapa colorindo-as de modo diferente.

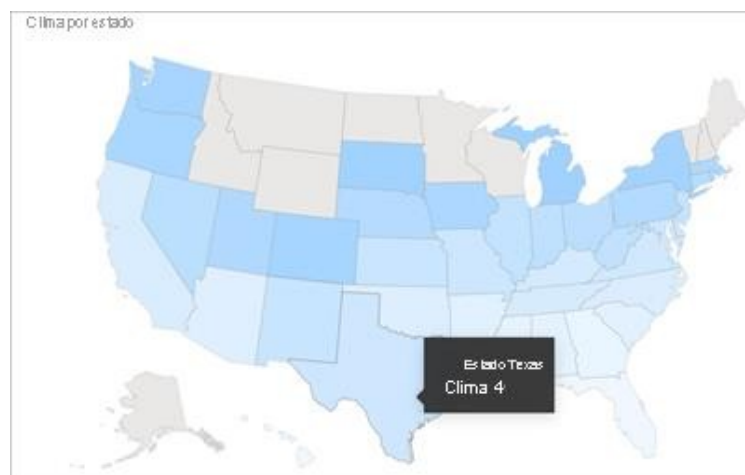


Figura 47

Obtenção de dados

Como o Power BI é possível a obtenção de dados de diversas fontes. Vamos observar as Figura 48 e Figura 49, e perceber que o caminho padrão para obter dados externos ao Power BI é praticamente o mesmo. Clica-se no botão Obter dados na página inicial e em seguida no tipo de fontes de dados desejada: Arquivo, Banco de Dados, Serviços Online etc.

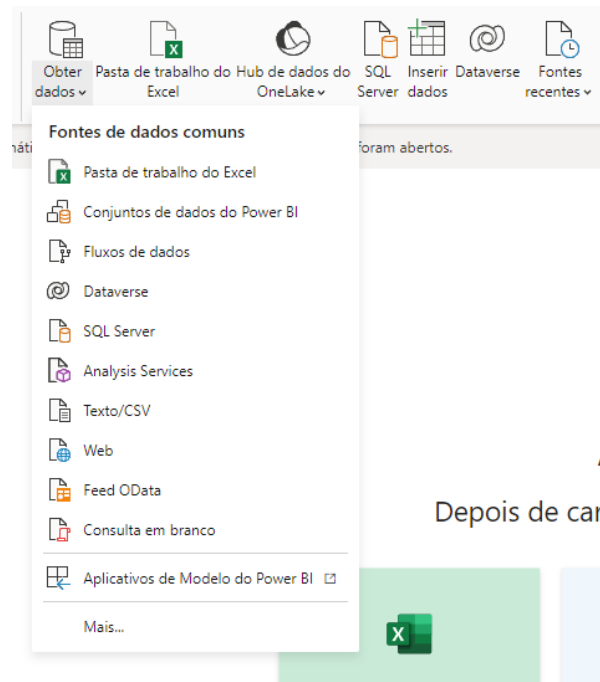


Figura 48

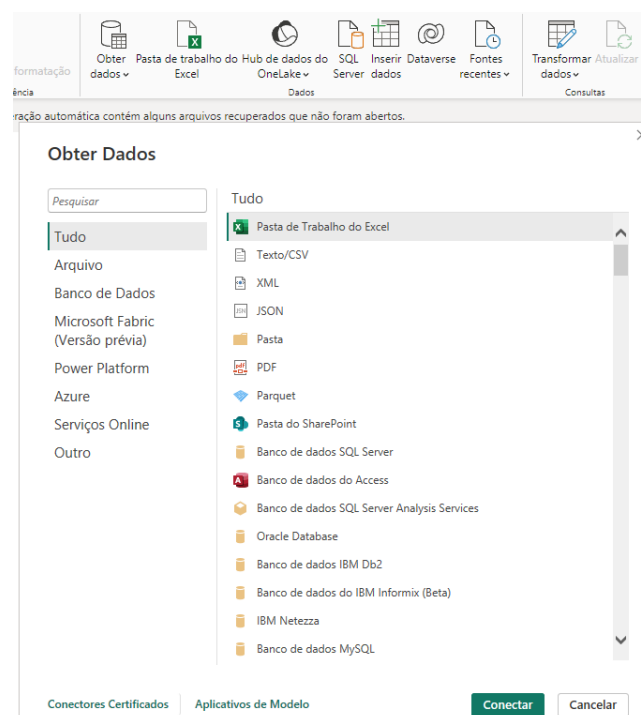


Figura 49

O Power BI possibilita a obtenção de dados de diversos formatos. O arquivo CSV e os delimitados por espaço ou símbolo são formatos muito comuns para troca de dados. Talvez esses formatos sejam os mais utilizados para troca de dados desde os primórdios da computação. A sigla CSV significa *Comma Separated Values*, ou seja, valores separados por vírgula. O que é interessante é que os arquivos com a extensão CSV não vem muitas vezes como o delimitador do tipo virgula, podem vir com outros símbolos. Talvez, por isso, o Power BI agrupa os arquivos txt (texto)/CSV em uma única opção.

Pode-se obter dados de um CSV, arquivo Excel, Banco de Dados Access pelo mesmo caminho. A Figura 50 demonstra como é carregado um arquivo CSV para uma tabela dentro do Power BI.

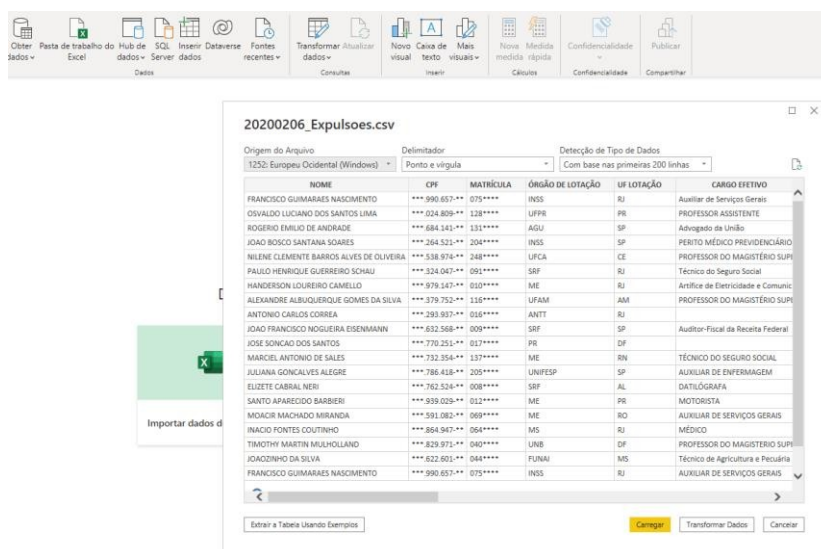


Figura 50

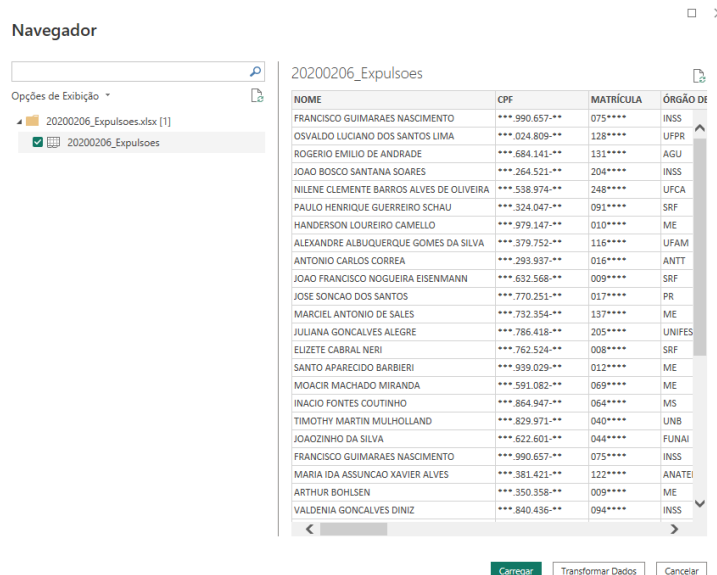


Figura 51

A Figura 37 demonstra como é similar uma importação de um arquivo CSV e um arquivo do Excel (.xlsx). Um detalhe importante a notar é que todas as duas fontes são do tipo tabular.

Para os bancos de dados Access a importação não muda muito, principalmente porque os dados em CSV, Excel e Bancos de dados Access são no formato de tabela. O formato tabular é o formato em duas dimensões, onde as colunas agrupam as observações de algum atributo de alguma entidade do mundo natural ou do mundo virtual. Por exemplo, o nome de um servidor público é um atributo e cada servidor tem o seu. Veja a Figura 52 e observe que a única diferença na importação dos dados do banco de dados Access é que temos mais de uma tabela que pode ser importada.

Navegador

Opções de Exibição ▾

- ✓ Servidores_Compras.accdb [3]
 - Compras Consulta
 - Compras
 - Servidores**

Servidores

CPF	NOME	ÓRGÃO DE LOTAÇÃO
990.657-11	FRANCISCO GUIMARAES NASCIMENTO	INSS
024.809-11	OSVALDO LUCIANO DOS SANTOS LIMA	UFPR
684.141-11	ROGERIO EMILIO DE ANDRADE	AGU
264.521-11	JOAO BOSCO SANTANA SOARES	INSS
538.974-11	NILENE CLEMENTE BARROS ALVES DE OLIVEIRA	UFPA
324.047-11	PAULO HENRIQUE GUERREIRO SCHAU	SRF
979.147-11	HANDERSON LOUREIRO CAMELO	ME
379.752-11	ALEXANDRE ALBUQUERQUE GOMES DA SILVA	UFAM
299.937-11	ANTONIO CARLOS CORREA	ANTT
632.568-11	JOAO FRANCISCO NOGUEIRA EISENMANN	SRF
770.251-11	JOSE SONCAO DOS SANTOS	PR
732.354-11	MARCEL ANTONIO DE SALES	ME
786.418-11	JULIANA GONCALVES ALEGRE	UNIFESP
762.524-11	ELIZETE CABRAL NERI	SRF
939.029-11	SANTO APARECIDO BARBIERI	ME
591.082-11	MOACIR MACHADO MIRANDA	ME
864.947-11	INACIO FONTES COUTINHO	MS
829.971-11	TIMOTHY MARTIN MULHOLLAND	UNB
622.601-11	JOAOZINHO DA SILVA	FUNAI
990.657-11	FRANCISCO GUIMARAES NASCIMENTO	INSS
381.421-11	MARIA IDA ASSUNCAO XAVIER ALVES	ANATEL
350.358-11	ARTHUR BOHLSSEN	ME
840.436-11	VALDENIA GONCALVES DINIZ	INSS

Selecionar Tabelas Relacionadas

Carregar Transformar Dados Cancelar

Figura 52

Para explicar como importar o XML precisamos defini-lo primeiramente. O XML, significa *Extensible Markup Language* (Linguagem de Marcação Extensível). Ele é uma linguagem de marcação amplamente utilizada para representar e armazenar dados de forma estruturada e hierárquica. Foi projetada para ser legível por humanos e máquinas, o que a torna uma escolha popular para troca de dados.

Vejamos alguns conceitos elementares relacionados ao XML:

Tags em XML agrupam os dados e os organizam em elementos, que são delimitados pelos símbolos matemáticos: menor-que e o símbolo maior-que. Uma tag de abertura, como <nome>, indica o início de um elemento, e uma tag de fechamento como </nome> indica o fim do elemento. Os dados podem ser colocados entre as tags, os quais podem ser outras tags.

Exemplo:

```
< Pessoa >
  < nome > João < / nome >
  < idade > 30 < / idade >
< / Pessoa >
```

Atributos podem ser dados a cada elementos XML e podem conter informações adicionais sobre o elemento. Os atributos são definidos dentro da tag de abertura do elemento.

Exemplo:

```
<peessoa id="123">
  <nome>João</nome>
  <idade>30</idade>
</peessoa>
```

A hierarquia dos elementos XML pode ser feita por meio do aninhamentos. O que significa que um elemento pode conter outros elementos, criando uma estrutura de árvore. Importante destacar que essa é a grande diferença do XML para os dados tabulares, o XML tem uma estrutura de árvore.

Exemplo:

```
<biblioteca>
  <livro>
    <titulo>O Senhor dos Anéis</titulo>
    <autor>J.R.R. Tolkien</autor>
  </livro>
  <livro>
    <titulo>O Hobbit</titulo>
    <autor>J.R.R. Tolkien</autor>
  </livro>
</biblioteca>
```

A declaração XML é a demonstração que o arquivo é um documento XML, a qual define a versão do XML e pode ser usada opcionalmente para definir a codificação de caracteres.

Exemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Em resumo, o XML é uma linguagem de marcação que permite estruturar dados de maneira hierárquica e legível, tornando-o adequado para uma variedade de aplicações onde a organização estruturada dos dados é importante.

