

# Sinergia de Dados: Excel para Power BI através do Ciclo ETL e SQL

Esta documentação tem como objetivo apresentar a arquitetura do serviço "**Sinergia de Dados: Excel para Power BI através do Ciclo ETL e SQL**", de forma clara e objetiva, a fim de simplificar a compreensão de todos os seus componentes e funcionamento. Esse serviço é responsável por fornecer soluções em Business Intelligence, atendendo os interessados em aprender mais sobre o ciclo de vida dos dados.

Ao longo deste documento, serão apresentados detalhes técnicos sobre o serviço, sua estruturação e como cada parte se relaciona com as demais. A documentação será útil tanto para os desenvolvedores que trabalham no serviço, quanto para aqueles que desejam entender melhor como a tecnologia pode ser aplicada em seu setor.

Além disso, a documentação foi projetada para atender às necessidades de diferentes públicos, desde os que possuem um conhecimento mais técnico em tecnologia até aqueles que possuem conhecimento limitado nessa área.

## Tópicos

- **Primeiro Contato**
  - Cenário e Desafios
  - Ferramentas
- **Excel**
  - Info das Planilhas
- **ETL e SQL**
  - Criação dos Serviços no Visual Studio - (Database, SSIS e SSRS)
  - Criação das Tabelas
  - Deploy no SQL Server
  - ETL
- **BI**
  - Dashboard
  - Report
  - Deploy Reports e Dashboards
  - Incorporar Reports e Dashboards

- **Arquitetura**
  - Exemplo Conceitual da Arquitetura

## 1. Primeiro Contato

### 1.1. Cenário e Desafios

Uma empresa de médio porte, possui uma operação em crescimento e está buscando maneiras de otimizar seus processos de gerenciamento de recursos humanos e análise de dados. Atualmente, eles confiam em planilhas do Excel para analisar e compartilhar informações sobre os funcionários, o que resulta em dificuldades na integridade dos dados e na colaboração eficiente.

Eles decidiram colaborar com a empresa de consultoria em Business Intelligence, Data Ape, para melhorar sua abordagem de gerenciamento de dados e permitir análises mais eficazes. O objetivo principal é realizar um "Proof of Concept" (POC) para demonstrar como a transformação de processos usando BI pode trazer eficiência, insights mais profundos e capacidade de tomada de decisões informadas.

### 1.2. Ferramentas

As **ferramentas utilizadas são:**

- Excel;
- Visual Studio 2019 - (SSIS / Database Project / SSRS);
- SQL Server;
- Power BI;
- Power BI Report Server.

## 2. Excel

### 2.1. Planilhas

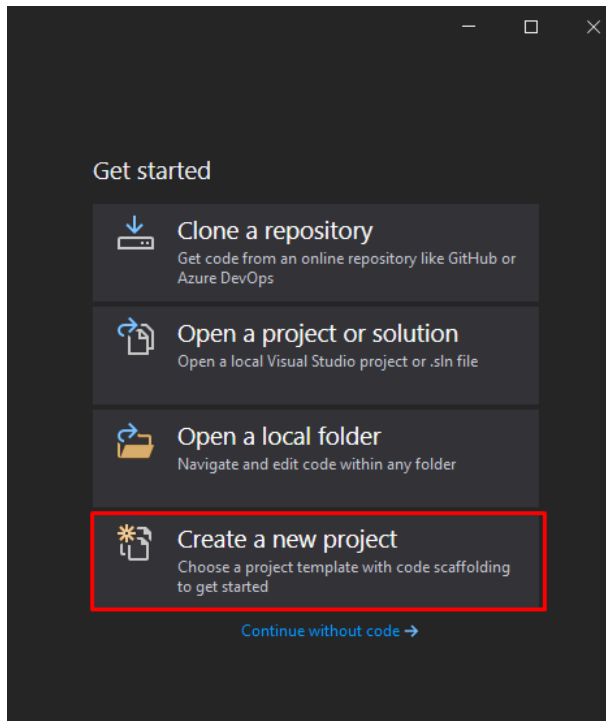
Foram encaminhadas duas planilhas:

- **Cargos.xlsx** - Planilha que possui as informações de todos os cargos da empresa e seu respectivo código
- **Funcionarios.xlsx** - Planilha que possui todos os funcionários que passaram pela empresa e as informações completas de cada funcionário

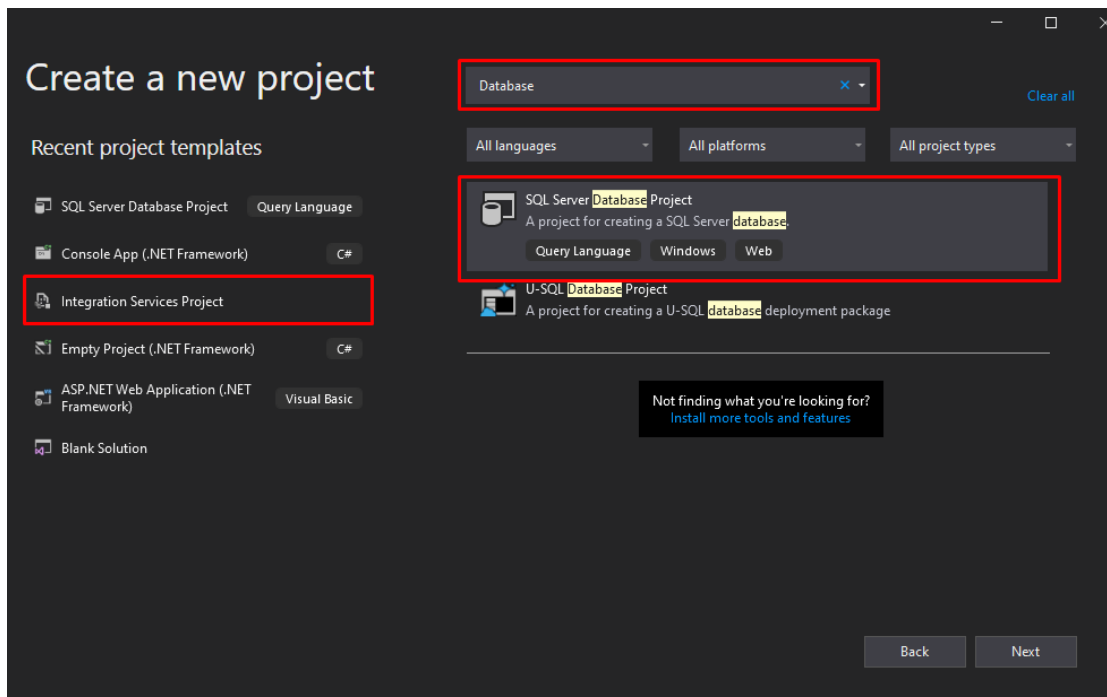
## 3. ETL e SQL

### 3.1. Criação dos Serviços no Visual Studio - (Database, SSIS e SSRS)

Para iniciar, vamos abrir o Visual Studio (*Estou usando a Versão 2019*) e clicar em "Create New Project".



Vamos iniciar pelo **Database Project**, basta pesquisar e selecionar:



Eu costumo adicionar o nome que será utilizado no banco de dados, mas você pode deixar como Database apenas. Depois de adicionar o nome, clique em "Create"

Configure your new project

SQL Server Database Project   Query Language   Windows   Web

Project name  
people\_analytics

Location  
C:\0 - Personal\portfolio\projects\0\

Solution name ⓘ  
people\_analytics

☐ Place solution and project in the same directory

Back   Create

Para adicionar as soluções do SSIS e SSRS, basta: botão **direito na solução >> Add >> New Project**. Pesquise o projeto, adicione o nome e crie.

Add a new project

Integration Services   Clear all

All languages   All platforms   All project types

Recent project templates

- SQL Server Database Project   Query Language
- Console App (.NET Framework)   C#
- Integration Services Project
- Empty Project (.NET Framework)   C#
- ASP.NET Web Application (.NET Framework)   Visual Basic

Integration Services Project  
This project may be used for building high performance data integration and workflow solutions that can be run on SSIS catalog, including extraction, transformation, and loading (ETL) operations for data warehousing.

Integration Services Project (Azure-Enabled)  
This project may be used for building high performance data integration and workflow solutions that can also be run/debugged on SSIS Platform-as-a-Service (PaaS) in Azure Data Factory.

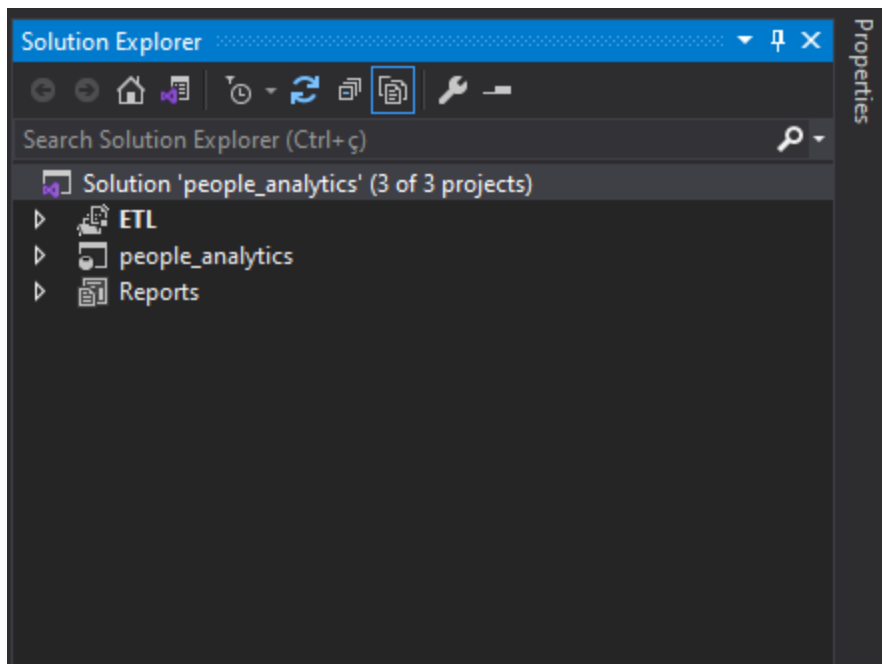
Integration Services Import Project Wizard  
A wizard that assists you in creating a new Integration Services (SSIS) project that is based on an existing one. Import from a project deployment file (.ispac extension) or from an Integration Services catalog on an instance of SQL Server.

