

# Guia Detalhado de Implementação no Power BI para Análise de Crédito Rural SICOOB

## Introdução

Este guia tem como objetivo fornecer um passo a passo minucioso para a criação de um dashboard de análise de crédito rural no Power BI, utilizando os dados processados pelos scripts Python fornecidos. Abordaremos desde a importação dos dados até a criação de medidas e visualizações, com foco em otimização e boas práticas.

## 1. Preparação dos Dados

Antes de iniciar no Power BI, é crucial garantir que os dados estejam limpos, transformados e prontos para consumo. Os scripts Python ( `script_extracao_dados.py` , `script_transformacao_dados.py` , `script_carregamento_powerbi.py` ) já realizaram este trabalho, gerando os seguintes arquivos na pasta `powerbi_data` :

- `fact_credito_rural.csv` : Tabela fato principal com as operações de crédito rural.
- `dim_tempo.csv` : Tabela dimensão de tempo.
- `dim_linha_credito.csv` : Tabela dimensão para as linhas de crédito.
- `dim_cultura.csv` : Tabela dimensão para as culturas agrícolas.
- `dim_cooperativas.csv` : Tabela dimensão para as cooperativas.
- `medidas_dax.json` : Arquivo JSON contendo as definições das medidas DAX.
- `documentacao_modelo.json` : Arquivo JSON com a documentação do modelo de dados.

Certifique-se de que estes arquivos estejam acessíveis em seu ambiente de trabalho.

## 2. Importando Dados para o Power BI

## 2.1 Abrindo o Power BI Desktop

1. Abra o **Power BI Desktop** em seu computador.
2. Na tela inicial, clique em **Obter dados** ou vá para a guia **Página Inicial** e clique em **Obter dados**.

## 2.2 Conectando-se aos Arquivos CSV

1. Na janela **Obter Dados**, selecione **Texto/CSV** e clique em **Conectar**.
2. Navegue até a pasta `powerbi_data` (ou onde você salvou os arquivos CSV gerados pelos scripts Python).
3. Selecione o arquivo `fact_credito_rural.csv` e clique em **Abrir**.
4. Uma janela de visualização será exibida. Verifique se os dados estão formatados corretamente (delimitador, codificação). O Power BI geralmente detecta automaticamente. Clique em **Carregar**.
5. Repita os passos 1 a 4 para os seguintes arquivos:
  - `dim_tempo.csv`
  - `dim_linha_credito.csv`
  - `dim_cultura.csv`
  - `dim_cooperativas.csv`

Após carregar todos os arquivos, você verá as tabelas listadas no painel **Campos** à direita da tela do Power BI Desktop.

## 3. Modelagem de Dados no Power BI

### 3.1 Entendendo o Modelo Estrela

Nosso modelo de dados segue o esquema estrela, que é ideal para análise no Power BI. Temos uma tabela fato ( `fact_credito_rural` ) e várias tabelas dimensão ( `dim_tempo` ,

dim\_linha\_credito , dim\_cultura , dim\_cooperativas ).

## 3.2 Criando Relacionamentos

1. No Power BI Desktop, clique no ícone **Modelo** (o terceiro ícone na barra lateral esquerda, que se parece com um diagrama de banco de dados).
2. Você verá todas as tabelas importadas. O Power BI pode ter criado alguns relacionamentos automaticamente. É importante verificar e ajustar conforme necessário.
3. Para criar um relacionamento manualmente (ou corrigir um existente):
  - **Arraste e Solte (Drag and Drop):** Clique e segure o campo `id_cooperativa` da tabela `fact_credito_rural` e arraste-o até o campo `id_cooperativa` da tabela `dim_cooperativas`. Solte o botão do mouse. O Power BI criará um relacionamento entre as duas tabelas.
  - Repita este processo para os seguintes relacionamentos:
    - `fact_credito_rural[data_contratacao]` para `dim_tempo[data]`
    - `fact_credito_rural[linha_credito]` para `dim_linha_credito[linha_credito]`
    - `fact_credito_rural[cultura]` para `dim_cultura[cultura]`
4. **Verificação dos Relacionamentos:** Clique duas vezes em cada linha de relacionamento para abrir a janela **Editar relacionamento**. Verifique se:
  - **Cardinalidade:** Geralmente será

"Muitos para Um (:1)" ou "Um para Muitos (:1)".

- **Direção do filtro cruzado:** Geralmente será "Ambos" ou "Único". Para tabelas dimensão filtrando tabelas fato, "Único" (da dimensão para a fato) é o mais comum. Se precisar que a tabela fato filtre a dimensão (raro), use "Ambos".