Quando se escolhe o formato XML na Figura 53 como fonte de dados devem ser feita uma série de análises e conversão pelo Power Query para se chegar em dados tabulares. Isso é feito principalmente porque o Power BI trabalha basicamente com tabelas e seus relacionamentos. Ou seja, o XML tem que ser convertido para dados para tabulares para ser importado no Power BI.

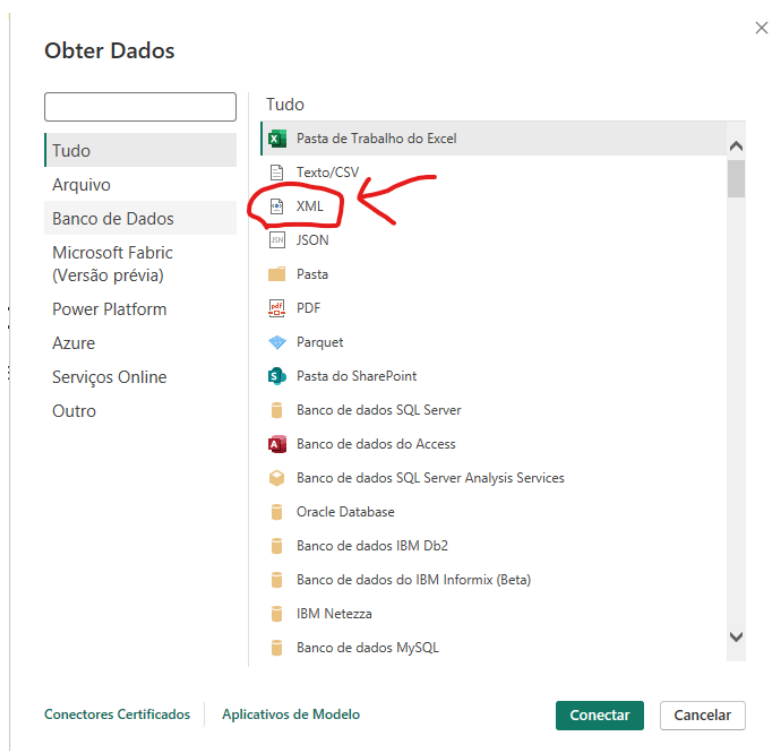


Figura 53

A Figura 54 demonstra que o Power Bi não reconhece os campos de dados diretamente no XML. Desse modo, é necessário transformar os dados. Ação que possível ao clicar em Transformar Dados, que irá abrir o Power Query.

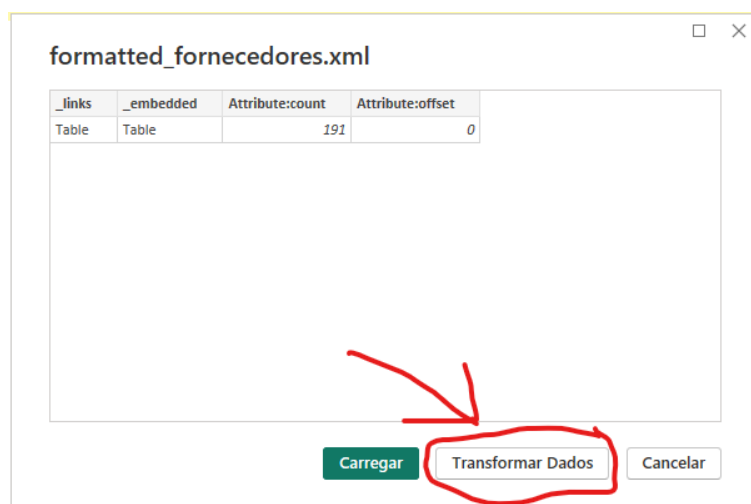


Figura 54

No caso dos dados que estamos tentando importar, temos que procurar a tag **resource**. Como pode ser visto na Figura 55, a tag **resource** é a que seria o equivalente a cada linha de nossa tabela alvo.

```
</resource>
<resource>
  <_links>
    <link href="/fornecedores/id/fornecedor_pj/00122327000136" rel="self"
    <link href="/fornecedores/id/municipio/81051" rel="municipio" title="
    <link href="/fornecedores/id/natureza_juridica/48" rel="natureza_juri
    <link href="/fornecedores/id/porte_empresa/2" rel="porte_empresa" tit
    <link href="/fornecedores/id/cnae/6436100" rel="cnae" title="CNAE 643
  </_links>
  <id>1228</id>
  <cnnpj>00122327000136</cnnpj>
  <nome>SANTINVEST S/A CREDITO FINANCIAMENTO E INVESTIMENTOS</nome>
  <ativo>true</ativo>
  <id_municipio>81051</id_municipio>
  <uf>SC</uf>
  <id_natureza_juridica>48</id_natureza_juridica>
  <id_porte_empresa>2</id_porte_empresa>
  <id_cnae>6436100</id_cnae>
  <habilitado_licitar>true</habilitado_licitar>
</resource>
<resource>
  <_links>
    <link href="/fornecedores/id/fornecedor_pj/00469296000194" rel="self"
    </_links>
    <link href="/fornecedores/id/municipio/60011" rel="municipio" title="
    <link href="/fornecedores/id/natureza_juridica/64" rel="natureza_juri
    <link href="/fornecedores/id/porte_empresa/2" rel="porte_empresa" tit
    <link href="/fornecedores/id/cnae/9420100" rel="cnae" title="CNAE 942
  </_links>
  <id>4842</id>
  <cnnpj>00469296000194</cnnpj>
  <nome>SINDICATO NACIONAL DOS TRABALHADORES NA PROTECAO AO VOO</nome>
  <ativo>true</ativo>
  <id_municipio>60011</id_municipio>
  <uf>RJ</uf>
  <id_natureza_juridica>64</id_natureza_juridica>
  <id_porte_empresa>2</id_porte_empresa>
  <id_cnae>9420100</id_cnae>
  <habilitado_licitar>true</habilitado_licitar>
</resource>
```

Figura 55

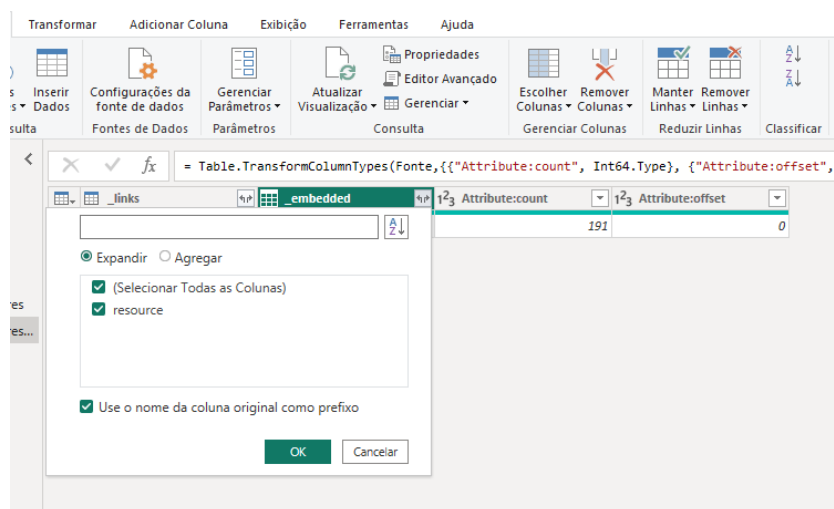


Figura 56

Sendo assim, é necessário expandir a tabela **resource**. O comando de expansão presente na Figura 56 resulta na identificação de cada linha de uma tabela como demonstra a Figura 57.

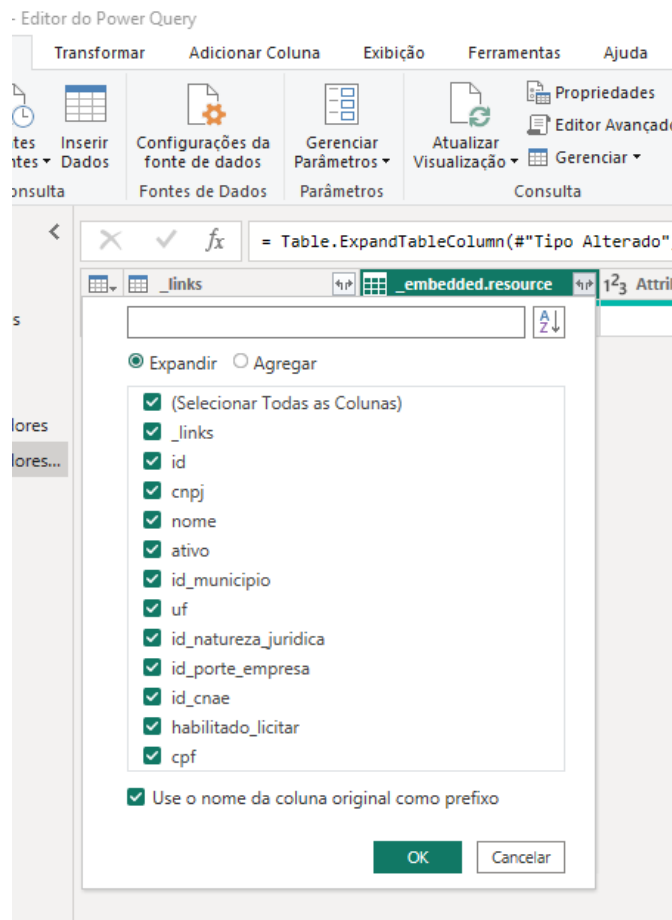


Figura 57

	ABC 123 _embedded.resource.id	ABC 123 _embedded.resource.cnpj	ABC 123 _embedded.resource.nome	ABC 123 _embedded.resource.ativo	ABC 123 _emb
1	658	00077905000160	M R PEREIRA CONSORTOS EM MOTOS	true	60011
2	1228	00122327000136	SANTINVEST S/A CREDITO FINANCIAMENTO E INVESTIMENTOS	true	81051
3	4842	00469296000194	SINDICATO NACIONAL DOS TRABALHADORES NA PROTECAO AO VOO	true	60011
4	5491	00523354000110	UNIAO DOS CONTRIBUINTES DA PREVIDENCIA SOCIAL DO BRASIL	true	58653
5	5962	00560102000161	ASSOCIACAO DOS SERV PUBLICOS DO GDF ASPGDF	true	97012
6	6817	00628107000189	FUNDACAO ASSISTENCIAL DOS SERVIDORES DO MINISTERIO DA FAZE...	true	97012
7	6985	00643742000135	FUNDACAO HABITACIONAL DO EXERCITO - FHE	true	97012
8	7093	00652663000190	GRAFIMA EDITORA GRAFICA E MATERIAL DE EXPEDIENTE LTDA	false	9210
9	7125	00655522000121	ASSOCIACAO DE POUPANCA E EMPRESTIMO POUPEX	true	97012
10	10442	00952415000165	COOPERATIVA DE CREDITO DE LIVRE ADMISSAO CREDFAZ LTDA	true	97012
11	13946	01242924000167	EMPRESA DE CONSERVACAO E ASSEIO LTDA	true	90514
12	25071	02172353000102	CAIXA ASSISTENCIAL UNIVERSITARIA DO RIO GRANDE DO NORTE	true	17612
13	25773	02231887000154	ORGANIZACAO SOCIAL INSTITUTO DO DEFICIENTE MILITAR E CIVIL DO ...	true	97012
14	26304	02275781000152	COOPERATIVA DE CREDITO MUTUO DOS SERVIDORES FEDERAIS NA PA...	true	20516
15	26478	02287929000179	BCARD ADMINISTRADORA DE CONVENIOS E BENEFICIOS LTDA	false	31054
16	32107	02674113000106	COOPERATIVA DE CREDITO DE SERVIDORES PUBLICOS COOPERPLAN L...	true	97012
17	32507	02703131000160	ROGERIO ARAJARIR TONETTO WINKLER EIRELI	false	88013
18	35640	02925815000107	MAGALHAES E MAGALHAES COMERCIO DE MOVEIS LTDA	true	97012
19	36281	02966385000171	CECM DOS SERV. DOS ORGAOS DO MIN. DA SAUDE NO EST. DO RJ	true	60011
20	36784	03000142000147	COOPERATIVA DE CREDITO, POUPANCA E INVESTIMENTO DOS POLICI...	true	88013
21	37054	03017677000120	BANCO J. SAFRA S.A	true	71072
22	37862	03076083000190	CONSTRUTORA NHAMBIQUARAS LTDA	true	91677
23	39115	03162368000144	ASSOCIACAO NACIONAL DOS ENGENHEIROS AGRONOMOS DO INCRA...	true	97012

Figura 58

Por fim, podemos ver na figura Figura 58 a tabela criada, que pode ser carregada no Power BI. Sendo assim, para importar os dados de um XML é necessário uma série de análises dos dados, expansões e conversões utilizando o Power Query.