Windows Service (.NET Framework)  
A project for creating Windows Services  
C#   Windows   Desktop   Service

Windows Service (.NET Framework)  
A project for creating Windows Services

Next

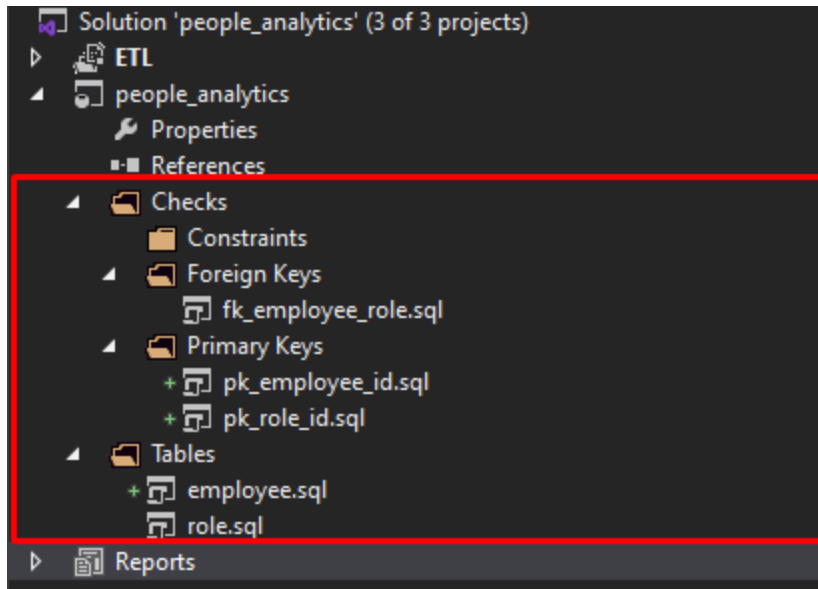
Fechamos a construção das solução no seguinte modelo:



### 3.2. Criação das Tabelas

Vamos deixar a estrutura das tabelas prontas em nossa solução, para isso, precisamos analisar os campos das planilhas e criar tabelas referenciando esses campos.

Eu gosto de organizar o projeto do banco de dados em pastas e dentro das pastas adiciono o que preciso. Exemplo:



Vamos realizar a criação das tabelas e chaves:

- **Tabela Funcionário:**

```
2 CREATE TABLE [dbo].[employee]
3 (
4     [id] [int] NOT NULL, -- ID do Funcionário
5     [name] [nvarchar](100) NOT NULL, -- Nome do Funcionário
6     [gender] [nvarchar](25) NOT NULL, -- Sexo do Funcionário
7     [hire_date] [date] NOT NULL, -- Data de Admissão
8     [role_id] [int] NOT NULL, -- ID do Cargo
9     [role_date] [date] NOT NULL, -- Data do Cargo
10    [contract_id] [int] NOT NULL, -- ID do Contrato
11    [contract_type] [nvarchar](50) NOT NULL, -- Tipo de Contrato
12    [marital_status_id] [int] NOT NULL, -- ID do Estado Civil
13    [marital_status] [nvarchar](50) NOT NULL, -- Estado Civil
14    [education_id] [int] NOT NULL, -- ID da Escolaridade
15    [education] [nvarchar](50) NOT NULL, -- Escolaridade
16    [birth_date] [date] NOT NULL, -- Data de Nascimento
17    [nationality_id] [int] NOT NULL, -- ID da Nacionalidade
18    [nationality] [nvarchar](50) NOT NULL, -- Nacionalidade
19    [race_id] [int] NOT NULL, -- ID da Raça
20    [race] [nvarchar](50) NOT NULL, -- Raça
21    [salary] [money] NOT NULL, -- Raça
22    [status_id] [int] NOT NULL, -- ID da Situação
23    [status] [nvarchar](50) NOT NULL, -- Situação
24    [absence_cause] [nvarchar](50) NOT NULL, -- Motivo do Afastamento
25    [absence_date] [date] NULL, -- Data do Afastamento
26    [work] [nvarchar](50) NOT NULL -- Forma de Trabalho
27 )
```

-

- **Tabela Cargos:**

```
3 CREATE TABLE [dbo].[role]
4 (
5     [id] [int] NOT NULL, -- Id do Cargo
6     [name] [nvarchar](100) NOT NULL -- Nome do Cargo
7 )
```

*Obs: Como é um POC vamos realizar um modelo simples, espelhando apenas as tabelas que recebemos, mas através da tabela de funcionário podemos identificar algumas tabelas de dimensão.*

- **Chaves Primárias**

```
2 -- Tabela: role
3 ALTER TABLE [dbo].[role]
4     ADD CONSTRAINT [pk_role_id]
5     PRIMARY KEY ([id]);
```

```
1 -- Tabela: employee
2 ALTER TABLE [dbo].[employee]
3     ADD CONSTRAINT [pk_employee_id]
4     PRIMARY KEY ([id]);
```

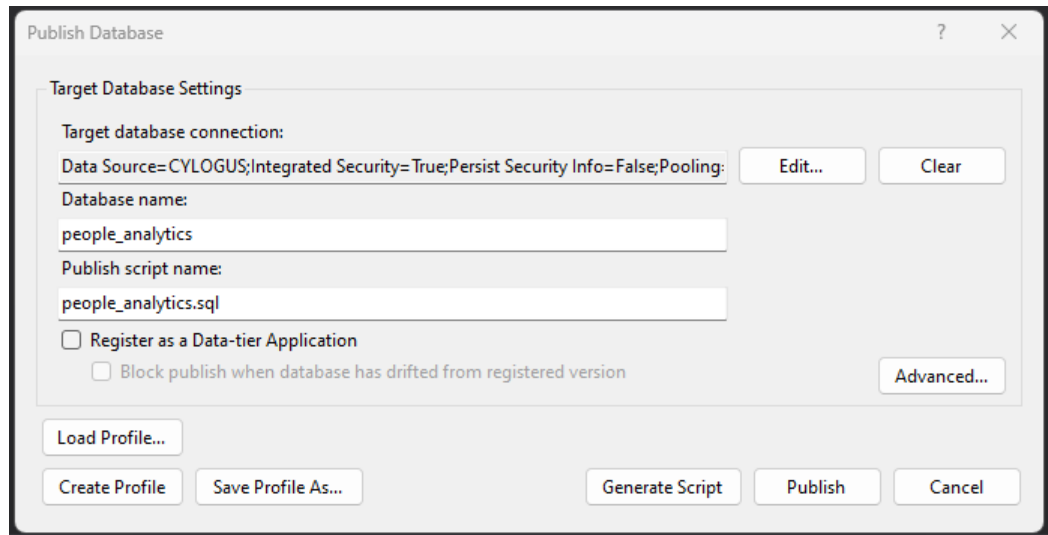
- **Chaves Estrangeiras**

```
1 -- Chave Estrangeira para Role
2 ALTER TABLE [dbo].[employee]
3     ADD CONSTRAINT [fk_employee_role]
4     FOREIGN KEY ([role_id])
5     REFERENCES [dbo].[role] ([id]);
```

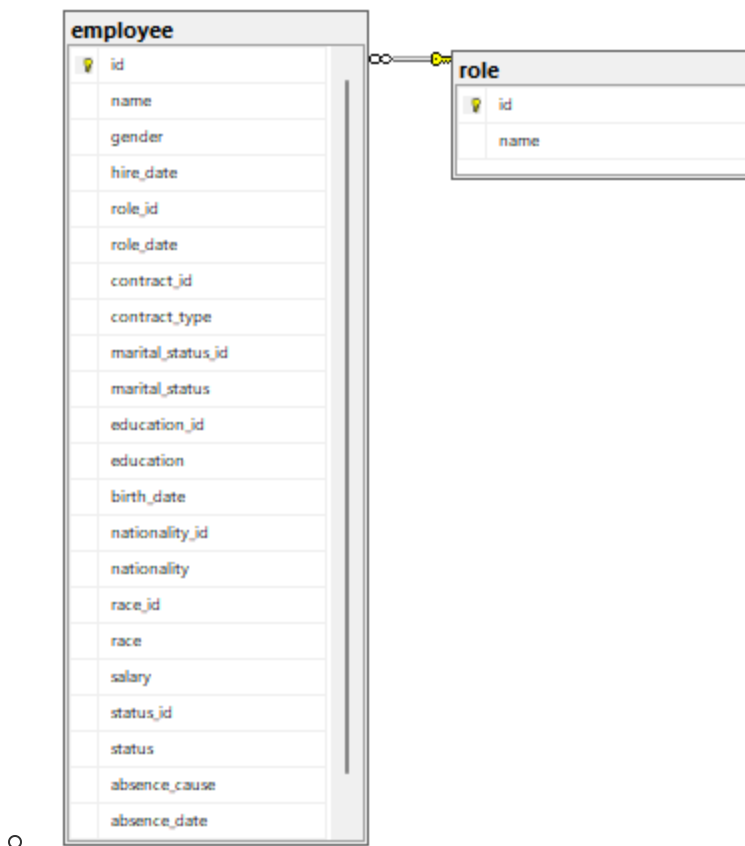
### 3.3. Deploy no SQL Server

Para realizar o Deploy no SQL Server, vamos seguir os seguintes passos:

- **Botão Direito na solução do banco de dados >> Publish** (Ele vai realizar um Build, se der algum erro não será possível publicar)
- **Target Database Connection** (Você configura a conexão com o SQL)
- **Database Name** (Nome do banco de dados)
- **Publicação**
  - **Generate Script** (Será gerado um Script de publicação)
  - **Publish** (Irá publicar direto)



Após a publicação, verifique o **SQL Server**.