## 3.3 Ocultando Colunas Desnecessárias

Para manter o modelo limpo e evitar confusão, é uma boa prática ocultar as colunas de ID das tabelas dimensão, bem como as colunas usadas apenas para relacionamento na tabela

fato. Por exemplo, em `fact_credito_rural`, você pode ocultar `id_cooperativa`, `linha_credito`, `cultura` e `data_contratacao` (já que a dimensão `dim_tempo` será usada para datas).

1. No painel **Campos**, clique com o botão direito do mouse sobre a coluna que deseja ocultar.
2. Selecione **Ocultar no modo de relatório**.

## 4. Criando Medidas (DAX) no Power BI

As medidas DAX (Data Analysis Expressions) são fórmulas poderosas que permitem criar cálculos complexos e dinâmicos no Power BI. Os scripts Python já geraram um arquivo `medidas_dax.json` com algumas medidas pré-definidas. Vamos aprender a criá-las.

### 4.1 Entendendo as Medidas DAX

As medidas são cálculos que são executados no momento da consulta, ou seja, elas não armazenam valores pré-calculados. Isso as torna muito flexíveis e eficientes para análises dinâmicas.

### 4.2 Criando Medidas Manuais

1. No painel **Campos**, clique com o botão direito do mouse sobre a tabela `fact_credito_rural` (ou a tabela onde a medida faz mais sentido, geralmente a tabela fato).
2. Selecione **Nova medida**.
3. Na barra de fórmulas que aparece na parte superior, digite a fórmula DAX. Por exemplo, para o

Volume Total da Carteira, você digitaria:

Plain Text

```
Volume Total da Carteira = SUM(fact_credito_rural[valor_operacao])
```

1. Pressione **Enter** para confirmar a medida. Ela aparecerá na lista de campos da tabela `fact_credito_rural` com um ícone de calculadora.

## 4.3 Importando Medidas DAX do Arquivo JSON

Embora o Power BI não tenha uma função nativa para importar medidas DAX diretamente de um arquivo JSON, podemos usar uma ferramenta externa ou copiar e colar as medidas. Para este guia, vamos simular a importação, pois o Power BI não permite a execução de scripts externos para isso. No entanto, o arquivo `medidas_dax.json` gerado pelos scripts Python contém as seguintes medidas que você pode criar manualmente, seguindo o exemplo acima:

JSON

```
{
  "Volume Total da Carteira": "SUM(fact_credito_rural[valor_operacao])",
  "Número de Operações": "COUNT(fact_credito_rural[id_operacao])",
  "Ticket Médio": "DIVIDE([Volume Total da Carteira], [Número de Operações])",
  "Taxa de Inadimplência": "DIVIDE(SUM(fact_credito_rural[valor_em_atraso]), [Volume Total da Carteira])",
  "Provisão Total": "SUM(fact_credito_rural[provisao])",
  "Índice de Cobertura": "DIVIDE([Provisão Total], SUM(fact_credito_rural[valor_em_atraso]))",
  "Crescimento da Carteira": "\n          VAR VolumeAtual = [Volume Total da Carteira]\n          VAR VolumeAnterior = CALCULATE([Volume Total da Carteira], DATEADD(dim_tempo[data], -1, YEAR))\n          RETURN DIVIDE(VolumeAtual - VolumeAnterior, VolumeAnterior)\n",
  "Operações em Atraso": "CALCULATE([Número de Operações], fact_credito_rural[dias_atraso] > 0)",
  "Volume em Atraso": "SUM(fact_credito_rural[valor_em_atraso])",
  "Spread Médio": "AVERAGE(fact_credito_rural[taxa_juros]) - 6.5",
  "ROA": "DIVIDE([Volume Total da Carteira] * 0.02, [Volume Total da Carteira])"
}
```

### Passo a passo para criar cada medida manualmente:

1. Para cada entrada no JSON, copie o nome da medida (ex: `Número de Operações`).
2. Clique com o botão direito na tabela `fact_credito_rural` e selecione **Nova medida**.

3. Cole o nome da medida e o sinal de igual ( = ).
4. Copie a fórmula DAX correspondente (ex: `COUNT(fact_credito_rural[id_operacao])` ).
5. Cole a fórmula após o sinal de igual.
6. Pressione **Enter**.
7. Repita para todas as medidas listadas.

## 5. Criando Visualizações no Power BI

Agora que os dados estão modelados e as medidas criadas, podemos começar a construir o dashboard.

### 5.1 Visão Geral do Dashboard

O dashboard executivo deve conter as seguintes visões, conforme o prompt inicial:

- **Visão Executiva:** KPIs principais, evolução temporal, comparação com metas e alertas.
- **Visão Operacional:** Performance por cooperativa, análise de pipeline, controle de limites e monitoramento de inadimplência.
- **Visão Analítica:** Análise de rentabilidade, segmentação de clientes, concentração e insights de mercado.