Outra forma interessante de se obter dados é a partir da Web. O chamado Web Scrating, que vasculha os códigos HTML (*HyperText Markup Language*) ou Linguagem de Marcação de Hipertexto, para encontrar tabelas de dados e as importá-las para o Power BI.

O processo de Web Scrating se inicia quando se seleciona a obtenção de dados da Web, como demonstra a Figura 59.

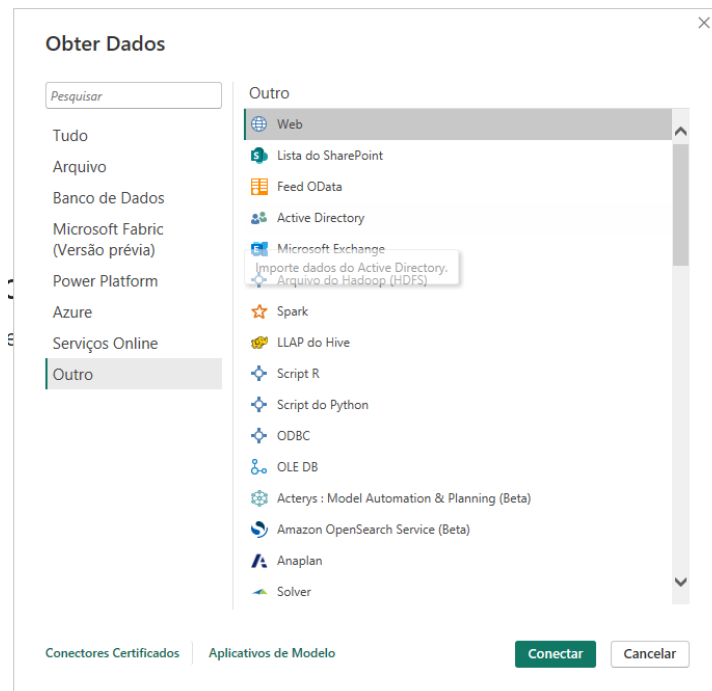


Figura 59

Certamente, é necessário ter um endereço da internet com os dados que se deseja importar para o Power BI. Em nosso exemplo vamos utilizar alguns dados do Google Shopping.

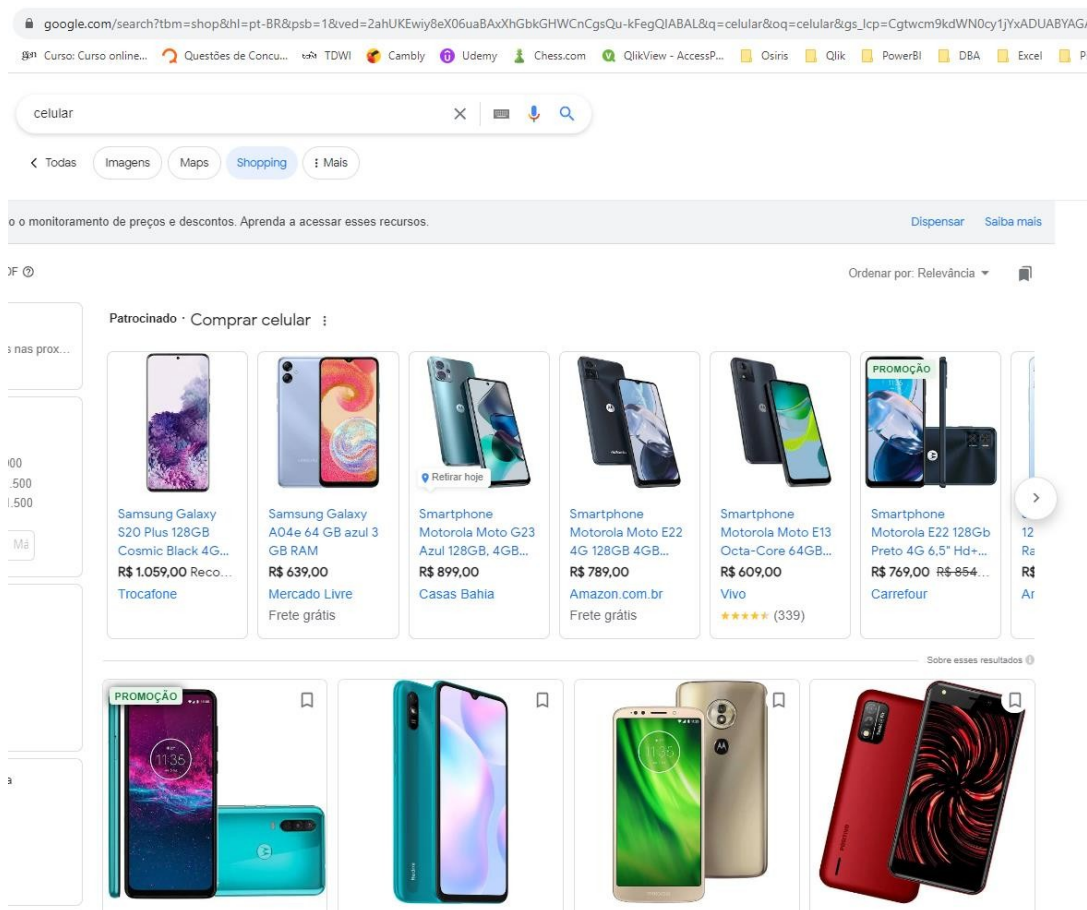


Figura 60

De posse do endereço de internet desejado, temos que copiá-lo para o Power BI. Como podemos ver a Figura 61.

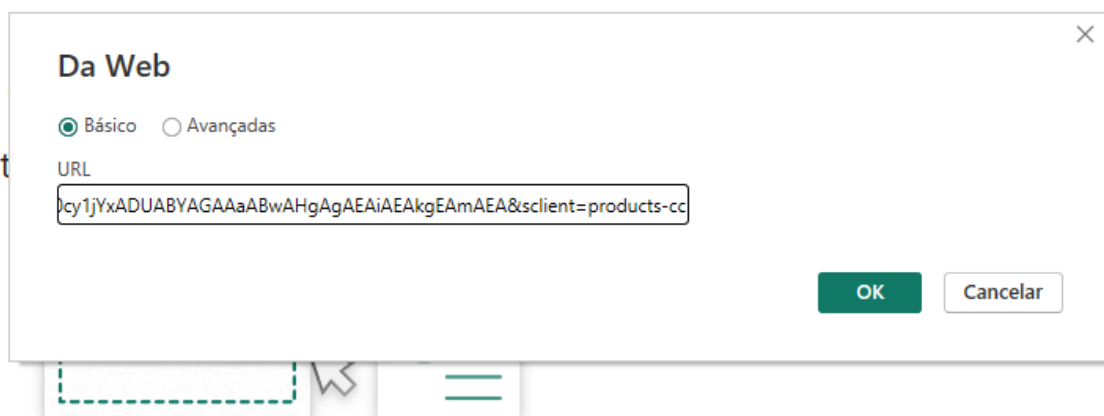


Figura 61

O Power BI vai vasculhar automaticamente toda a página procurando dados tabulares para serem importados. Como pode ser visto na Figura 62 o Power BI encontrou várias tabelas. O trabalho agora é ir procurando os dados desejados e possivelmente trabalhá-los no Power Query.

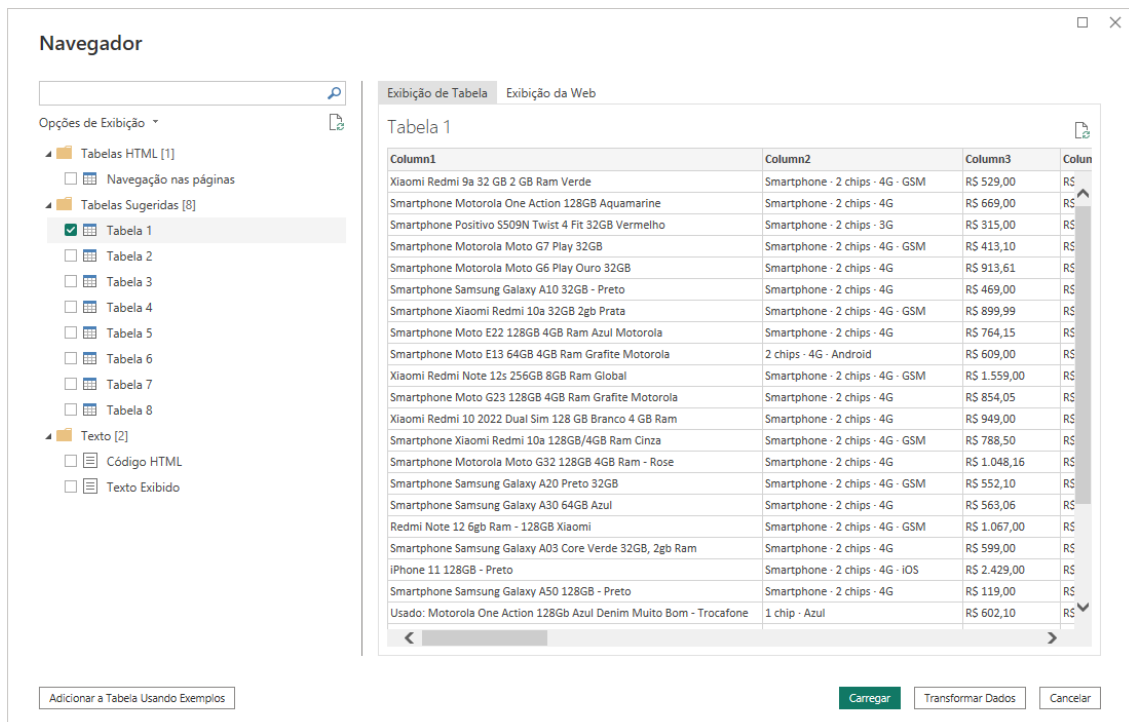


Figura 62

Power Query (Consultas) e a linguagem M

Como visto, o Power BI possibilita a obtenção de dados de diversas fontes. Para se trabalhar os dados a serem carregados no Power BI pode-se utilizar a ferramenta Power Query, selecionando a opção Transformar Dados, na Figura 62, chegando na tela da Figura 63. Onde, vários recursos podem ser usados para transformar os dados antes de carregá-los no Power BI. Por exemplo, retirar linhas duplicadas ou dividir-se colunas.

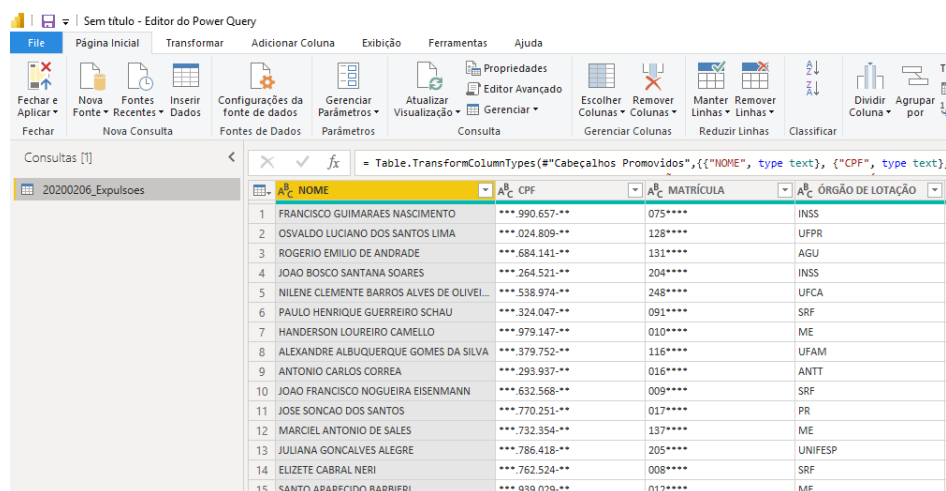


Figura 63

O Microsoft Power Query fornece uma experiência de obtenção de dados que abrange vários recursos. Uma das principais funcionalidades do Power Query é filtrar e combinar dados, ou seja, realizar o mashup ou mesclagem de dados de uma ou mais coleções e fontes de dados.

Além disso, cada etapa da transformação é armazenada e pode ser aproveitada com a linguagem M. A linguagem M é usada para o ETL – extrair os dados, transformar para poder carregá-los no Power BI.

A facilidade está no fato de se trabalhar em linguagem textual e, portanto, pode-se modificar a origem e fórmulas aplicadas apenas se modificando o texto. Além disso, pode-se facilmente voltar etapas.