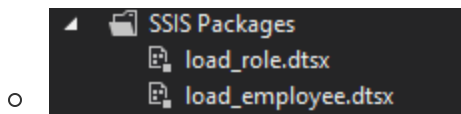
### 3.4. ETL

Vamos realizar dois pacotes:

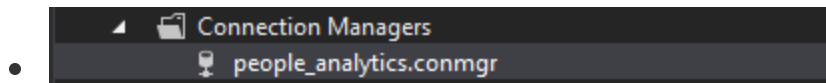
- load\_employee (Carrega a tabela de funcionários)



- load\_role (Carrega a tabela de cargos)

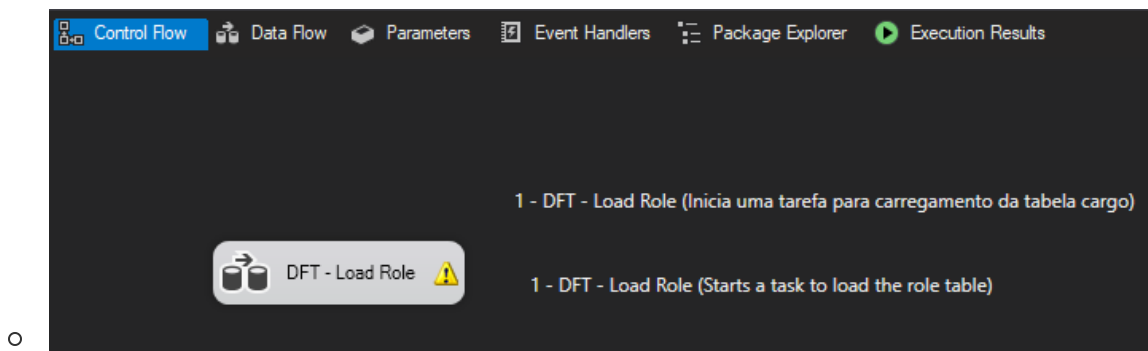


Conexão oficial para todos os pacotes:

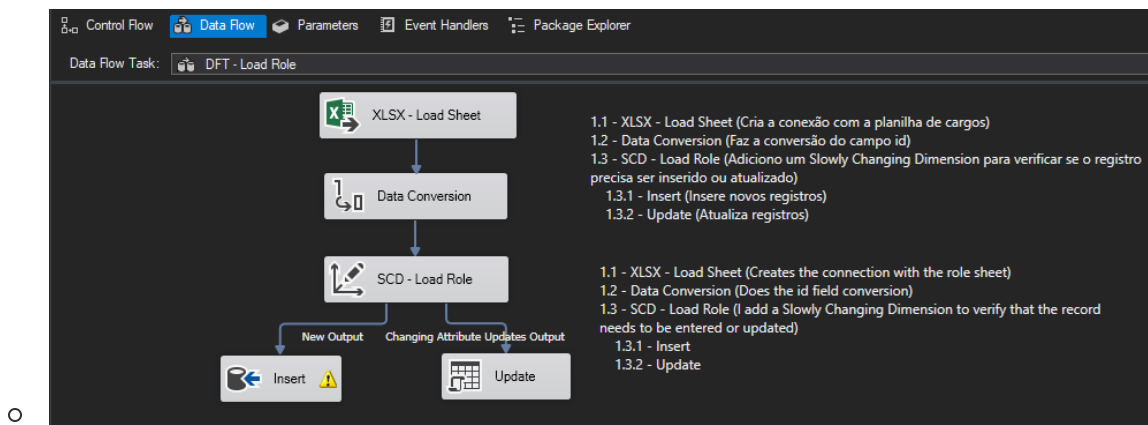


**load\_role** - Foquei em uma abordagem mais prática e direta, para evitar erros adicionei um SCD direto da planilha.

- **Control Flow**

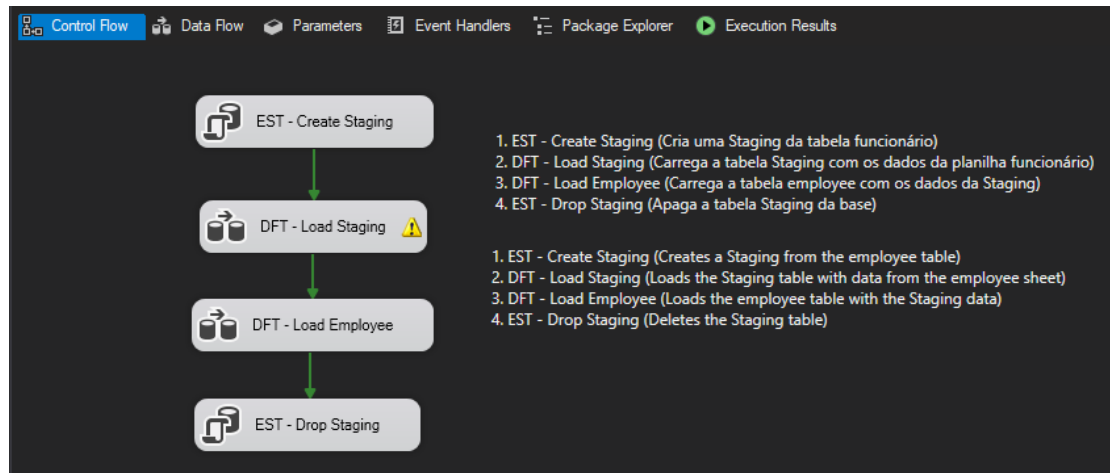


- **Data Flow**

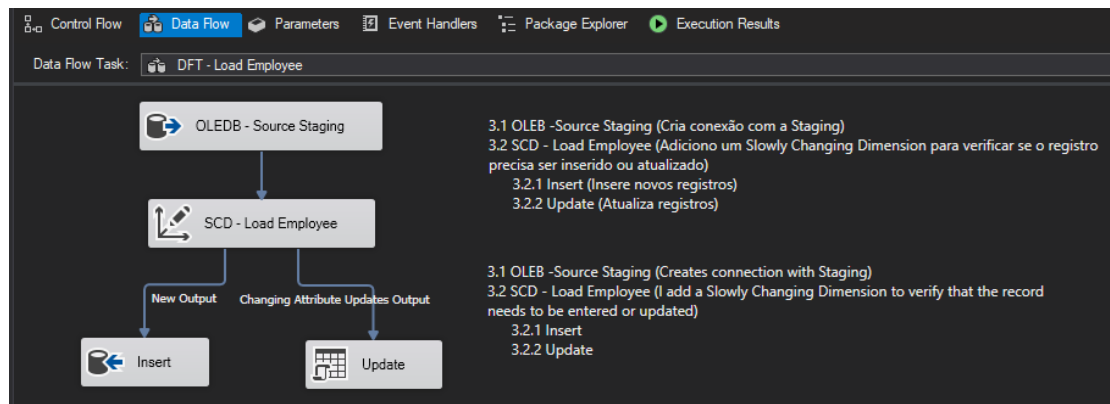
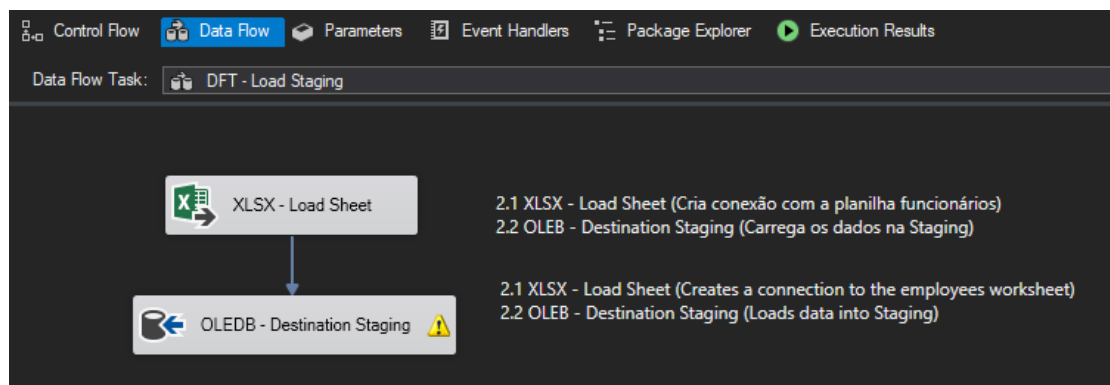


**load\_employee** - Apliquei uma abordagem que gosto bastante de utilizar e acredito ser muito funcional (não tanto nesse caso, mas queria compartilhar). Eu crio uma Staging no começo do Job e carrego a planilha nessa Staging, o principal motivo é termos uma task (Data Flow Task) somente focado no tratamento da Staging.