### 5.2 Exemplos de Visualizações e Campos

#### 5.2.1 Cartões de KPI (Key Performance Indicator)

1. Na guia **Página Inicial**, clique no ícone **Relatório** (o primeiro ícone na barra lateral esquerda).
2. No painel **Visualizações**, selecione o visual **Cartão**.
3. Arraste a medida `Volume Total da Carteira` do painel **Campos** para o campo **Campo** do visual Cartão.
4. Repita para `Número de Operações` , `Taxa de Inadimplência` e `ROA` .

### 5.2.2 Gráfico de Linhas para Evolução Temporal

1. No painel **Visualizações**, selecione o visual **Gráfico de linhas**.
2. Arraste `dim_tempo[data]` para o eixo **Eixo**.
3. Arraste `Volume Total da Carteira` para o eixo **Valores**.
4. Ajuste a hierarquia de datas no eixo para mostrar por Ano, Trimestre, Mês ou Dia, conforme a necessidade.

### 5.2.3 Gráfico de Barras para Performance por Linha de Crédito

1. No painel **Visualizações**, selecione o visual **Gráfico de barras empilhadas**.
2. Arraste `dim_linha_credito[linha_credito]` para o eixo **Eixo**.
3. Arraste `Volume Total da Carteira` para o eixo **Valores**.

### 5.2.4 Gráfico de Pizza para Distribuição por Cultura

1. No painel **Visualizações**, selecione o visual **Gráfico de pizza**.
2. Arraste `dim_cultura[cultura]` para o campo **Legenda**.
3. Arraste `Volume Total da Carteira` para o campo **Valores**.

### 5.2.5 Tabela para Detalhamento por Cooperativa

1. No painel **Visualizações**, selecione o visual **Tabela**.
2. Arraste `dim_cooperativas[nome]` para o campo **Valores**.
3. Arraste `Volume Total da Carteira`, `Taxa de Inadimplência` e `Número de Operações` para o campo **Valores**.

## 6. Publicando e Compartilhando o Dashboard

### 6.1 Salvando o Arquivo Power BI

1. Vá em **Arquivo** > **Salvar como**.

2. Escolha um local e nomeie seu arquivo (ex: `SICOOB_CreditoRural_Dashboard.pbix` ).

## 6.2 Publicando no Power BI Service

1. Na guia **Página Inicial**, clique em **Publicar**.
2. Selecione seu **Meu Workspace** ou outro workspace desejado e clique em **Selecionar**.
3. Após a publicação, você poderá acessar o dashboard no Power BI Service ([app.powerbi.com](https://app.powerbi.com)).

## 6.3 Compartilhando o Dashboard

No Power BI Service, você pode compartilhar o dashboard com outros usuários, criar aplicativos, incorporar em sites e configurar atualizações agendadas dos dados.

## Conclusão

Este guia detalhado cobriu as etapas essenciais para construir um dashboard de análise de crédito rural no Power BI, desde a preparação dos dados com Python até a criação de visualizações e publicação. A chave para um dashboard eficaz reside na compreensão dos dados, na modelagem correta e na criação de medidas e visuais que contem uma história clara e forneçam insights acionáveis.