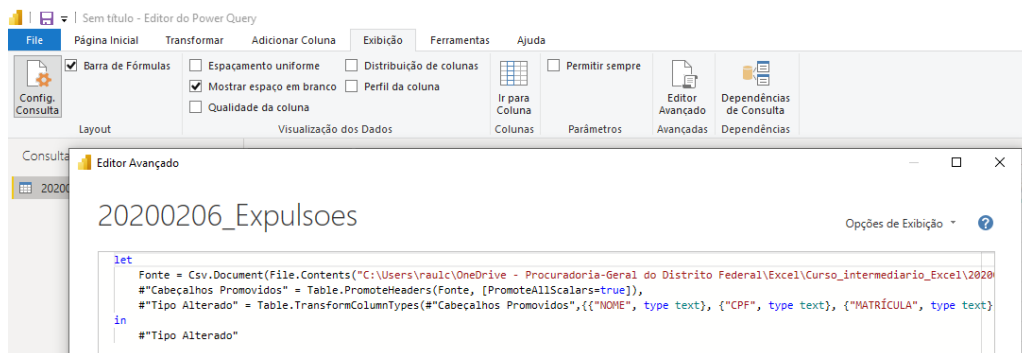


Figura 64

Transformando dados.

Trabalhar diretamente com a linguagem M requer um conhecimento mais avançado. Nós vamos ter uma noção nas seções seguintes. Estudaremos essa linguagem, principalmente, para entender como o Power Query registra os passos executados nas transformações e como conseguimos voltar nesses passos e reproduzi-los em outros dados.

Para facilitar o trabalho de transformação dos dados no processo ETL para o Power BI, o Power Query traz um menu para a transformação desses dados. Com opções facilitadas no estilo clicar e executar. O menu ou guia Transformar pode ser dividido em cinco grupos básicos: Tabelas, Colunas, Texto, Número e Data.

O grupo tabela serve para se realizar operações de transformação em toda a tabela. Por exemplo, a contagem de linhas será feita com um clique.



Figura 65

O grupo para trabalhar com colunas realiza operações em uma coluna selecionada. Por exemplo, substituir valores em cada célula da coluna.

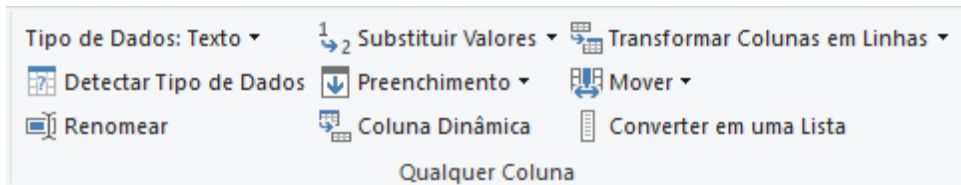


Figura 66

Por sua vez. O grupo que trabalha com texto vai agregar funcionalidades que ajudam a trabalhar especificamente com colunas do tipo de dados texto. Por exemplo, dividir a coluna por um delimitador.

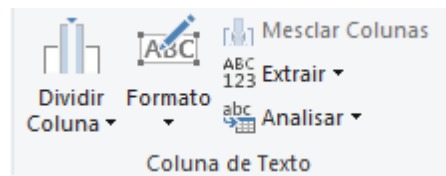


Figura 67

Já no caso das funcionalidades direcionadas aos números temos várias funções, inclusive os padrões como soma, subtração multiplicação e divisão.

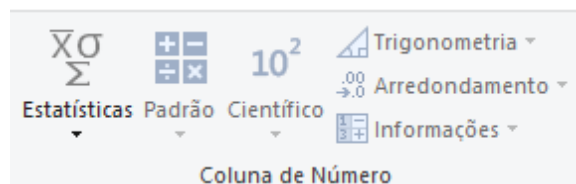


Figura 68

Por fim, mas não menos importante, temos as funcionalidades que trabalham com datas. Por exemplo, podemos retirar apenas o ano de uma data no formato dd/mm/aaaa – dia, mês e ano.

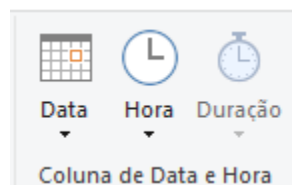


Figura 69

Cruzando dados

Uma função muito importante no Power Query é o cruzamento de dados, quando se realiza uma mesclagem de duas tabelas ou também conhecido como cruzamento de dados. Suponhamos que já se tenha obtido duas tabelas: Um com informações de compras públicas e dos servidores responsáveis por aquelas compras. A tabela compras possui um campo ou uma coluna onde tem a matrícula do responsável por aquela compra.

A Figura 70 apresenta o painel de dados do Power BI, o qual demonstra que foram carregadas duas tabelas: A tabela Compras e a tabela Servidores. Na tabela Compras podemos observar a coluna Responsável e na tabela Servidores podemos observar o campo MATRICULA.

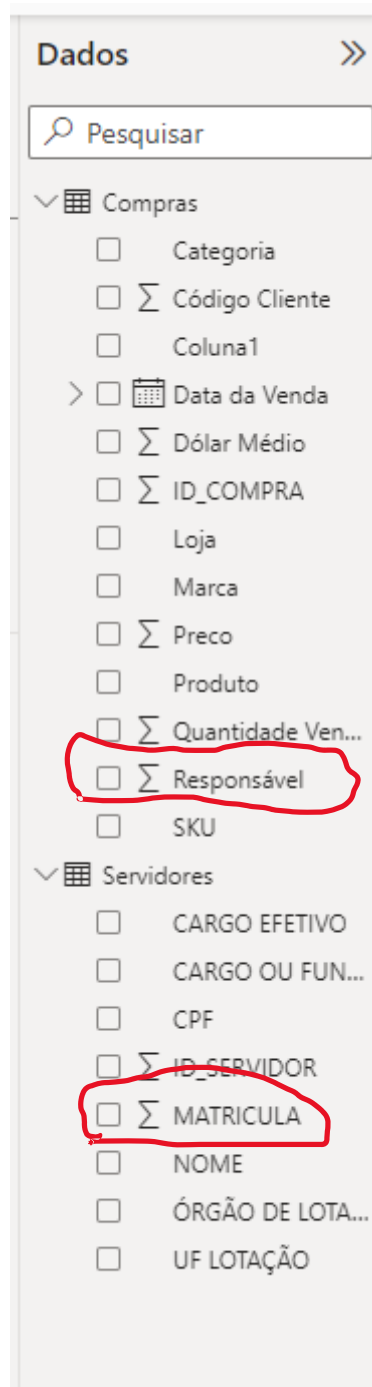


Figura 70

Caso sua análise de dados necessitasse cruzar os dados de compras realizadas por um servidor ou os servidores que fizeram mais compras em um determinado período, se faz necessário uma mesclagem das duas tabelas. A

Figura 71 e Figura 72 mostram como retornamos ao Power Query para realizar uma consulta que vai mesclar duas tabelas e criar uma nova.

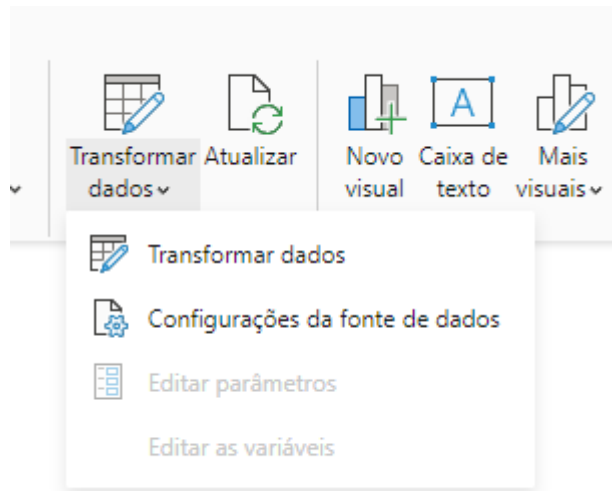


Figura 71

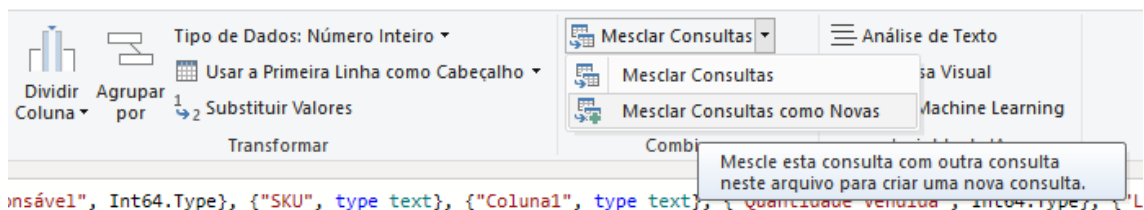


Figura 72

Porém, antes de fazermos as demais etapas do passo a passo de um cruzamento de dados, devemos entender o conceito de produto cartesiano. O produto cartesiano é um conceito fundamental na matemática que descreve uma operação que combina elementos de dois conjuntos para criar um, novo, conjunto de pares ordenados. Esses pares ordenados representam todas as possíveis combinações de elementos dos dois conjuntos originais. O produto cartesiano é frequentemente denotado como $A \times B$, onde A e B são os conjuntos de origem.

Vamos considerar dois conjuntos como exemplo:

$$A = \{1, 2\}$$

$$B = \{a, b, c\}$$

O produto cartesiano de A e B , denotado como $A \times B$, seria:

$$A \times B = \{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c)\}$$

Neste exemplo, cada par ordenado representa uma combinação de um elemento de A com um elemento de B. Existem, portanto, 2 elementos em A e 3 elementos em B, resultando em $2 * 3 = 6$ pares ordenados no produto cartesiano.

O produto cartesiano pode ser estendido para mais de dois conjuntos. Se você tiver três conjuntos, A, B e C, o produto cartesiano $A \times B \times C$ consistirá em triplas ordenadas que combinam elementos de A, B e C. O mesmo princípio pode ser aplicado para mais conjuntos.

Desse modo quando mesclamos duas tabelas realizamos o produto cartesiano entre o consulto de todas as tuplas/linhas de cada tabela. A Figura 73 demonstra como fazemos um produto cartesiano onde apenas as linhas que os campos matrícula e responsável irão coincidir serão retornadas.

Mesclar

Selecione as tabelas e as colunas correspondentes para criar uma tabela mesclada.

Servidores

ID_SERVIDOR	MATRICULA	CPF	NOME	ÓRGÃO DE LOTAÇÃO
1	5382670750	***.990.657-**	FRANCISCO GUIMARAES NASCIMENTO	INSS
2	5340611280	***.024.809-**	OSVALDO LUCIANO DOS SANTOS LIMA	UFPR
3	6286021310	***.684.141-**	ROGERIO EMILIO DE ANDRADE	AGU
4	6491262040	***.264.521-**	JOAO BOSCO SANTANA SOARES	INSS

Compras

ID_COMPRA	Responsável	SKU	Coluna1	Quantidade Vendida	Loja	Data da Venda	Código C
1	4142750700	HL1013		13	Rio de Janeiro	01/01/2018	
2	8043020170	HL1019		9	Guarulhos	01/01/2018	
3	9438551500	HL1007		14	Salvador	01/01/2018	
4	7729990220	HL1006		7	Recife	01/01/2018	

Tipo de Junção

Externa esquerda (todas a partir da primeira, corresp...

Externa esquerda (todas a partir da primeira, correspondência a partir da segund...

Externa direita (todas a partir da segunda, correspondência a partir da primeira)

Externa Completa (todas as linhas de ambos)

Interna (apenas linhas correspondentes)

Anti esquerda (linhas apenas na primeira)

Anti direita (linhas apenas na segunda)

OK

Cancelar

Figura 73

Quando se clica Ok na tela anterior a situação da Figura 74 aparecerá. Neste caso cada linha da tabela servidor tem um conjunto de linhas da tabela consulta que o campo responsável corresponde ao campo matrícula.

The screenshot shows the 'Config. Consulta' window. The 'CARGO OU FUNÇÃO DE CONFIANÇA' dropdown is set to 'Compras'. The 'Propriedades' section shows the 'Nome' field set to 'Servidores_vs_Compras'. The 'Etapas Aplicadas' section shows the 'Fonte' field set to 'Fonte'.

Figura 74

Uma operação de expansão, como demonstra a Figura 75 resulta na tela da Figura 76, que tem cada uma das linhas de servidor e sua compra correspondente.