- **Control Flow**



## • Data Flow

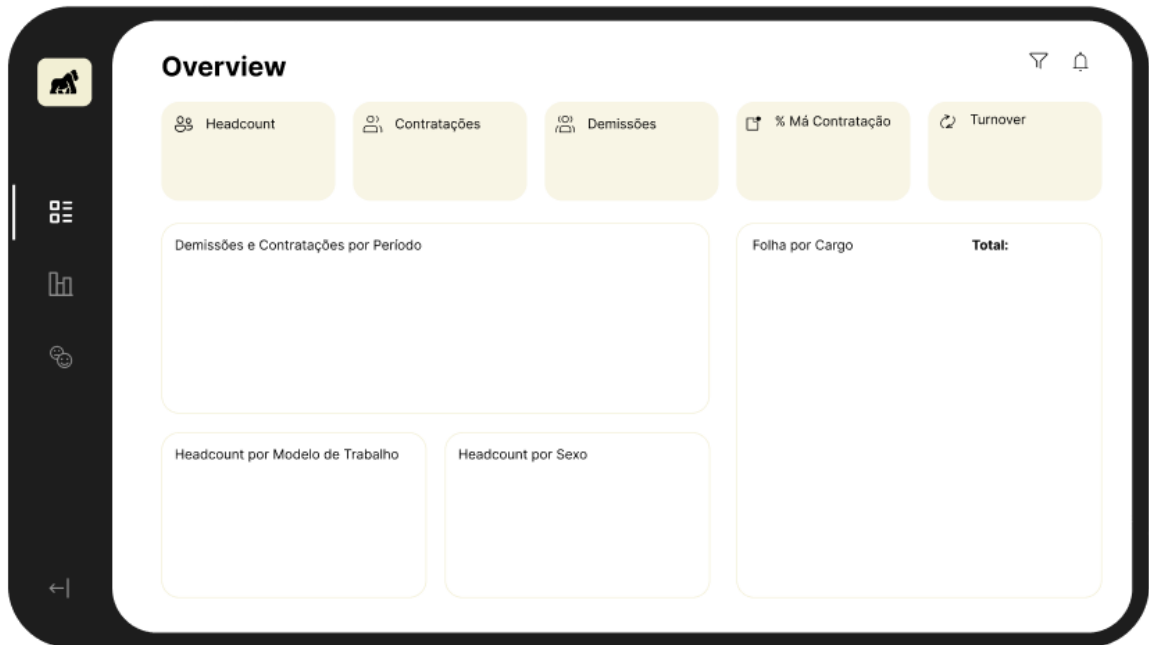


## 4. BI

### 4.1. Dashboard

### Template

O template utilizado foi criado no Figma, **tudo está disponível nos arquivos do projeto.**



## Conexão

No Power BI, vamos utilizar a conexão SQL Server Database e carregar as duas tabelas:

## SQL Server database

Server ⓘ

Database (optional)

Data Connectivity mode ⓘ

☒ Import

☐ DirectQuery

⚡ Advanced options

Command timeout in minutes (optional)

SQL statement (optional, requires database)

☒ Include relationship columns

☐ Navigate using full hierarchy

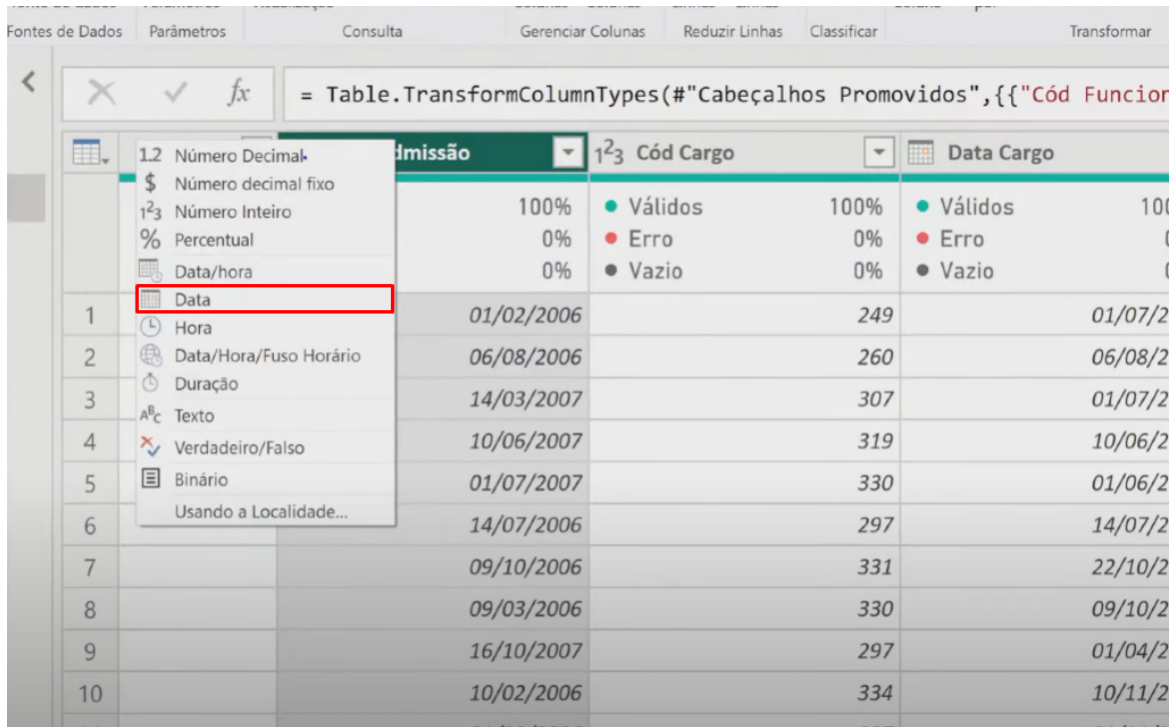
☐ Enable SQL Server Failover support

OK

Cancel

## Transformações

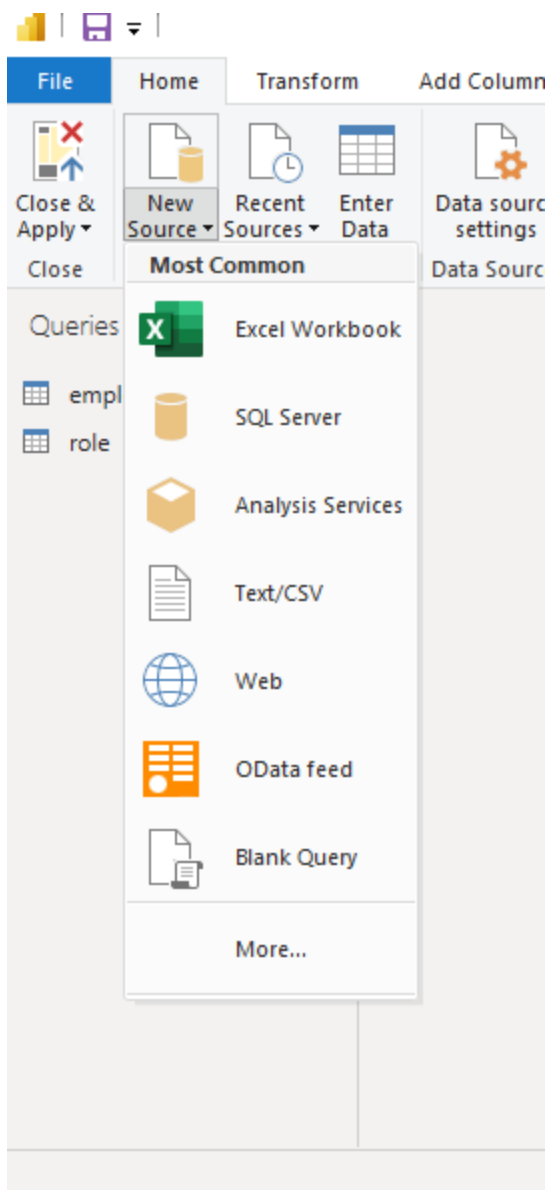
Apenas adicione o ícone "**Data**" em todas as tabelas que estão com "**Data/Hora**"