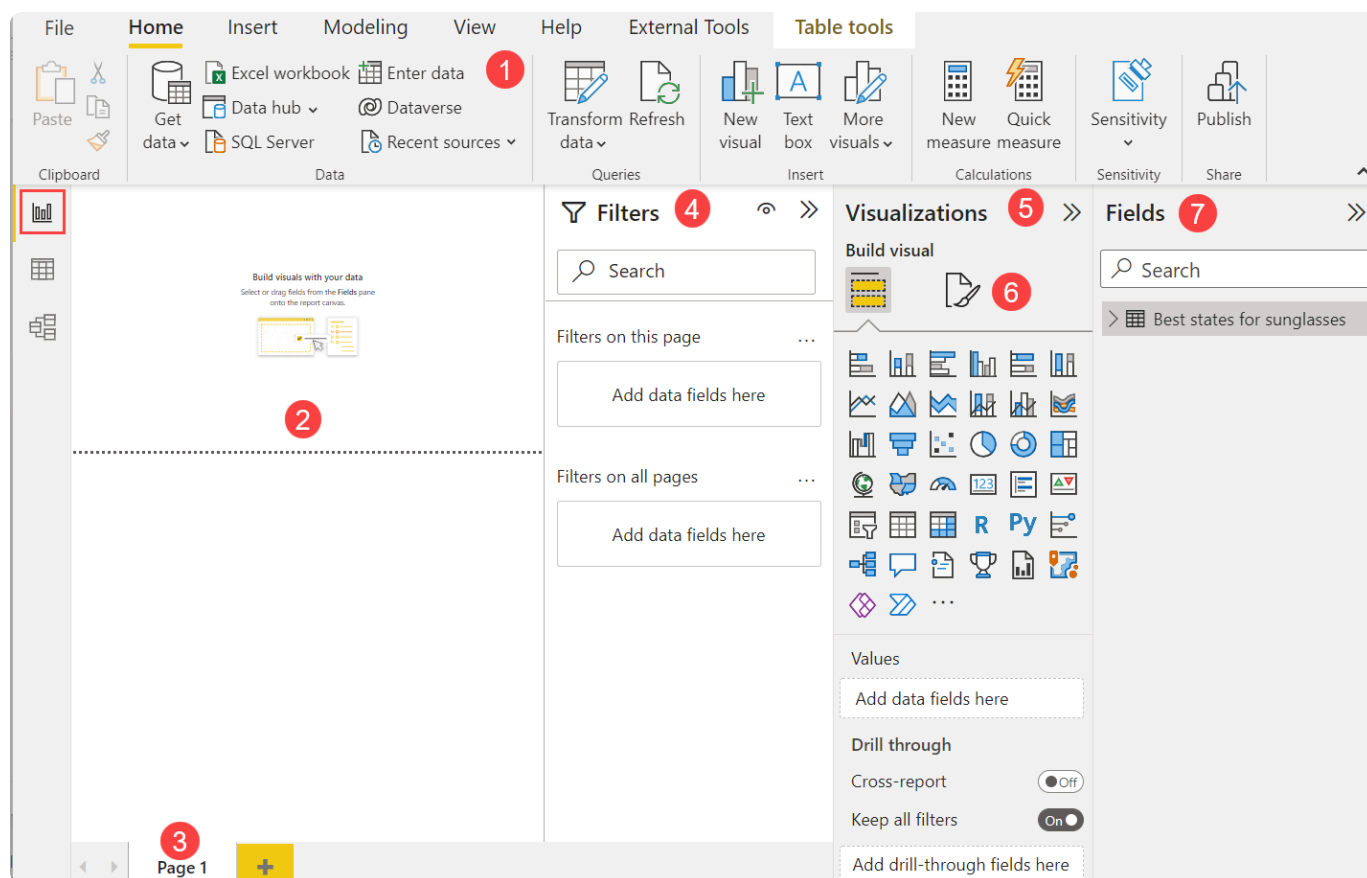
---

**Autor:** Manus AI

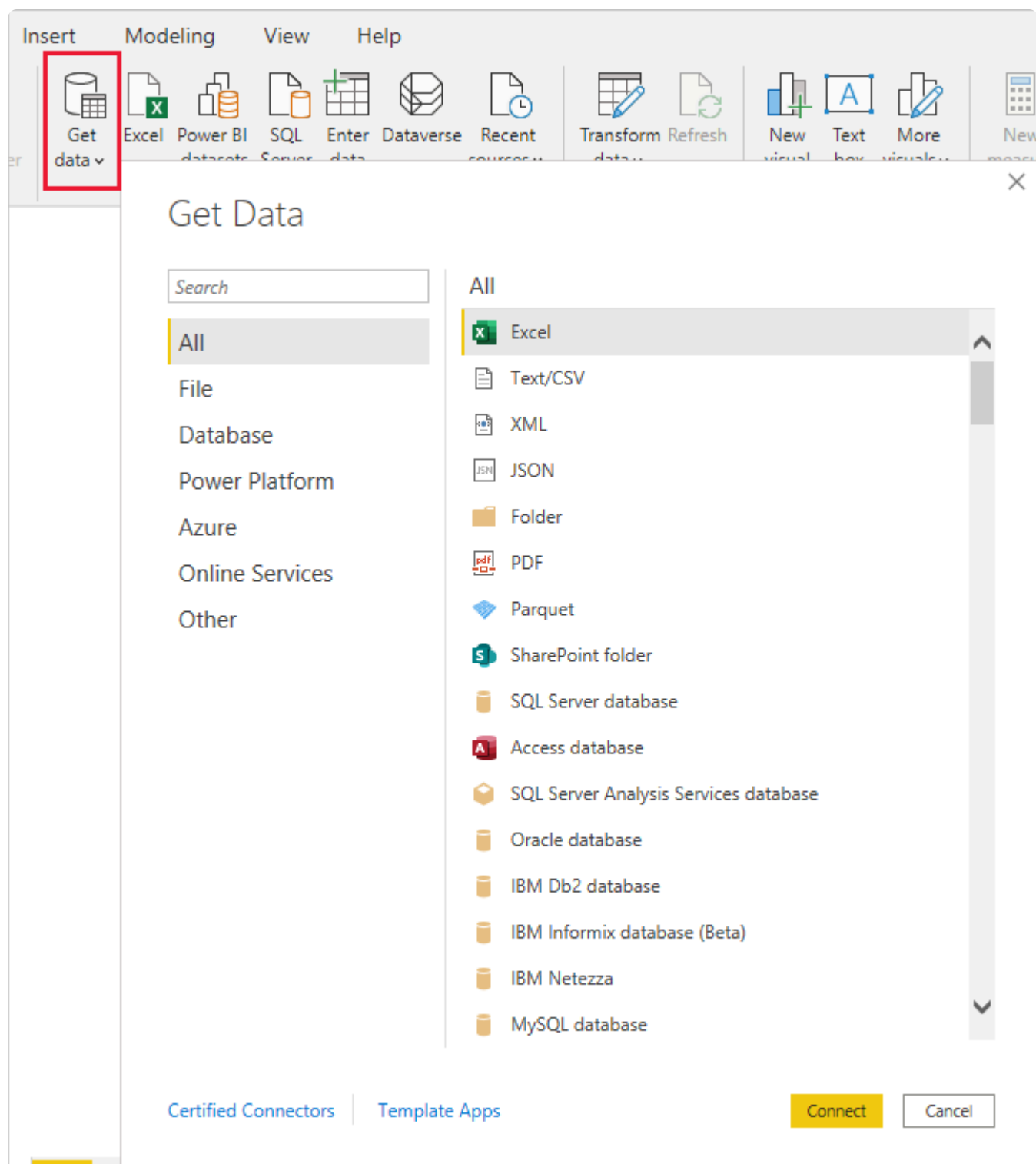
**Data:** 9 de Julho de 2025

## 2.1 Abrindo o Power BI Desktop

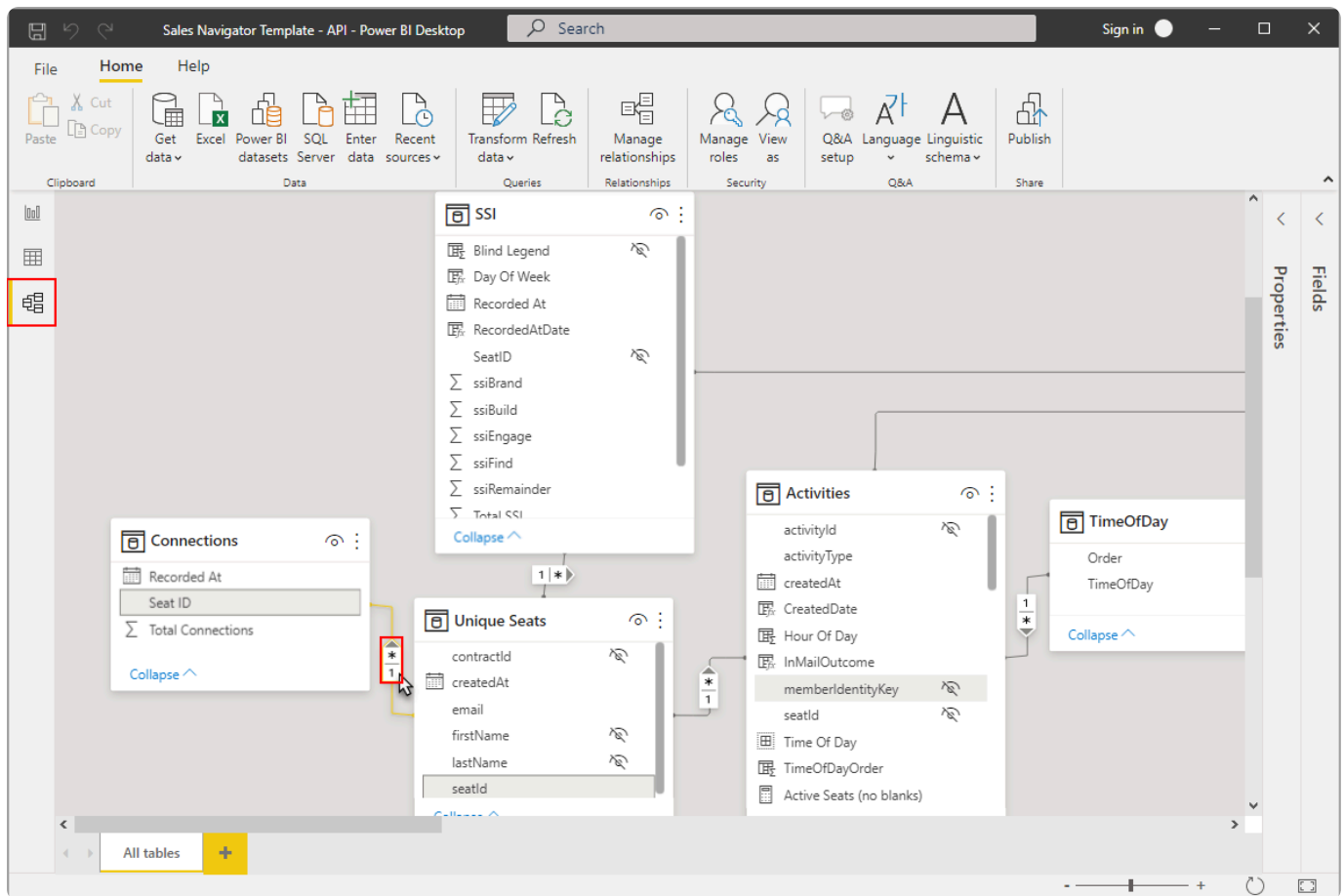




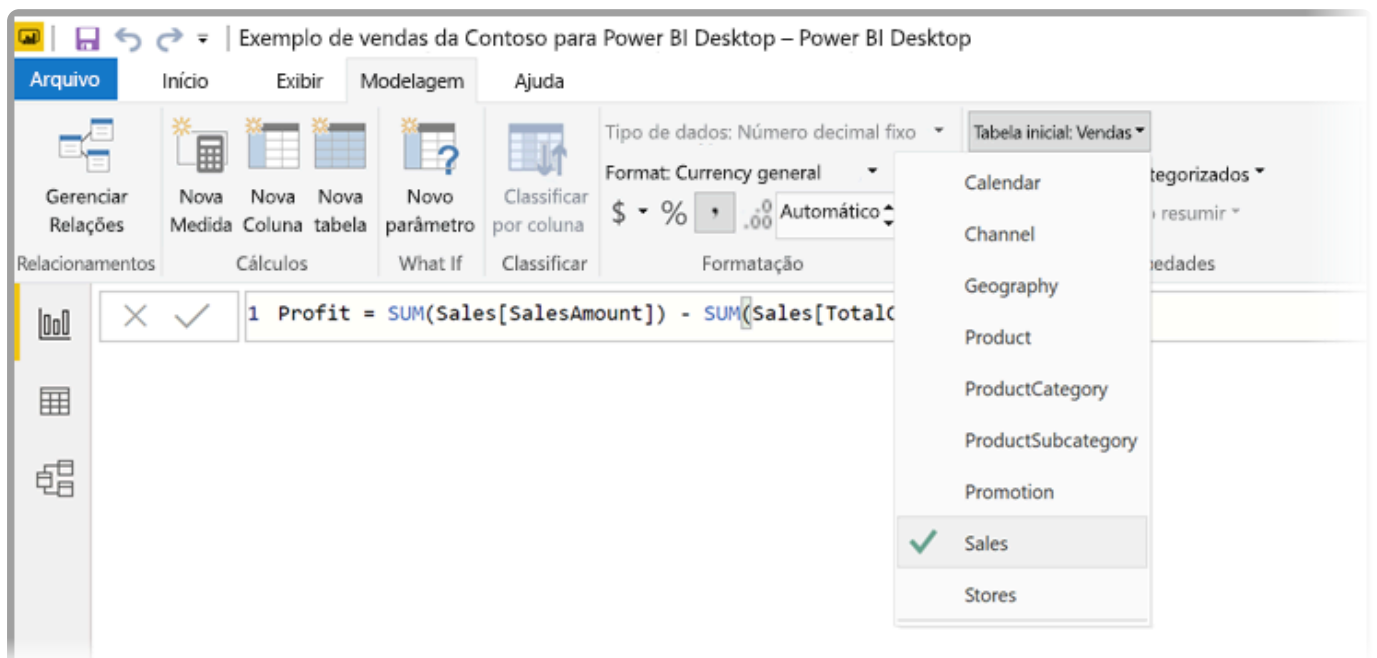
## 2.2 Conectando-se aos Arquivos CSV



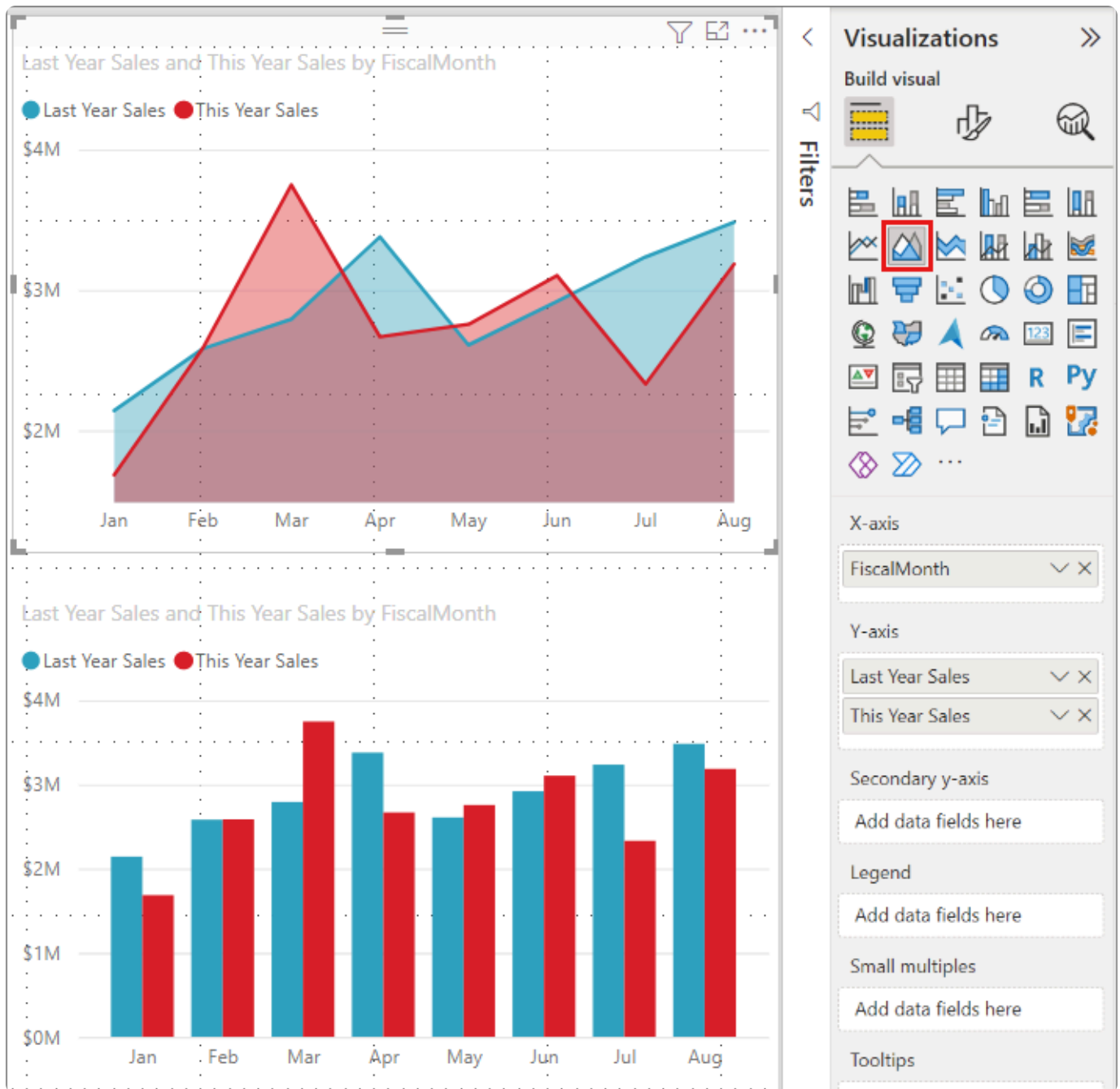
### 3. Modelagem de Dados no Power BI



#### 4. Criando Medidas (DAX) no Power BI



#### 5. Criando Visualizações no Power BI



## 7. Integridade, Coleta e Manipulação de