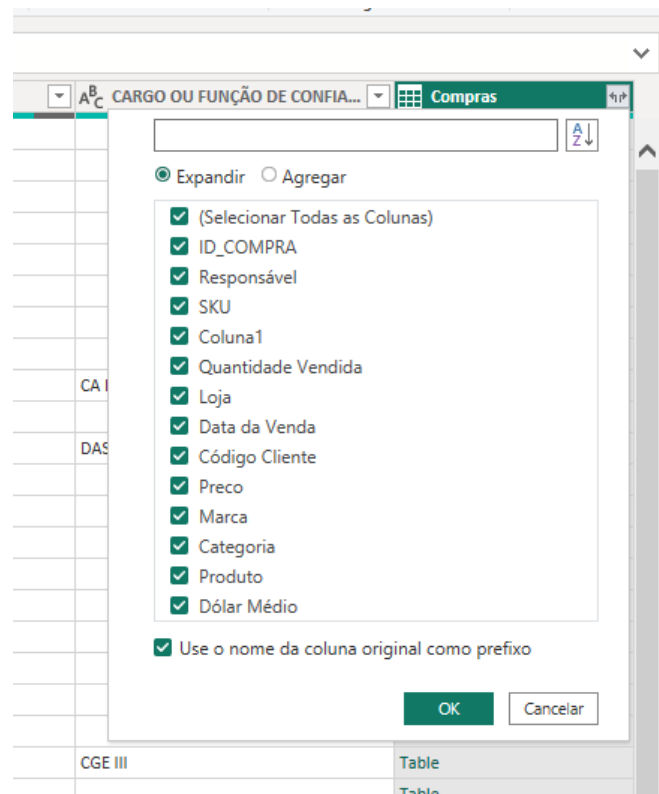


Figura 75


```

/JX = table.expandtablecolumn(fonte, "Compras", {"ID_COMPRA", "Responsavel", "SKU", "Coluna1", "Quantidade Vendida", "Loja",

```

A ^B C CARGO EFETIVO	A ^B C CARGO OU FUNÇÃO DE CONFIA...	1 ² 3 Compras.ID_COMPRA	A ^B C Compras.Responsável	A ^B C Com
Auxiliar de Serviços Gerais		498	5382670750	HL1003
Agente de Atividade Agropecuária		1	4142750700	HL1013
Agente de Atividade Agropecuária		884	4142750700	HL1019
Delegado de Polícia Federal		2	8043020170	HL1019
PROFESSOR DO MAGISTÉRIO SUPERIOR		137	5447052480	HL1009
PROFESSOR DO MAGISTÉRIO SUPERIOR		1262	3488521160	HL1018
PROFESSOR DO MAGISTÉRIO SUPERIOR		2040	3488521160	HL1003
POLICIAL RODOVIÁRIO FEDERAL		11	2153361060	HL1002
TÉCNICO DO SEGURO SOCIAL		323	1288311370	HL1018
AUXILIAR DE ENFERMAGEM		645	3247542050	HL1016
AGENTE ANALISTA GRAF FIS SUP		13	2195290440	HL1021

Figura 76

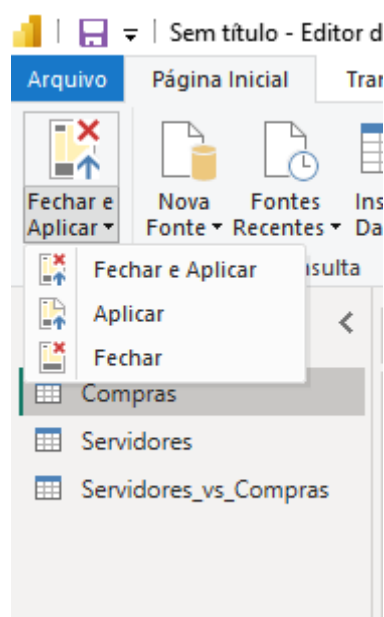


Figura 77

Quando se fecha e aplica as alterações de mesclagem, como pode ser visto na Figura 77, o Power BI carrega a nova tabela no modelo de dados do relatório e os dados poderão ser trabalhados nas visualizações.

Introdução a linguagem M

Para dominar realmente qualquer consulta de dados é importante entender as expressões usadas na Linguagem de fórmula do Power Query (informalmente conhecida como "M"). O Power Query insere scripts M em uma ampla variedade de produtos da Microsoft, incluindo Excel, Power BI e Analysis Services.

A linguagem M é uma linguagem de expressões para limpeza, transformações e combinação de dados, especificada como uma linguagem funcional essencialmente pura, de ordem mais alta, dinamicamente tipada e parcialmente lenta. Nesta seção vamos ter uma noção do que é a linguagem M e o que ela pode fazer, basicamente.

Expressões e valores

O constructo central no M é a *expressão*. Uma expressão pode ser avaliada (computada) e produzindo um *return* – resultado, que pode ser um valor ou tabela/consulta.

Embora muitos valores possam ser escritos literalmente como uma expressão, um valor não é uma expressão. Por exemplo, a expressão 1 é avaliada como o valor 1, por sua vez a expressão 1+1 é avaliada como o valor 2. Essa distinção é sutil, mas importante. As expressões são receitas para avaliação e os valores são os resultados da avaliação.

Os exemplos a seguir ilustram os diferentes tipos de valores disponíveis em M. Por convenção, um valor é escrito usando o formato literal em que eles apareceriam em uma expressão que é avaliada apenas para esse valor.

Um valor *primitivo* é um valor de parte única, como um número, um valor lógico, um texto ou valor nulo. Um valor nulo pode ser usado para indicar a ausência de dados.

```
123                // A number
true              // A logical
"abc"            // A text
null             // null value
```

Um valor de *lista* é uma sequência ordenada de valores. O M dá suporte a listas infinitas, mas, se escritas como um literal, as listas têm um comprimento fixo. Os caracteres de chave { e } denotam o início e o fim de uma lista.

```
{123, true, "A"} // list containing a number, a logical,
and a text
{1, 2, 3}        // list of three numbers
```

Um *registro* é um conjunto de *campos*. Um campo é um par de nome/valor em que o nome é um valor de texto exclusivo dentro do registro do campo. A sintaxe literal para valores de registro permite que os nomes sejam gravados sem aspas, um formato também conhecido como *identificadores*. O exemplo a seguir mostra um registro que contém três campos denominados "A", "B" e "C", que têm os valores 1, 2 e 3.

```
[
    A = 1,
    B = 2,
    C = 3
]
```

Nome	Idade
Raul	43
Issac	4

Figura 78

Uma *tabela* é um conjunto de registros organizados em colunas (que são identificadas por nome) e linhas. Não há nenhuma sintaxe literal para criação de uma tabela, mas há várias funções padrão que podem ser usadas para criar tabelas com base em listas ou registros. A expressão a seguir cria uma tabela com o presente na Figura 78.

```
Pessoa = #table( {"Nome", "Idade"}, { {"Raul", 43}, {"Issac", 4} } )
```

Uma *função* é um valor que, quando invocado com argumentos, produz um novo valor. As funções são identificadas por sua assinatura – por seu nome e pela listagem dos *parâmetros* da função entre parênteses – seguidos pelo símbolo de ir para =>, seguido pela expressão que define a função. Essa expressão normalmente se refere aos parâmetros (por nome).

```
Nome = (x, y) => (x + y) / 2
```

Avaliação

O modelo de avaliação da linguagem M é um modelo de avaliação normalmente encontrado em planilhas, em que a ordem dos cálculos pode ser determinada com base nas dependências entre as fórmulas nas células.

Se você tiver escrito fórmulas em uma planilha como o Excel, poderá reconhecer que as fórmulas à esquerda na Figura 79 resultarão nos valores à direita quando calculadas.

	A		A
1	=A2 * 2	1	4
2	=A3 + 1	2	2
3	1	3	1

Figura 79

Na linguagem M, uma expressão pode referenciar outras partes da expressão por nome, e o processo de avaliação determinará automaticamente a ordem na qual as expressões referenciadas serão calculadas.

Podemos usar um registro para produzir uma expressão equivalente ao exemplo da planilha acima. Ao inicializar o valor de um campo, podemos fazer referência a outros campos dentro do registro pelo uso do nome do campo, da seguinte maneira:

```
[
    A1 = A2 * 2,
    A2 = A3 + 1,
    A3 = 1
]
```

A expressão acima é equivalente à seguinte (no sentido de que ambas são avaliadas para valores iguais):

```
[
    A1 = 4,
    A2 = 2,
    A3 = 1
]
```

Funções

No M, uma *função* é um mapeamento de um conjunto de valores de entrada para um valor de saída. Para escrever uma função, primeiro é preciso nomear o conjunto necessário de valores de entrada (os parâmetros para a função) e, em seguida, fornecer uma expressão que calcule o resultado da função usando esses valores de entrada (o corpo da função) após o símbolo de ir para (\Rightarrow). Por exemplo:

```
(x) => x + 1                // função que adiciona um
(x, y) => x + y             // função soma dois valores
```

Uma função é um valor, assim como um valor de número ou de texto. O exemplo a seguir mostra uma função que é o valor de um campo *Add*, que é *invocada* ou executada de vários outros campos. Quando uma função é chamada, um conjunto de valores é especificado, que são substituídos logicamente pelo conjunto necessário de valores de entrada dentro da expressão do corpo da função.

```
[
    Soma = (x, y) => x + y,
    UmMaisUm = Soma(1, 1),    // 2
    DoisMaisDois = Soma(1, 2) // 3]
```

Biblioteca, pacote ou conjunto de funções e constantes prontas

O M inclui um conjunto comum de definições disponíveis para uso por meio de uma expressão chamada de *biblioteca padrão* ou simplesmente biblioteca. Essas definições consistem em um conjunto de valores nomeados. Os nomes de valores

fornecidos por uma biblioteca estão disponíveis para uso em uma expressão sem terem sido definidos explicitamente pela expressão. Por exemplo:

```
Number.E // Euler's number e (2.7182...)
Text.PositionOf("Hello", "ll") // 2
```

Operadores

O M inclui um conjunto de operadores que podem ser usados em expressões. Os *operadores* são aplicados a *operandos* para formar expressões simbólicas. Por exemplo, na expressão `1 + 2` os números `1` e `2` são operandos e o operador é o operador de adição (+).

O significado de um operador pode variar dependendo do tipo de valores dos seus operandos. Por exemplo, o operador de adição pode ser usado com outros tipos de valores que não sejam números:

```
1 + 2 // adição que resulta em 3
#time(12,23,0) + #duration(0,0,2,0) // time arithmetic: #time(12,25,0)
```

Outro exemplo de um operador com significado dependente do operando é o operador de combinação (&):

```
"A" & "BC" // Concatenação de texto: "ABC"
{1} & {2, 3} // Concatenação de lista: {1, 2, 3}
[ a = 1 ] & [ b = 2 ] // Merge em registro: [ a = 1, b = 2 ]
```

Observe que nem todas as combinações de valores são compatíveis com um operador. Por exemplo:

```
1 + "2" // error: adicionar um número a um texto não é suportado
```

As expressões que, quando avaliadas, encontram condições de operador indefinidas são avaliadas como erros.

Metadados

Os *metadados* são informações sobre um valor que está associado a esse valor. Os metadados são representados como um valor de registro, chamado de *registro de metadados*. Os campos de um registro de metadados podem ser usados para armazenar os metadados de um valor.

Cada valor tem um registro de metadados. Se o valor do registro de metadados não tiver sido especificado, o registro de metadados estará vazio (não terá nenhum campo).

Os registros de metadados fornecem uma forma de associar informações adicionais a qualquer tipo de valor de maneira discreta. A associação de um registro de metadados com um valor não altera o valor nem o comportamento dele.

Um valor de registro de metadados é associado a um valor existente usando a sintaxe `x meta y`. Por exemplo, este código associa um registro de metadados aos campos `Rating` e `Tags` com o valor de texto "Mozart":

```
"Mozart" meta [ Rating = 5, Tags = {"Classical"} ]
```

Para valores que já contêm um registro de metadados não vazio, o resultado de aplicar o operador meta é que a mesclagem do registro de metadados novo e do existente é computada. Por exemplo, as seguintes duas expressões são equivalentes uma à outra e à expressão anterior:

```
("Mozart" meta [ Rating = 5 ]) meta [ Tags = {"Classical"} ]
"Mozart" meta ([ Rating = 5 ] & [ Tags = {"Classical"} ])
```

Um registro de metadados pode ser acessado para um determinado valor usando a função *Value.Metadata*. No exemplo a seguir, a expressão no campo *ComposerRating* acessa o registro de metadados do valor no campo *Composer* e, em seguida, acessa o campo *Rating* do registro de metadados.

```
[
  Composer = "Mozart" meta [Rating = 5, Tags = {"Classical"} ],
  ComposerRating = Value.Metadata(Composer)[Rating] // 5
]
```

Expressão let

Muitos dos exemplos mostrados até agora incluíam todos os valores literais da expressão no resultado da expressão. A expressão *let* permite que um conjunto de valores sejam computados, que nomes sejam atribuídos a eles e, depois, que esses valores sejam usados em uma expressão subsequente que segue a instrução *in*. Por exemplo, em um exemplo de dados de vendas, poderíamos fazer:

```
let
  Vendas2007 =
    [
      Ano = 2007,
      Valor_Semestre_1 = 1000,
      Valor_Semestre_2 = 1100,
      Total = Valor_Semestre_1 + Valor_Semestre_2 // 2100
    ],
  Vendas2008 =
    [
      Ano = 2008,
      Valor_Semestre_1 = 1200,
      Valor_Semestre_2 = 1300,
      Total = Valor_Semestre_1 + Valor_Semestre_2 // 2500
    ]
in
  Vendas2007[Total] + Vendas2008[Total] // 4600
```

O resultado da expressão acima é um valor numérico (4600) que foi computado com base nos valores associados aos nomes *Vendas2007* e *Vendas2008*.