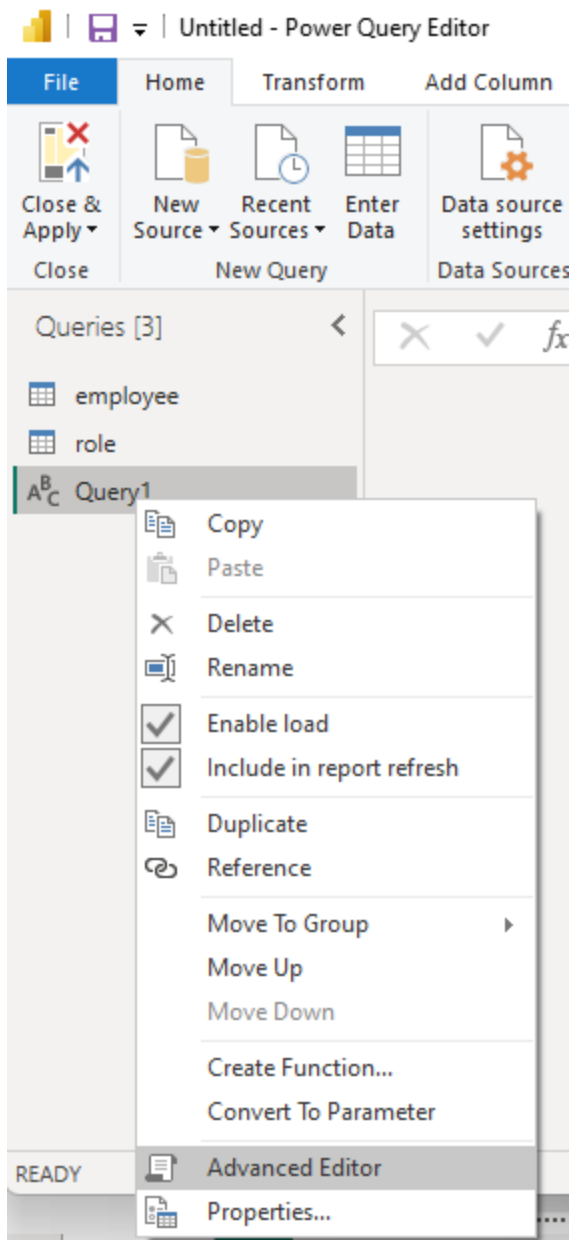
## Tabela Calendário

<https://github.com/minhasplanilhas/PowerBI/blob/master/dCalendarioCompleto>

Clique em **New Source + Blank Query**



• **Botão direito + Advanced Editor**



• Cole o código compartilhado pelo link acima e altere apenas a parte do print:

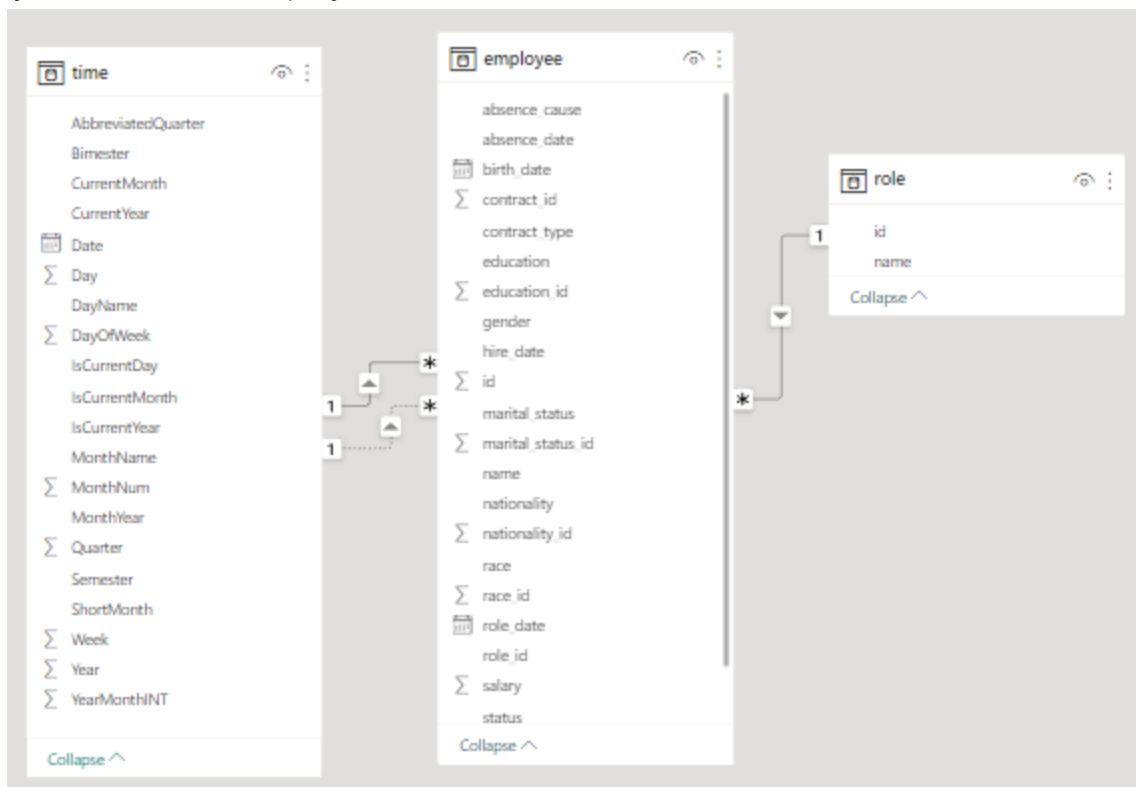


**No meu exemplo eu altero o código da tabela calendário em inglês, você encontra nos arquivos: *tableTimePowerQuery.txt***

## Relacionamentos

employee[role\_id] --> role[id]

employee[hire\_date] / employee[absence\_date] --> time[Date]

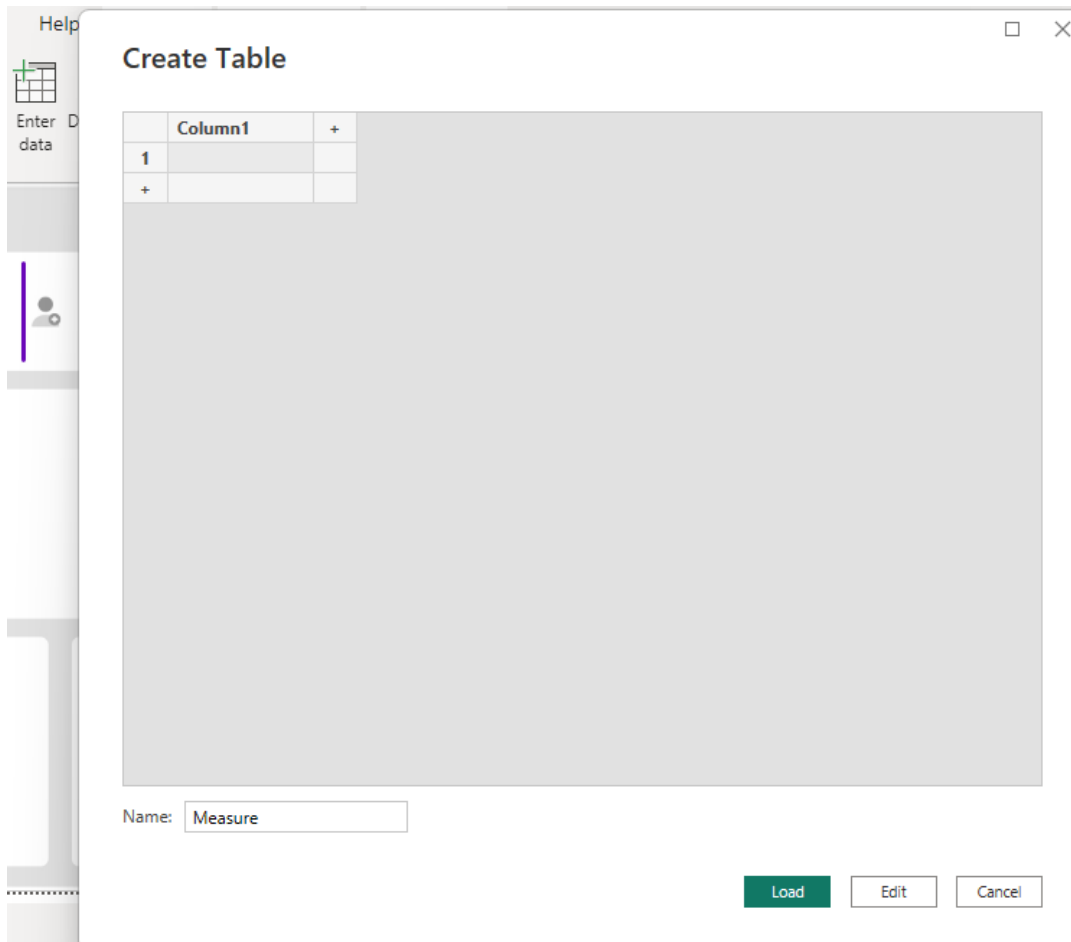


## Medidas DAX

Crie uma tabela **Measure** chamada para organizarmos as medidas DAX (Measures).

**Enter Data + Load**





## Medida 01 - Contratações

- `Total Hiring = COUNTROWS(employee)`

## Medida 02 - Demissões

- ```
layoffs =
CALCULATE(
    [Total Hiring],
    employee[status_id] = 7,
    USERELATIONSHIP('time'[Date], employee[absence_date])
)
```

## Medida 03 - Headcount (Acumulado dos funcionários)

```

Headcount =
VAR _Active =
CALCULATE(
    [Total Hiring] - [layoffs],
    FILTER(ALL('time'),
        'time'[Date] <= MAX('time'[Date])
    )
)
RETURN
    IF(_Active <> 0, _Active)

```

**Medida 04 - Tempo de Retenção** (A **retenção** de **funcionários** é uma estratégia que as empresas usam para impedir ou diminuir a rotatividade de colaboradores.). Vamos utilizar a DAX em uma nova coluna

**Obs: Vamos utilizar a DAX em uma nova coluna na tabela de funcionários**

- `retention_time = DATEDIFF(employee[hire_date], employee[absence_date], DAY)`

**Medida 05 - Total de Má Contratações** (Cada empresa analisa a sua medida como padrão, nesse modelo foi adicionado a retenção menor que 60)

```

Tot Bad Hire =
CALCULATE(
    [Total Hiring],
    employee[retention_time] < 60,
    employee[retention_time] <> BLANK()
)