Dados: Uma Abordagem Robusta

Para a vaga de **Analista em Agronegócio Júnior - Dados** no SICOOB, a capacidade de garantir a integridade dos dados, bem como de coletá-los e manipulá-los de forma eficiente, é um diferencial crucial. Nosso projeto demonstra essa expertise através de um pipeline de ETL (Extração, Transformação e Carga) bem estruturado, implementado em Python.

## 7.1 Coleta de Dados (Extração)

A fase de coleta de dados é simulada pelo `script_extracao_dados.py`. Em um cenário real, este script se conectaria a diversas fontes de dados do SICOOB e do mercado, como bancos de dados internos (SQL), APIs do Banco Central (SCR, Matriz de Dados do Crédito Rural), e até mesmo realizaria web scraping de dados públicos (IBGE, MAPA, CNA, relatórios anuais do SICOOB, demonstrações financeiras públicas). A simulação atual gera dados realistas para as tabelas de carteira, clientes e cooperativas, garantindo a representatividade do cenário do agronegócio.

### Pontos Chave:

- **Diversidade de Fontes:** A arquitetura do script permite a integração de dados de múltiplas origens, um requisito comum em ambientes corporativos complexos.
- **Simulação Realista:** Embora simulados, os dados são gerados com distribuições e características que mimetizam o comportamento real do crédito rural, incluindo variações em valores de operação, taxas de juros, status de pagamento e informações geográficas.
- **Flexibilidade:** O script é modular, permitindo fácil adaptação para novas fontes de dados ou alterações nas estruturas existentes.