Expressão if

A expressão `if` seleciona entre duas expressões com base em uma condição lógica. Por exemplo:

```
if 2 > 1 then
    2 + 2
else
    1 + 1
```

A primeira expressão (`2 + 2`) será selecionada se a expressão lógica (`2 > 1`) for verdadeira e a segunda expressão (`1 + 1`) será selecionada se ela for falsa. A expressão selecionada (nesse caso, `2 + 2`) é avaliada e torna-se o resultado da expressão `if` (4).

Errors

Um *erro* é uma indicação de que o processo de avaliação de uma expressão não pôde produzir um valor.

Os erros são gerados por operadores e funções que encontraram condições de erros ou usaram a expressão de erro. Os erros são tratados usando a expressão `try`. Quando um erro é gerado, é especificado um valor que pode ser usado para indicar porque o erro ocorreu.

Uma expressão `try` converte valores e erros adequados em um valor de registro que indica se a expressão `try` tratou um erro ou não, bem como o valor apropriado ou o registro de erro que ela extraiu ao tratar o erro. Por exemplo, considere a seguinte expressão que gera um erro e o trata imediatamente:

```
try error "negative unit count"
```

Um caso comum é substituir erros por valores padrão. A expressão `try` também pode ser usada com uma cláusula `otherwise` opcional para obter apenas isso em um formato compacto:

```
try error "negative unit count" otherwise 42 // 42
```

Exemplos

Criar uma coluna randômica.

Criar uma mesclagem para mapa coroplético

Fazer uma condição de transformação com `if`

Acessando um Web Service

```
let
    Source1 =
        Json.Document(Web.Contents("https://receitaws.com.br/v1/cnpj/09346601000125")),
    #"Converted to Table" = Record.ToTable(Source1)
in
    #"Converted to Table"
```

Criando medidas, colunas e tabelas no Power BI.

Pode-se criar medidas, colunas e tabelas no menu Modelagem. Uma medida vai retornar um valor único e necessita de uma expressão DAX, que será estudada adiante. Por exemplo, clicando em Nova medida no menu Modelagem, como pode ser visto na Figura 80 deve-se incluir a expressão DAX presente na Figura 81 para se calcular o preço médio dos produtos.

Média de Preço =
`AVERAGE('Compras'[Preco])`



Figura 80

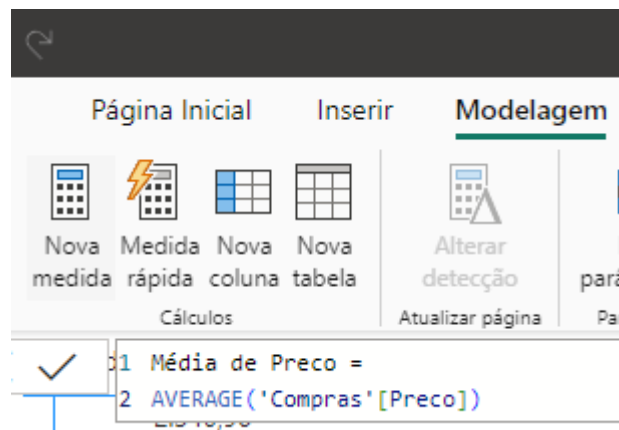


Figura 81

Medidas podem ser criadas com a opção de Medida rápida, como pode ser visto na Figura 82.

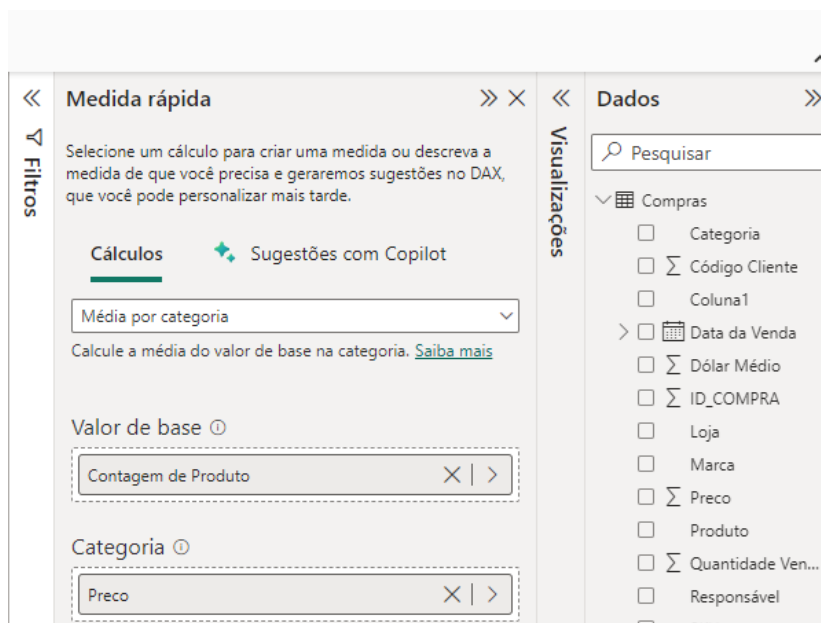


Figura 82

Para adicionar uma coluna a opção de Nova coluna deve ser selecionada, como pode ser visto na Figura 80, outra expressão DAX deve ser utilizada, como pode ser visto na Figura 83, mas agora o efeito é causado em cada linha da tabela, como pode ser visto na Figura 84.

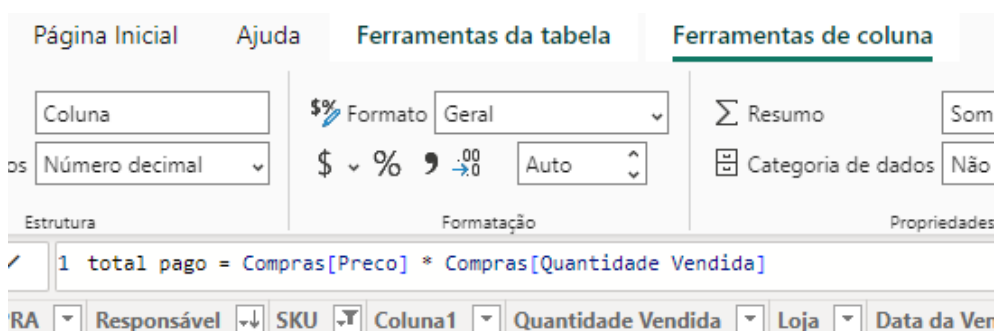


Figura 83

Produto	Dólar Médio	total pago
Galaxy J8	3,28	7700
Smart Watch Galaxy	4,11	2900
Galaxy S10	3,89	4800
Smart TV 50" 4K	3,75	18400
Smart TV 75" 4K	4,11	13500
Smart TV LED Full HD 55"	3,22	26000
Smart TV 75" 4K	3,53	5200
Smart TV LED Full HD 55"	3,9	8000
Smart TV LED 32"	3,77	5600
Smart TV 75" 4K	3,68	10400
Smart TV LED Full HD 55"	3,25	20000
Smart TV LED 32"	3,22	4200
Smart TV LED 32"	4,11	3000
Smart TV LED 32"	3,93	2400
Smart TV LED 32"	3,86	4500
Smart TV LED 32"	3,0	2800

Figura 84

Para se criar uma nova tabela a opção de Nova tabela da Figura 80 é selecionada e uma nova expressão DAX é necessária para, por exemplo, criar uma tabela com três colunas relacionadas: o preço por produto, a quantidade vendida e a multiplicação de cada uma das linhas resultando no valor pago por compra, como pode ser visto na Figura 85.

Arquivo Página Inicial Ajuda Ferramentas da tabela Ferramentas de coluna

Nome: Preço Formato: Texto Resumo: Não resumir Categoria de dados: Não categorizado

Estrutura Formatação Propriedades

1 Tabela = SELECTCOLUMNS(Compras,"Preço",[Preço],"Quantidade",[Quantidade Vendida],"Preço x Quantidade", [total pago])

Preço	Quantidade	Preço x Quantidade
2000	17	34000
1550	15	23250
1500	15	22500
1500	15	22500
700	15	10500
700	15	10500

Figura 85

Introdução a linguagem Dax

Como pode ser visto o DAX é muito importante no Power BI, A DAX (Data Analysis Expressions) é uma linguagem de expressão de fórmula usada nos Analysis Services, no Power BI, no SQL Server e no Power Pivot no Excel. As expressões DAX são usadas em **colunas calculadas**, **tabelas calculadas**, **medidas etc.**

Antes de avançar no estudo da linguagem DAX, vamos aprender sobre a ferramenta DaxStudio. O DaxStudio é uma ferramenta utilizada para analisar as expressões DAX. Ela possibilita a otimização e aumento do desempenho dos relatórios e painéis e Power BI. Uma vantagem interessante do DAX Studio é receber retornos visíveis tabulares, como se fosse uma consulta no SQL Server.

Por padrão, todas as expressões DAX retornam tabelas. Ou seja, há uma dificuldade natural de se imprimir medidas do Power BI. Para retornar medidas é necessário utilizar a função ROW, que retorna uma linha em DAX. Praticamente todas as Queries no DAX Studio iniciam com EVALUATE devido a sua funcionalidade de avaliar as funções permitindo melhoria de desempenho.



As consultas DAX podem ser criadas e executadas no SSMS (SQL Server Management Studio) e em ferramentas de software livre, como o DAX Studio (daxstudio.org). Ao contrário das fórmulas de cálculo DAX, que só podem ser criadas em modelos de dados tabulares, as consultas DAX também podem ser executadas em modelos multidimensionais do Analysis Services. Uma consulta DAX é uma instrução, semelhante a uma instrução SELECT no T-SQL. O tipo mais básico de consulta DAX é uma instrução EVALUATE.

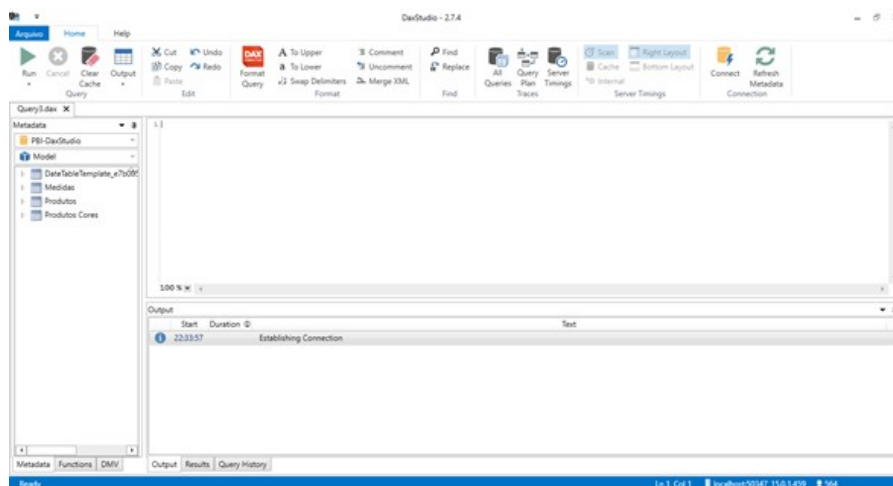


Figura 86

Por exemplo:

```
EVALUATE
( FILTER ( 'DimProduct', [SafetyStockLevel] < 200 ) )
ORDER BY [EnglishProductName] ASC
```

O exemplo anterior retorna nos resultados uma tabela que lista apenas esses produtos com um SafetyStockLevel menor do que 200, em ordem crescente por EnglishProductName.

EVALUATE

```
ROW("Subtração", sum('2020_OrcamentoDespesa zip'[ORÇAMENTO INICIAL (R$)]) - SUM('2020_OrcamentoDespesa zip'[ORÇAMENTO ATUALIZADO (R$)]))
```

O exemplo anterior retorna uma linha com a subtração do orçamento inicial menos o orçamento atualizado.

EVALUATE

```
ROW("Licitacao", COUNT('202001_Licitacao'[Número Licitação]), "Licitacao distinta", DISTINCTCOUNT('202001_Licitacao'[Número Licitação]))
```

O exemplo anterior retorna uma linha com a contagem das licitações e outra medida que apresenta a quantidade de licitações distintas, baseado no número da licitação.

EVALUATE

```
ROW("soma", CALCULATE(SUM('202001_Licitacao'[Valor Licitação]), FILTER(DISTINCT('202001_Licitacao'), [Município] = "RECIFE")))
```

O exemplo anterior retorna soma todos os valores das licitações de RECIFE.