```

**Medida 06 - % Má Contratações** (Apenas calculamos a porcentagem para utilizar como medida)

- `% Bad Hire = DIVIDE([Tot Bad Hire], [Total Hiring])`

**Medida 07 - Turnover** (O **turnover** é a taxa de rotatividade de funcionários, que mede o número de funcionários que saem de uma organização durante um período de tempo)

```

Turnover =
VAR _Headcount =
CALCULATE(
    [Headcount],
    PREVIOUSMONTH('time'[Date])
)
VAR _Calculation = DIVIDE([Total Hiring] + [layoffs], 2)
RETURN
    DIVIDE(_Calculation, _Headcount)

```

- 

**Medida 08 - Folha de Pagamento** (Um total de quanto é gasto com cada funcionário, desconsiderando os demitidos)

```

Payroll Value =
CALCULATE(
    SUM(employee[salary]),
    employee[status_id] <> 7)

```

- 

**Medida 09 - Demissões para o gráfico** (Um truque para deixar o valor negativo, utilizado por fins de visualização)

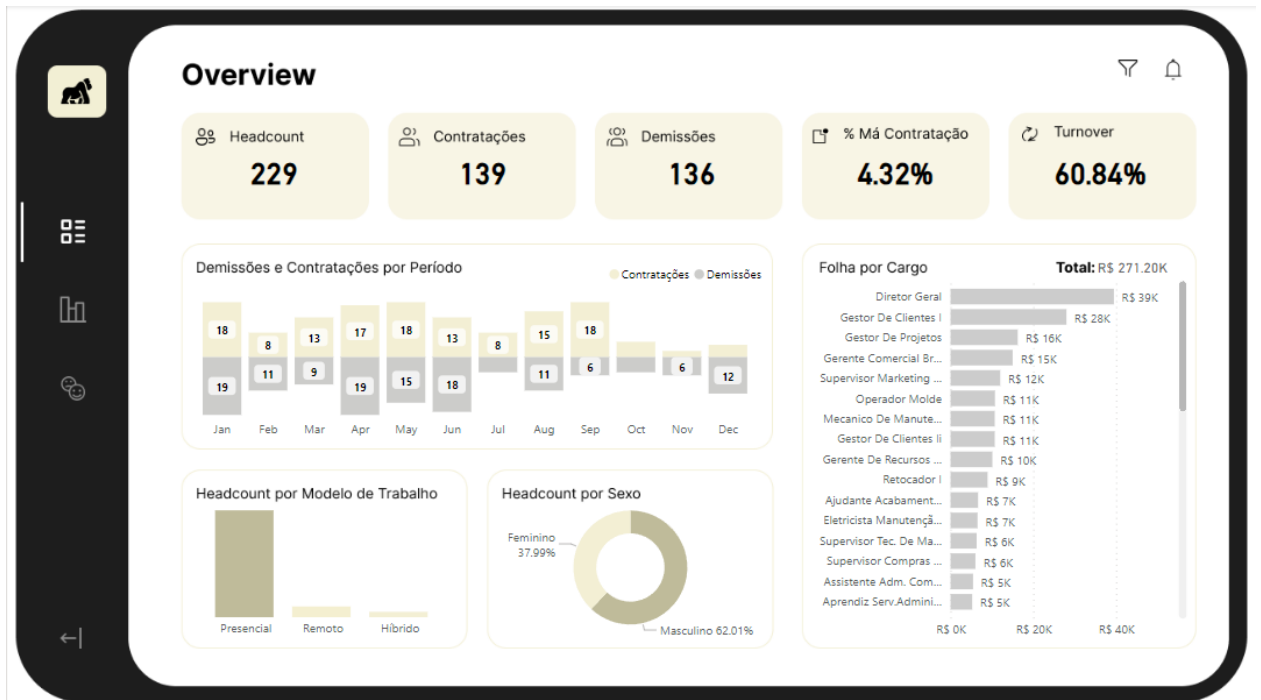
```

Layoffs Chart = - [layoffs]

```

- 

**Dashboard Final**



## 4.2. Report

### Conexão (Data Sources)

Data Source Properties

Change name, type, and connection options.

Name: people\_analytics

☒ Embedded connection:

Type: Microsoft SQL Server

Connection string: Data Source=.;Initial Catalog=people\_analytics

☐ Use shared data source reference

☐ Use single transaction when processing the queries

Build... Edit... New...

Help OK Cancel

## Conjunto de Dados (Datasets)

- Funcionário

Choose a data source and create a query.

Name:  
employee

☐ Use a shared dataset.  
☒ Use a dataset embedded in my report.

Data source:  
people\_analytics New...

Query type:  
☒ Text ☐ Table ☐ Stored Procedure

Query:

```
SELECT
    e.[name],
    e.[hire_date],
    e.[education],
    e.[work],
    e.[salary]
FROM [employee] e
WHERE [role_id] IN (@role)
AND [status_id] IN (@status)
```

fx

Query Designer... Import... Refresh Fields

Time out (in seconds):  
0

OK Cancel

- Cargo

Choose a data source and create a query.

Name:

role

- ☐ Use a shared dataset.  
☒ Use a dataset embedded in my report.

Data source:

people\_analytics

New...

Query type:

- ☒ Text ☐ Table ☐ Stored Procedure

Query:

SELECT \* FROM [role]



Query Designer...

Import...

Refresh Fields

Time out (in seconds):

0

OK

Cancel

- Status

Choose a data source and create a query.

Name:

status

- ☐ Use a shared dataset.  
☒ Use a dataset embedded in my report.

Data source:

people\_analytics

New...

Query type:

- ☒ Text ☐ Table ☐ Stored Procedure

Query:

```
SELECT DISTINCT
    [status_id],
    [status]
FROM [employee]
```



Query Designer...

Import...

Refresh Fields

Time out (in seconds):

0

OK

Cancel

## Parâmetros (Parameters)

- Cargos (Employee Role)

Report Parameter Properties



General

Available Values

Default Values

Advanced

Change name, data type, and other options.

Name:

role

Prompt:

Employee Role:

Data type:

Text

☐ Allow blank value ("")

☐ Allow null value

☒ Allow multiple values

Select parameter visibility:

☒ Visible

☐ Hidden

☐ Internal

Help

OK

Cancel



Report Parameter Properties

General  
Available Values  
Default Values  
Advanced

Choose the available values for this parameter.

Select from one of the following options:

☐ None  
☐ Specify values  
☒ Get values from a query

Dataset: (Warning: Possible performance impact)

role

Value field:

id

Label field:

name

Help OK Cancel

- Filtros dos Status (Employee Status)

Report Parameter Properties

General

Available Values

Default Values

Advanced

Change name, data type, and other options.

Name:

status

Prompt:

Employee Situation:

Data type:

Text

☐ Allow blank value ("")

☐ Allow null value

☒ Allow multiple values

Select parameter visibility:

☒ Visible

☐ Hidden

☐ Internal

Help

OK

Cancel

Report Parameter Properties

General  
**Available Values**  
 Default Values  
 Advanced

Choose the available values for this parameter.

Select from one of the following options:

☐ None  
☐ Specify values  
☒ Get values from a query

Dataset: (Warning: Possible performance impact)

status

Value field:

status\_id

Label field:

status

Help OK Cancel

## Report Final:

Design Preview

Cargo do Funcionário: Situação do Funcionário:

**FUNCIONÁRIO**

| Nome do Funcionário | Contratação | Escolaridade | Trabalho | Salário |
|---------------------|-------------|--------------|----------|---------|
| [name]              | «Expr»      | [education]  | [work]   | «Expr»  |

Relatório Gerado em: [ExecutionTime]

«Expr»

Cargo do Funcionário: Operador de Caldeira


Situação do Funcionário: Trabalhando, Demitido

1

100%

Find | Next

F U N C I O N Á R I O



| Nome do Funcionário          | Contratação | Escolaridade                | Trabalho   | Salário     |
|------------------------------|-------------|-----------------------------|------------|-------------|
| Gabriel Dias Correia         | 2018-03-09  | Ensino Médio Completo       | Presencial | R\$2,081.40 |
| Ana Carvalho Pereira Cardoso | 2019-06-16  | Ensino Médio Incompleto     | Presencial | R\$2,081.40 |
| Arthur Nobre Moraes          | 2020-06-22  | Ensino Médio Incompleto     | Presencial | R\$2,081.40 |
| Evelyn Correa                | 2021-01-16  | Ensino Fundamental Completo | Presencial | R\$2,081.40 |
| Carla Nobre Lima             | 2021-09-17  | Ensino Médio Completo       | Presencial | R\$2,081.40 |

Relatório Gerado em: 9/22/2023 4:37:18 PM

1 de 1

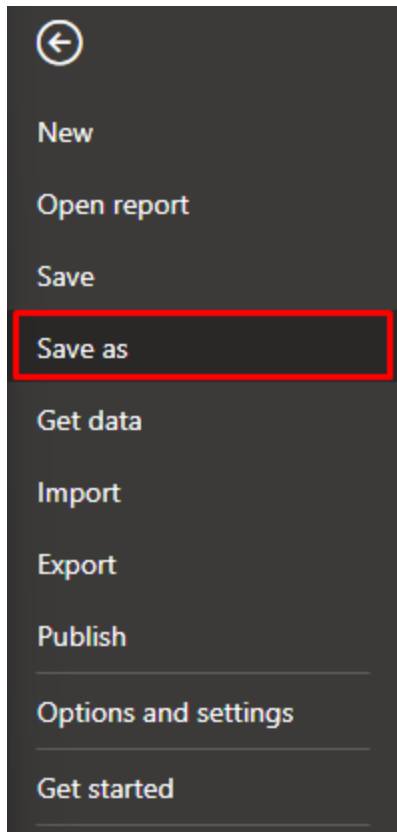
## 4.2. Deploy Reports e Dashboards

Primeiro de tudo, você precisa configurar o Power BI Report Server, você também pode implantar ambos no Power BI Service, mas nosso exemplo está focado em algo que custa 0 ou menos.