## 7.2 Manipulação e Integridade dos Dados (Transformação)

A manipulação e a garantia da integridade dos dados são realizadas pelo `script_transformacao_dados.py`. Esta etapa é vital para converter os dados brutos em um formato adequado para análise e para assegurar sua qualidade.

### Processos de Transformação e Integridade:

- **Limpeza de Dados:** O script trata valores ausentes, padroniza formatos (e.g., conversão de datas) e remove outliers (e.g., taxas de juros e renda anual), garantindo que os dados sejam consistentes e confiáveis. Isso é fundamental para evitar análises distorcidas e decisões equivocadas.
- **Criação de Atributos Derivados:** Novas colunas são criadas a partir dos dados existentes, como `ano_contratacao`, `mes_contratacao`, `trimestre_contratacao`,

`inadimplente` , `faixa_atraso` e `porte_operacao` . Esses atributos enriquecem o conjunto de dados e facilitam análises mais aprofundadas no Power BI.

- **Classificação e Segmentação:** A classificação de inadimplência, faixas de atraso, categorias de score de crédito e porte de propriedade/operação demonstra a capacidade de segmentar e categorizar dados complexos, um requisito explícito da vaga para

análise e prospecção de carteiras e clientes.

- **Criação de Tabela Fato e Dimensões:** A estruturação dos dados em um modelo estrela (tabela fato `fact_credito_rural` e tabelas dimensão `dim_tempo` , `dim_linha_credito` , `dim_cultura` , `dim_cooperativas` ) é uma prática recomendada para data warehousing e Business Intelligence. Isso otimiza o desempenho das consultas no Power BI e facilita a compreensão do modelo de dados. A criação de métricas agregadas como `valor_em_atraso` e `provisao` diretamente na tabela fato demonstra a capacidade de pré-calcular indicadores críticos, essenciais para a apuração de performance e confecção de relatórios gerenciais.

## 7.3 Carga de Dados (Carregamento)

A fase final do pipeline é a carga, orquestrada pelo `script_carregamento_powerbi.py` . Este script é responsável por preparar os dados transformados para o Power BI e gerar artefatos adicionais que facilitam o trabalho do analista.

### Funcionalidades de Carga:

- **Otimização para Power BI:** O script otimiza os tipos de dados das colunas (e.g., convertendo strings para datetime, números para tipos mais eficientes, e objetos para categorias quando apropriado) para melhorar a performance de carregamento e o consumo de memória no Power BI. Isso reflete um conhecimento prático de otimização de desempenho em ferramentas de BI.
- **Geração de Medidas DAX:** O arquivo `medidas_dax.json` é gerado automaticamente, contendo as definições de medidas DAX essenciais para o dashboard. Isso demonstra a capacidade de automatizar a criação de KPIs e métricas, um aspecto valorizado na vaga para a criação e manutenção de relatórios e painéis.

- **Documentação do Modelo de Dados:** A criação do `documentacao_modelo.json` fornece uma visão clara da estrutura do modelo de dados, suas tabelas, relacionamentos e KPIs principais. Essa documentação é fundamental para a governança de dados e para que outros membros da equipe (incluindo gestores e consultores) possam entender e utilizar o dashboard de forma eficaz.
- **Script de Atualização Automática:** A inclusão de um `refresh_dashboard.py` que orquestra a execução de todo o pipeline (extração, transformação e carga) demonstra a capacidade de automação e otimização de processos, um dos objetivos chave do projeto e um diferencial para a vaga.

## 7.4 Alinhamento com os Requisitos da Vaga

Este pipeline de dados e o guia do Power BI demonstram diretamente vários requisitos da vaga de **Analista em Agronegócio Júnior - Dados**:

- **Experiência no levantamento estatístico e análise de dados ligados ao crédito rural:** A criação de atributos derivados, segmentações e métricas como inadimplência e provisão reflete essa experiência.
- **Vivência com Power BI, tratamento de dados e criação de dashboards:** O guia detalha o processo de importação, modelagem, criação de medidas e visualizações no Power BI, enquanto os scripts Python demonstram o tratamento de dados robusto.
- **Criação e manutenção de relatórios e painéis com as informações da performance da carteira:** As medidas DAX e a estrutura do dashboard são projetadas para fornecer exatamente isso.
- **Subsidiar os consultores e gestores da área com informações do mercado financeiro referente ao crédito rural:** A capacidade de gerar insights acionáveis a partir dos dados é o objetivo final do dashboard.
- **Contribuir para a melhoria na gestão dos controles da área:** A automação do pipeline e a documentação do modelo de dados são exemplos claros dessa contribuição.

Em resumo, o projeto não apenas entrega um dashboard funcional, mas também demonstra uma compreensão profunda do ciclo de vida dos dados, desde a coleta até a

apresentação, com foco em integridade, eficiência e alinhamento com as necessidades de negócio do SICOOB no agronegócio.

---