Colunas calculadas

Voltando ao estudo da linguagem DAX. Vamos entender as colunas calculadas. Uma coluna calculada se trata de uma coluna que é adicionada a uma tabela existente por meio de uma fórmula DAX, similar as colunas calculadas do Excel. Quando uma coluna calculada contiver uma fórmula DAX válida, os valores serão calculados para cada linha da tabela. Os valores serão armazenados no modelo de dados na memória. Por exemplo, um valor para cada linha na tabela é calculado inserindo valores da coluna Concatenação:

Concatenação = 'Tabela 1'[Nome] & " " & 'Tabela 1'[Sobrenome]

O resultado de cada linha na coluna calculada é calculado imediatamente e é exibido na área de dados do Power BI. Os valores de coluna apenas serão recalculados se a tabela ou qualquer tabela relacionada for processada ou o modelo for descarregado da memória e recarregado, como ocorre ao fechar e reabrir um arquivo do Power BI Desktop.

Tabelas calculadas

Uma tabela calculada é um objeto computado com base em uma expressão DAX (fórmulas). Em vez de consultar e carregar valores em uma coluna calculada ele cria uma nova tabela de uma fonte de dados.

Tabelas calculadas são úteis ao configurar um conjunto de linhas filtradas ou um subconjunto ou superconjunto de colunas de outras tabelas existentes. Com isso, você pode manter a tabela original intacta ao criar variações dessa tabela para dar

suporte a cenários específicos. Ou seja, tabelas calculadas podem ser consideradas um estágio a mais na análise de dados.

Tabelas calculadas dão suporte a relações com outras tabelas. As colunas na tabela calculada têm tipos de dados, formatação e podem pertencer a uma categoria de dados. Tabelas calculadas podem ser nomeadas e exibidas ou ocultas assim como qualquer outra tabela. Tabelas calculadas serão recalculadas se qualquer uma das tabelas das quais elas recebem dados forem renovadas ou atualizadas.

Exemplo:

Tabela =

```
(CROSSJOIN(SELECTCOLUMNS('202001_Licitação','numero lic',[Número
Licitação]),SELECTCOLUMNS('20200206_Expulsoes','sigla org',
[ÓRGÃO DE LOTAÇÃO])))
```

Trabalhando com tabelas e colunas

Tabelas em modelos de dados tabulares se parecem com tabelas do Excel, mas são diferentes na maneira como funcionam com os dados e com fórmulas:

- As fórmulas só funcionam com tabelas e colunas, e não com células individuais, referências de intervalos ou matrizes.
- As fórmulas podem usar relações para obter valores de tabelas relacionadas. Os valores recuperados sempre são relacionados ao valor da linha atual.
- Não pode haver dados irregulares, como ocorre em uma planilha do Excel. Cada linha de uma tabela deve conter o mesmo número de colunas. No entanto, pode haver valores vazios em algumas colunas. As tabelas de dados do Excel e as tabelas de dados do modelo tabular não são intercambiáveis.
- Como um tipo de dados é definido para cada coluna, cada valor nessa coluna deve ser do mesmo tipo.

Referenciando tabelas e colunas em fórmulas

É possível referenciar qualquer tabela e coluna usando seu nome. Por exemplo, a seguinte fórmula ilustra como referenciar colunas de duas tabelas usando o nome *totalmente qualificado*:

= SUM('2020_OrcamentoDespesa zip'[ORÇAMENTO INICIAL (R\$)]) –

SUM('2020_OrcamentoDespesa zip'[ORÇAMENTO ATUALIZADO (R\$)])

Quando uma fórmula é avaliada, o designer do modelo primeiro verifica a sintaxe geral e, em seguida, os nomes das colunas e tabelas que você fornece em relação a possíveis colunas e tabelas no contexto atual. Se o nome for ambíguo ou se a coluna ou tabela não puder ser encontrada, você obterá um erro na fórmula.

Relações de tabela

Ao criar relações entre tabelas, você obtém a habilidade de valores relacionados em outras tabelas a serem usados em cálculos. Em muitos casos, porém,

uma relação talvez não seja necessária. Muitas funções DAX exigem uma relação existente entre as tabelas, ou entre várias tabelas, para localizar as colunas que você referenciou e retornar resultados que tenham sentido.

Os modelos de dados tabulares dão suporte a várias relações entre tabelas. Para evitar confusão ou resultados incorretos, apenas uma relação por vez é designada como a ativa, mas você pode alterar a relação ativa, conforme necessário, para atravessar conexões diferentes nos dados em cálculos. A função **USERELATIONSHIP** pode ser usada para especificar uma ou mais relações a serem usadas em um cálculo específico.

É importante observar essas regras de design de fórmula ao usar relações:

- Quando tabelas forem conectadas por uma relação, você deve verificar se as duas colunas usadas como chaves têm valores correspondentes. A integridade referencial não é imposta, portanto, é possível ter valores não correspondentes em uma coluna de chave e ainda criar uma relação. Se isso ocorrer, você deve estar ciente de que valores em branco ou não correspondentes pode afetar os resultados das fórmulas.
- Ao vincular tabelas em seu modelo usando relações, você amplia o escopo, ou *contexto*, no qual as fórmulas são avaliadas. As alterações no contexto resultantes da adição de novas tabelas, novas relações, ou de alterações na relação ativa podem fazer seus resultados serem alterados de maneira imprevista.

Exemplo:

Tabela =

```
VAR dados = DISTINCT(CROSSJOIN(SELECTCOLUMNS('202001_Licitação','numero
lic',[Número Licitação]),SELECTCOLUMNS('202001_Licitação','nome org',[Nome
Órgão]))) // cria uma variável no DAX.

RETURN // finaliza declaração das variáveis.

FILTER(ADDCOLUMNS(dados,
"sigla",LOOKUPVALUE('completa'[unidades.sigla],'completa'[unidades.nome],[nome
org])),[sigla]<>"") \\ faz o filtro diante do cruzamento das tabelas.
```

Medidas

As medidas são fórmulas de cálculo dinâmico em que os resultados mudam dependendo do contexto. As medidas são usadas em relatórios que dão suporte à combinação e à filtragem dos dados de modelo usando vários atributos, como um relatório do Power BI ou a Tabela Dinâmica ou o Gráfico Dinâmico do Excel. As medidas são criadas usando a barra de fórmulas DAX no designer do modelo.

Uma fórmula em uma medida pode usar as funções de agregação padrão criadas automaticamente usando o recurso Autossoma, como **COUNT** ou **SUM**, ou você pode definir sua própria fórmula usando a barra de fórmulas DAX. Medidas nomeadas podem ser passadas como um argumento para outras medidas.

Quando você define uma fórmula para uma medida na barra de fórmulas, um recurso de Dica de ferramenta mostra uma visualização rápida do que os resultados seriam para o total no contexto atual, mas, do contrário, os resultados não são produzidos imediatamente em qualquer lugar.

A razão pela qual você não pode consultar os resultados filtrados do cálculo imediatamente é que o resultado de uma medida não pode ser determinado sem contexto. Para avaliar uma medida, é necessário um aplicativo cliente de relatório que pode fornecer o contexto necessário para recuperar os dados relevantes para cada célula e, em seguida, avaliar a expressão para cada célula.

Independentemente do cliente, uma consulta separada é executada para cada célula nos resultados. Isso significa que cada combinação de cabeçalhos de linha e de coluna em uma Tabela Dinâmica ou cada seleção de segmentações e filtros em um relatório do Power BI gera um subconjunto diferente de dados sobre o qual a medida é calculada. Por exemplo, usando esta fórmula de medida muito simples:

```
n_licitacoes = COUNT('202001_Licitação'[Número Licitação])
```

Diferentemente das colunas calculadas, a sintaxe de uma medida inclui o nome da medida que antecede a fórmula. No exemplo que acabou de ser fornecido, o nome **Total de Vendas** é exibido antes da fórmula. Após criar uma medida, o nome e sua definição serão exibidos na lista Campos do aplicativo cliente de relatório e dependendo se as perspectivas e funções estarão disponíveis a todos os usuários do modelo.

Outros exemplos :

```
AVERAGE('202001_Licitação'[Valor Licitação])
```

```
MAX('202001_Licitação'[Valor Licitação])
```

```
MIN('202001_Licitação'[Valor Licitação])
```

```
DIVIDE(SUM('202001_Licitação'[Valor Licitação]),1000000))
```

```
DISTINCTCOUNT('202001_Licitação'[Número Licitação])
```

Valor booleano

Uma fórmula DAX pode ser avaliada como uma condição booleana TRUE/FALSE.

```
= DIVIDE(SUM('202001_Licitação'[Valor Licitação]),1000000) > 500
```

Fórmulas

As fórmulas DAX são essenciais para criar cálculos em colunas calculadas e medidas.

Conceitos básicos de fórmula

As fórmulas DAX podem ser muito simples ou muito complexas. A tabela a seguir mostra alguns exemplos de fórmulas simples que poderiam ser usadas em uma coluna calculada.

Fórmula	Definição
= TODAY()	Insere a data de hoje em cada linha de uma coluna calculada.
= 3	Insere o valor 3 em cada linha de uma coluna calculada.
= [Column1] + [Column2]	Adiciona os valores na mesma linha de [Coluna1] e [Coluna2] e coloca os resultados na coluna calculada da mesma linha.

Se a fórmula criada for simples ou complexa, você poderá usar as seguintes etapas ao criar uma fórmula:

1. Cada fórmula deve começar com um sinal de igual (=).
2. Você pode digitar ou selecionar um nome de função ou digitar uma expressão.
3. Comece digitando as primeiras letras da função ou dando o nome desejado. AutoComplete exibe uma lista de funções, tabelas e colunas disponíveis. Pressione TAB para adicionar um item da lista AutoComplete à fórmula.
4. Forneça os argumentos para a função selecionando-os em uma lista suspensa de possíveis tabelas e colunas ou digitando valores.
5. Veja se há erros de sintaxe: verifique se todos os parênteses estão fechados e se as colunas, as tabelas e os valores estão referenciados corretamente.
6. Pressione ENTER para aceitar a fórmula.

Variáveis

Você pode criar variáveis dentro de uma expressão usando [VAR](#). Tecnicamente, o VAR não é uma função; é uma palavra-chave para armazenar o resultado de uma expressão como uma variável nomeada. Essa variável pode ser passada como um argumento para outras expressões de medida. Por exemplo:

VAR

TotalQty = SUM (Sales[Quantity])

Return

IF (
 TotalQty > 1000,
 TotalQty * 0.95,


```
TotalQty * 1.25
)
```

Neste exemplo, TotalQty pode ser passado como uma variável nomeada para outras expressões. As variáveis podem ser de qualquer tipo de dados escalares, incluindo tabelas. O uso de variáveis em suas fórmulas DAX pode ser incrivelmente eficiente.

Tipos de dados

É possível importar para um modelo de diversas fontes de dados diferentes que podem dar suporte a diferentes tipos de dados. Quando você importa os dados em um modelo, os dados são convertidos em um dos tipos de dados de modelo de tabela. Quando os dados de modelo são usados em um cálculo, os dados são convertidos em um tipo de dados DAX para a duração e a saída do cálculo. Quando você cria uma fórmula DAX, os termos usados na fórmula determinarão o tipo de dados de valor retornado automaticamente.

O DAX dá suporte aos seguintes tipos de dados:

Tipo de dados em modelo	Tipos de dados em DAX	Descrição
Número Inteiro	Um valor inteiro de 64 bits (oito bytes) ^{1, 2}	Números sem casas decimais.
Número Decimal	Um número real de 64 bits (oito bytes) ^{1, 2}	Números reais são números que podem ter casas decimais
Booleano	Booleano	Um valor True ou False.
Texto	Cadeia de caracteres	Uma cadeia de caracteres de dados de caractere.
Data	Data/hora	Datas e horas em uma representação de data-hora aceita.
Moeda	Moeda	O tipo de dados de moeda permite valores com dígitos decimais de precisão fixa.
N/D	Em branco	Um espaço em branco é um tipo de dados no DAX que representa e substitui nulos SQL. É possível criar um espaço em branco usando a função BLANK e testar se há espaços em branco usando a função lógica, ISBLANK.

Modelos de dados tabulares também incluem o tipo de dados de *tabela* como a entrada ou a saída para muitas funções DAX. Por exemplo, a função [FILTER](#) usa uma

tabela como entrada e gera outra tabela que contém apenas as linhas que atendam às condições do filtro. Ao combinar funções de tabela com funções de agregação, você pode executar cálculos complexos em conjuntos de dados definidos de forma dinâmica.

Operadores

A linguagem DAX usa quatro tipos diferentes de operadores de cálculo em fórmulas:

- Operadores de comparação para comparar valores e retornar um valor TRUE\FALSE lógico.
- Operadores aritméticos para executar cálculos aritméticos que retornam valores numéricos.
- Operadores de concatenação de texto para unir duas ou mais cadeias de caracteres de texto.
- Operadores lógicos que combinam duas ou mais expressões para retornar um único resultado.

Trabalhando com o DAX.