**Link:** <https://spgeeks.devoworx.com/install-and-configure-power-bi-report-server/>

### Deploy do Dashboard

**Click in File >> Save as >> Power BI Report Server**



## Save as



Browse



Power BI Report Server

Add the Report Server URL >> Ok >> Ok

## Power BI Report Server Selection



Choose the report server you would like to save your report to. You can select from the recent report server list or enter a new report server address.

Recent report servers

|                   |
|-------------------|
| localhost/Reports |
|-------------------|

New report server address (Example: <http://reportserver/reports> or <https://reportserver/reports>)

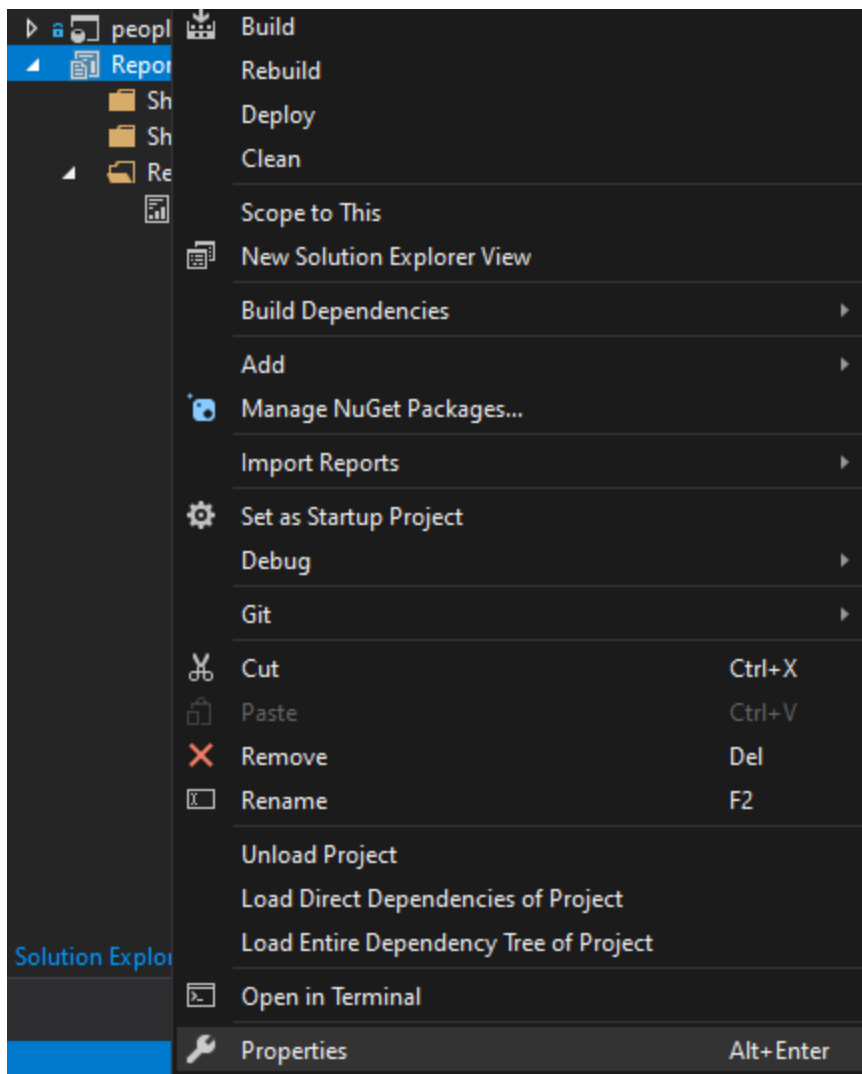
|                                                                 |
|-----------------------------------------------------------------|
| <a href="http://localhost/Reports">http://localhost/Reports</a> |
|-----------------------------------------------------------------|

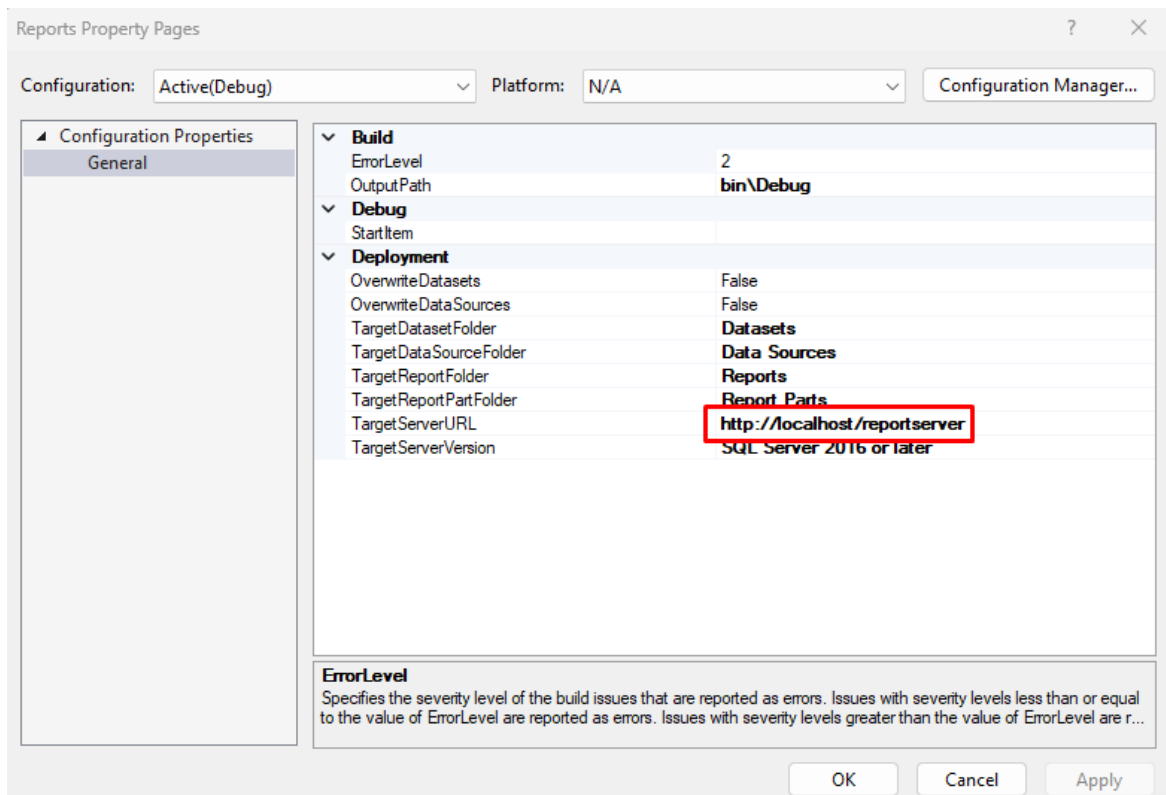
OK

Cancel

- ### Deploy do Report

Configure primeiro a URL do servidor de relatório. **Botão direito na solução >> Properties >> Target Server URL**