Outros exemplos interessantes são o COUNTROWS, DISTINCT, FILTER e CALCULATE.

A função COUNTROWS conta o número de linhas na tabela especificada ou em uma tabela definida por uma expressão.

= COUNTROWS('Orders')

DISTINCT retorna uma tabela de coluna única que contém os valores distintos da coluna especificada. Em outras palavras, valores duplicados são removidos e apenas valores exclusivos são retornados.

= DISTINCT('202001_Licitação')

FILTER retorna uma tabela que representa um subconjunto de outra tabela ou expressão.

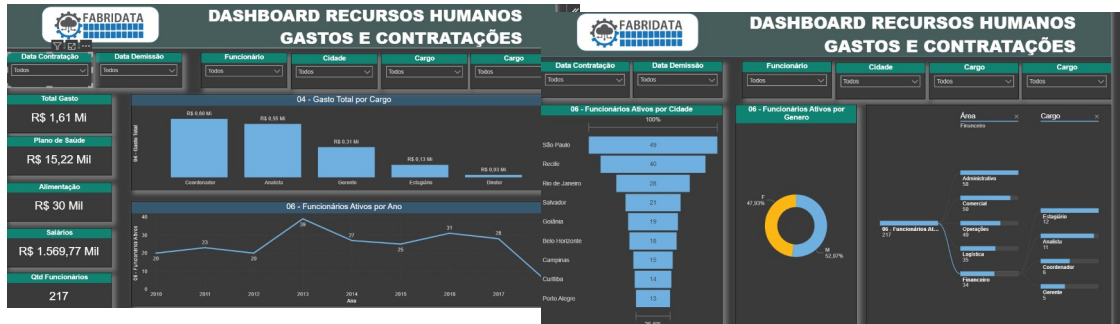
= FILTER(DISTINCT('202001_Licitação'), [Município] = "RECIFE")

CALCULATE avalia uma expressão em um contexto de filtro modificado.

= CALCULATE(SUM('202001_Licitação'[Valor Licitação]),
FILTER(DISTINCT('202001_Licitação'), [Município] = "RECIFE"))

DASHBOARD

RECURSOS HUMANOS



Colunas Calculadas

Total Gasto por Funcionário =

'Funcionários'[Salario] + 'Funcionários'[Plano de Saúde]

+ 'Funcionários'[Vale Alimentação]

Medidas

01 - Total Salário = SUM('Funcionários'[Salario])

02 - Total Plano de Saúde = SUM('Funcionários'[Plano de Saúde])

03 - Total Alimentação = SUM('Funcionários'[Vale Alimentação])

04 - Gasto Total = [01 - Total Salário] + [02 - Total Plano de Saúde]
+ [03 - Total Alimentação]

05 - Qtd Contratações = COUNTROWS('Funcionários')

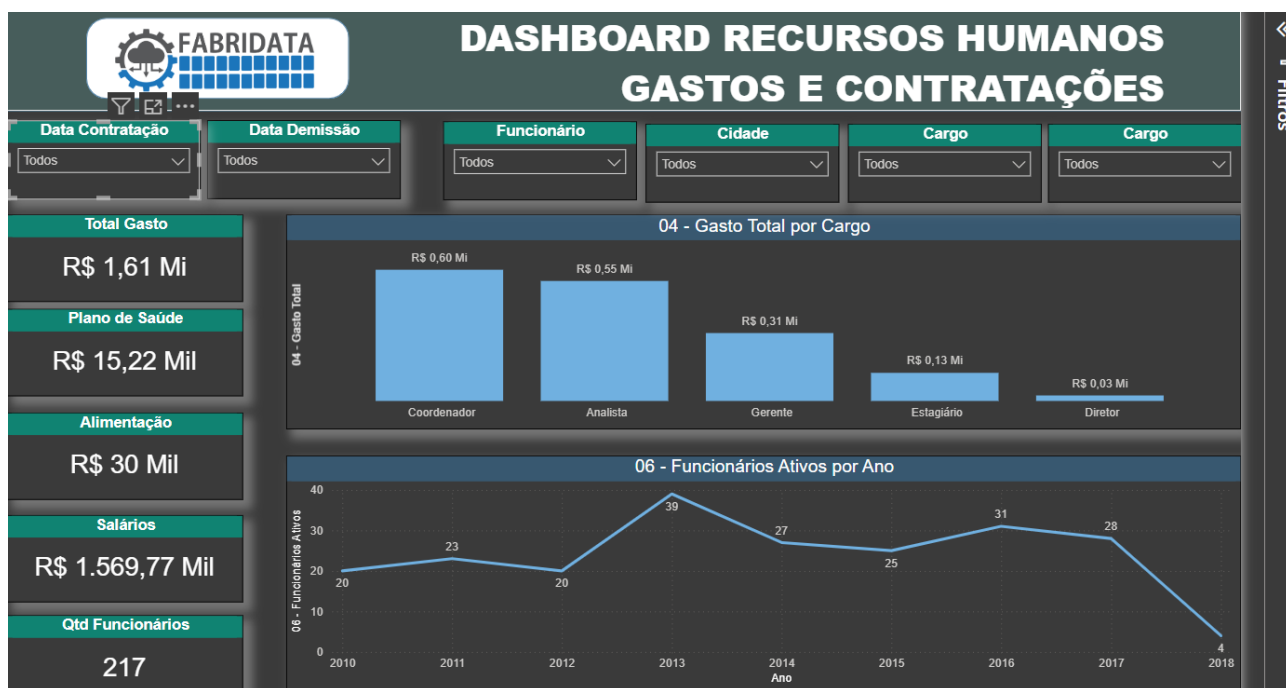
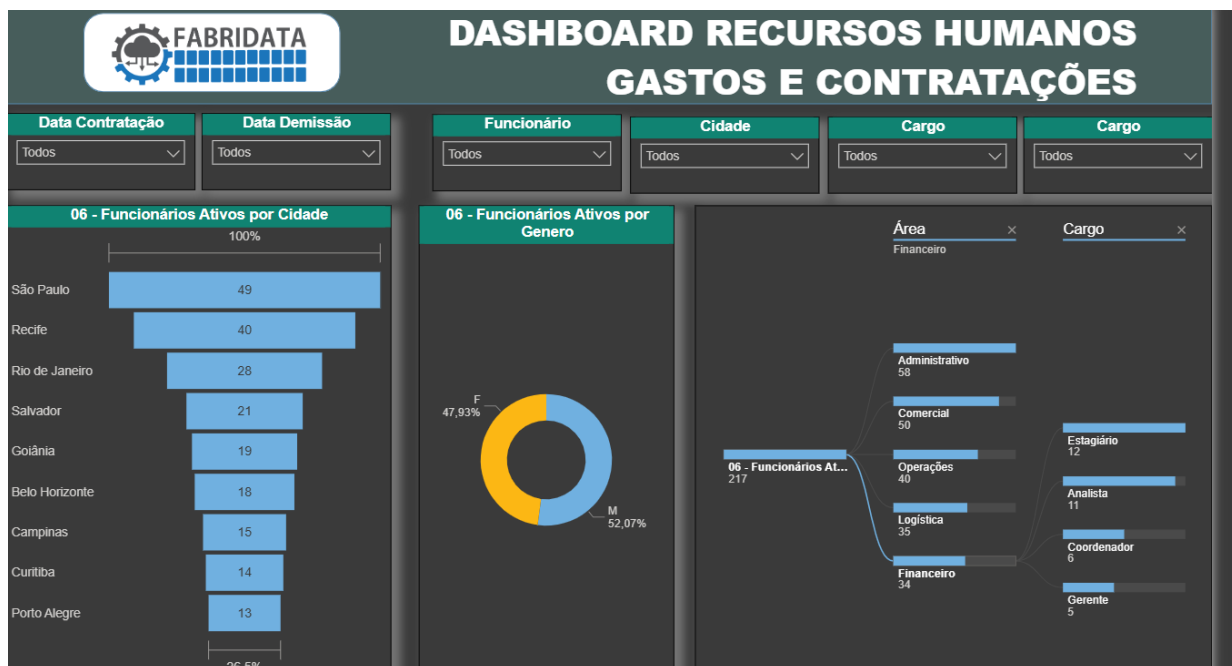
06 - Funcionários Ativos = CALCULATE(COUNTROWS('Funcionários'),
'Funcionários'[Data de Demissao] = BLANK())

07 - 07 - Qtd Demissões = CALCULATE(COUNTROWS('Funcionários'),
'Funcionários'[Data de Demissao] <> BLANK())

08 - 8 - % Turnover = [07 - Qtd Demissões] / [05 - Qtd Contratações]

DASHBOARD

RECURSOS HUMANOS



DASHBOARD VENDAS



Medidas

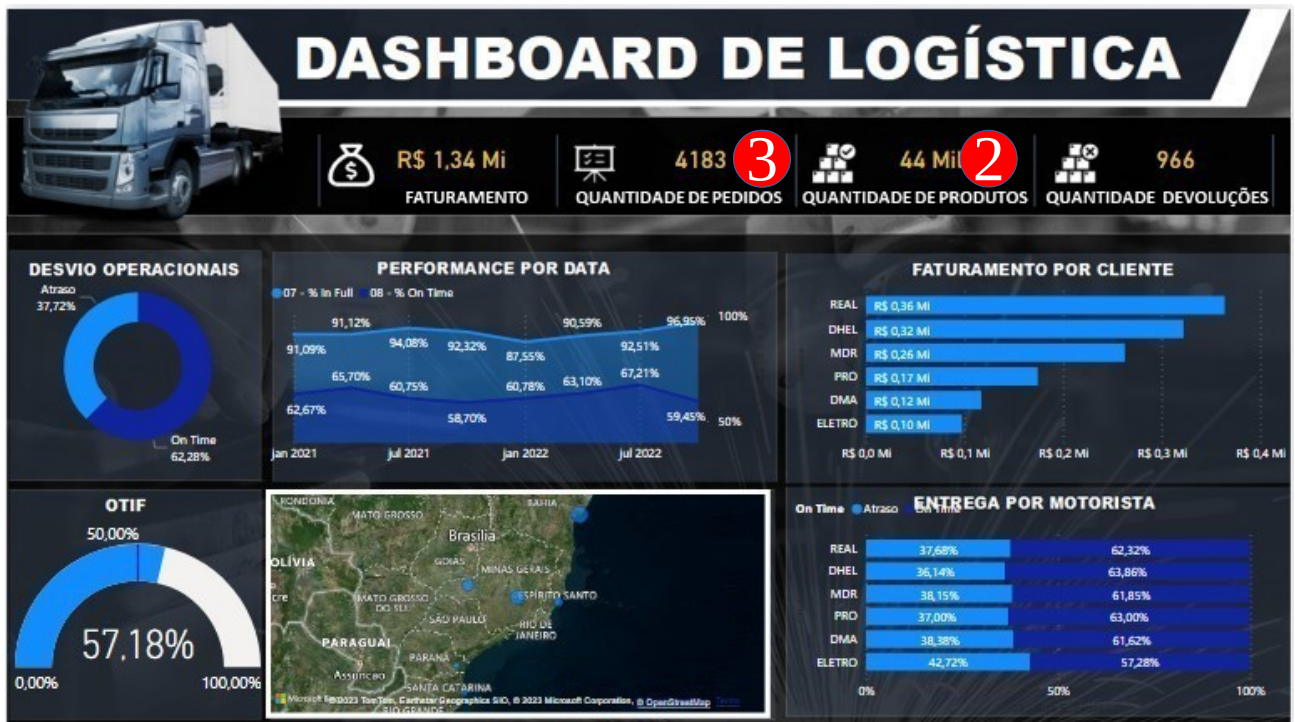
Quantidade de Pedidos = COUNTROWS(pedidos)

Ticket Média = AVERAGE(pedidos[valor_venda])

Total de Vendas = SUM(pedidos[valor_venda])

DASHBOARD

Logística



Formulas DAX

Colunas Calculadas

Total Vendas = Base_Logistica[Quantidade] * Base_Logistica[Valor Pedido]

Medidas

01 - Faturamento = SUM(Base_Logistica[Total Vendas])

02 - Qtd Produtos = SUM(Base_Logistica[Quantidade])

03 - Qtde Entregas = COUNTROWS(Base_Logistica)

04 - Qtd Devoluções = sum((Base_Logistica[Devoluções]))

05 - Qtd On Time = CALCULATE([COUNTROWS(Base_Logistica),Base_Logistica[On Time] = "On Time")

06 - Qtd In Full = CALCULATE(COUNTROWS(Base_Logistica),Base_Logistica[In Full] = "In Full")

07 - % In Full = DIVIDE([06 - Qtd In Full],[03 - Qtde Entregas])

08 - % On Time = DIVIDE([05 - Qtd On Time],[03 - Qtde Entregas])

09 - Meta = 0.50

10 - OTIF = [07 - % In Full] * [08 - % On Time]