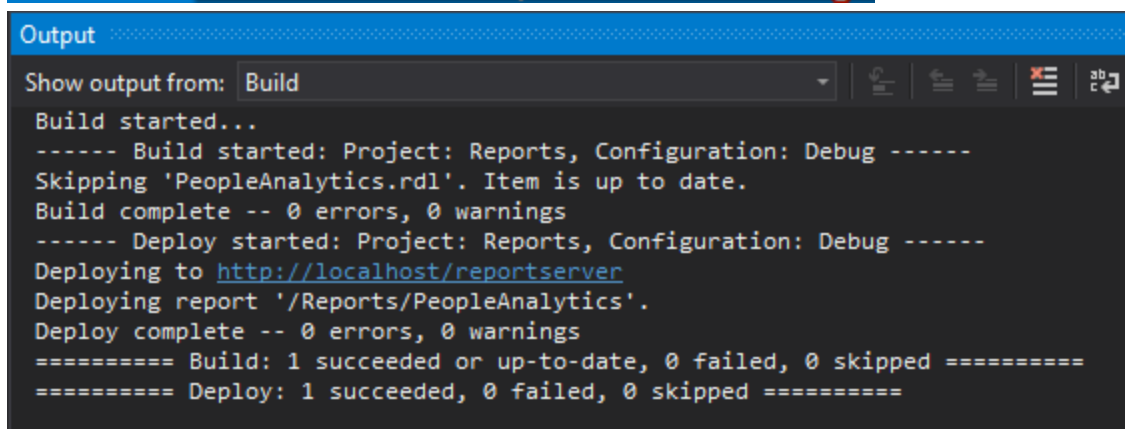
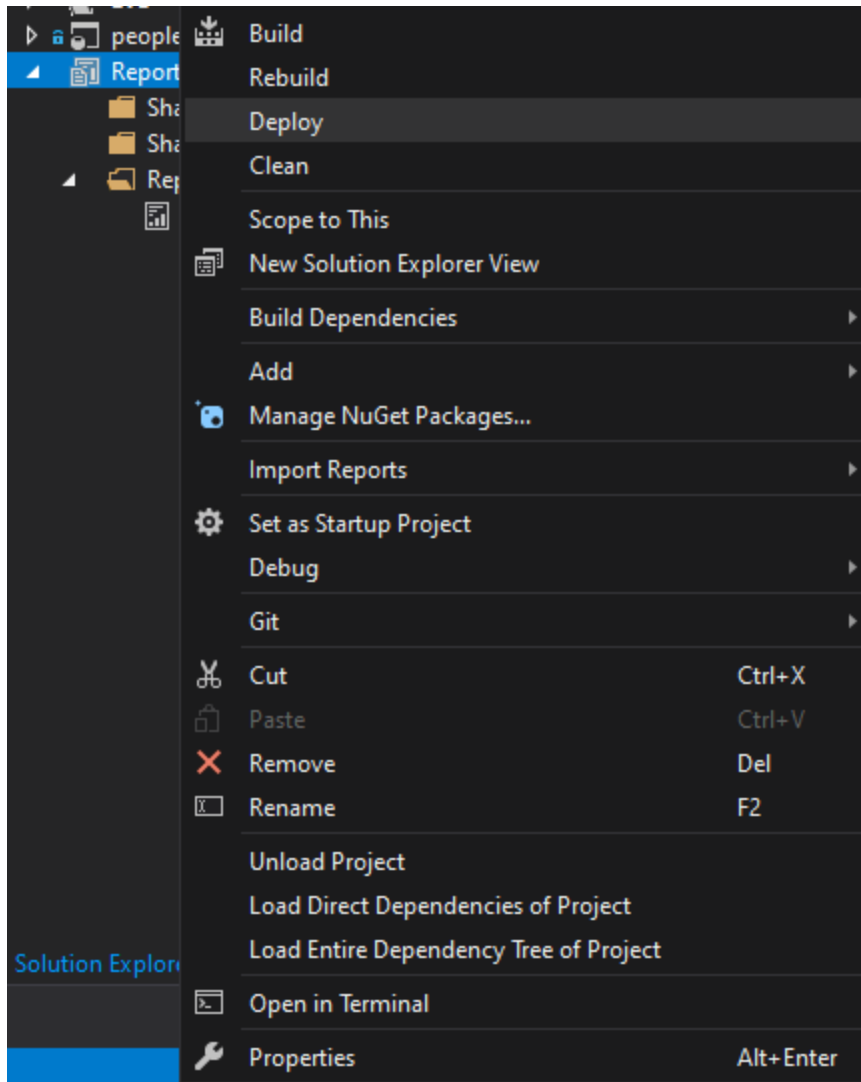




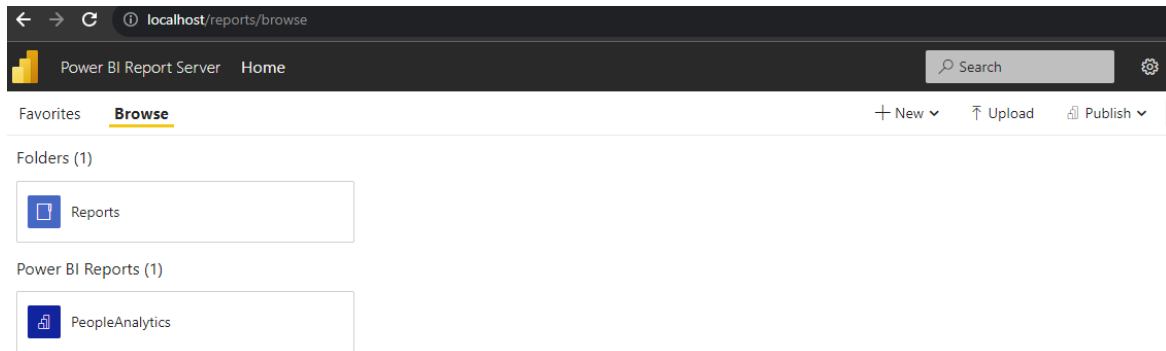
•

**Botão direito na solução >> Deploy >> Espera finalizar o Build e o Deploy**





Verifique o Report Server



## 4.4. Incorporar Reports e Dashboards

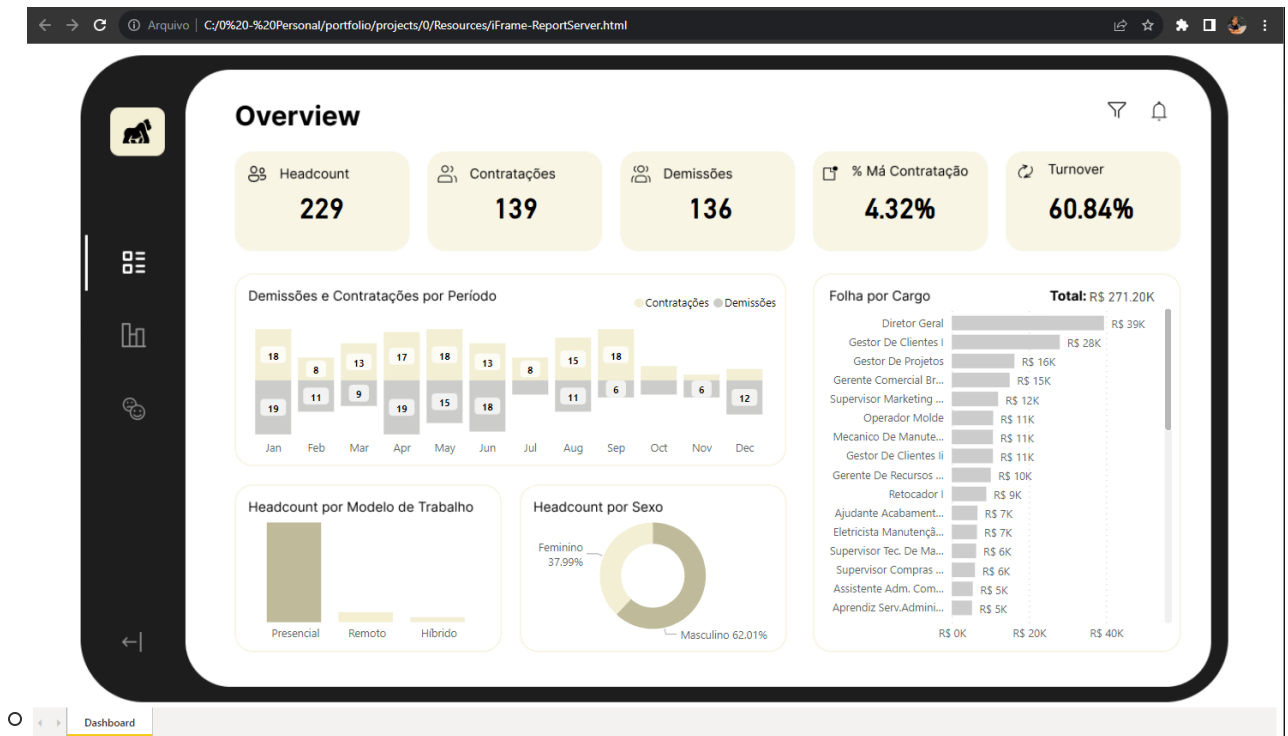
Nesta etapa, quero mostrar como podemos incorporar relatórios e painéis em nosso site ou compartilhá-los com pessoas da sua organização.

Claro que no Report Server temos menos funcionalidades e menos suporte da Microsoft, mas de qualquer forma, acredito que boa parte das empresas só usa o básico e esse básico o Report Server entrega muito bem.

Incorporar o Dashboard

- URL: <http://cylogus/Reports/powerbi/PeopleAnalytics?rs:embed=true>
- iFrame exemplo:

```
1 <html>
2 <head>
3 |   <title> "Teste PBIRS" </title>
4 </head>
5 <body>
6
7 <iframe src="http://localhost/Reports/powerbi/PeopleAnalytics?rs:embed=true" width="100%" height="800" frameborder="0"></iframe>
8
9 </body>
10 </html>
```



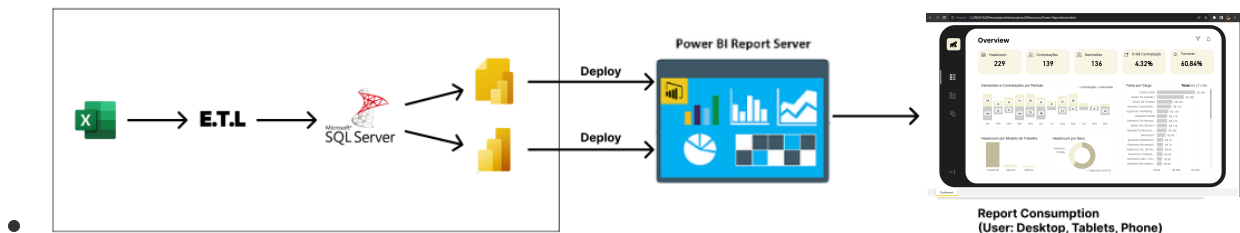
A mesma lógica é usada no Relatório

Você pode incorporar isso em um Iframe no seu site ou compartilhar a URL do servidor de relatório. Você também pode personalizar o servidor de relatório com sua marca, como um servidor para sua organização.

**Link abaixo para personalização:** <https://learn.microsoft.com/en-us/sql/reporting-services/branding-the-web-portal?view=sql-server-ver16>

## 5. Arquitetura

### 5.1. Exemplo Conceitual da Arquitetura



**Link para o Figma:** <https://www.figma.com/file/MseE0ZwKpp5sgZTgiEwPx7/Untitled?type=design&node-id=0%3A1&mode=design&t=mcfuQi1Y9VloD6JH-1>