



**INSTITUTO FEDERAL GOIANO – CAMPUS MORRINHOS – CURSO SUPERIOR
DE TECNOLOGIA EM SISTEMAS PARA INTERNET**

MARIA GABRIELA CUENCA OLIVA

RELATÓRIO TÉCNICO: GRÁFICOS COM PYTHON NO POWER BI

**MORRINHOS – GO
2023**

MARIA GABRIELA CUENCA OLIVA

RELATÓRIO TÉCNICO: GRÁFICOS COM PYTHON NO POWER BI

Relatório técnico apresentado ao Curso Superior de Tecnologia de Sistemas para Internet do Instituto Federal Goiano – Campus Morrinhos, como requisito parcial para obtenção de título de Tecnólogo em Sistemas para Internet.

Área de concentração: Ciência da computação.

Orientador: Norton Coelho Guimarães.

Sistema desenvolvido pelo ICMC/USP
Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas - Instituto Federal Goiano

Cr Cuenca Oliva, Maria Gabriela
Relatório técnico: GRÁFICOS COM PYTHON NO POWER BI
/ Maria Gabriela Cuenca Oliva; orientador Norton
Coelho Guimarães. -- Morrinhos, 2023.
39 p.

TCC (Graduação em Tecnologia Sistemas para
Internet) -- Instituto Federal Goiano, Campus
Morrinhos, 2023.

1. Python. 2. Pandas. 3. Power BI. 4. Gráficos.
I. Coelho Guimarães, Norton, orient. II. Título.

MARIA GABRIELA CUENCA OLIVA

RELATÓRIO TÉCNICO: GRÁFICOS COM PYTHON NO POWER BI

Resultado: APROVADO

Ata nº 17/2023 - CCEPTNM-MO/CEPTNM-MO/DE-MO/CMPMHOS/IFGOIANO

BANCA EXAMINADORA

**Profº Me. Norton Coelho Guimarães
IFGOIANO campus Morrinhos**

**Profª Esp. Odilon Fernandes Neto
IFGOIANO campus Morrinhos**

**Profª Me. Rodrigo Elias Francisco
IFGOIANO campus Morrinhos**

**MORRINHOS – GO
2023**

RESUMO

Os gráficos são uma maneira eficiente de visualizar dados e obter insights a partir deles. O *Power BI* é uma ferramenta poderosa para análise de dados que permite criar *dashboards* e relatórios interativos com diversos tipos de gráficos.

O *Python* é uma linguagem de programação popular para análise de dados e visualização de gráficos. O *Power BI* tem a capacidade de integrar o *Python* para criar visualizações personalizadas e avançadas.

Para usar *Python* no *Power BI*, é necessário instalar o pacote *Python* no *Power BI Desktop*. Depois de instalado, é possível criar um *script* em *Python* para gerar visualizações personalizadas. O *script* em *Python* pode ser integrado ao *Power BI* para criar uma visualização interativa.

Existem diversas bibliotecas de visualização de dados em *Python* que podem ser usadas no *Power BI*, como Matplotlib, Plotly e Bokeh. Essas bibliotecas oferecem uma ampla gama de gráficos, como gráficos de barras, gráficos de linhas, gráficos de dispersão, mapas e muito mais.

Além disso, o *Python* também oferece recursos avançados de visualização de dados, como gráficos de rede e visualizações em 3D. Esses tipos de gráficos podem ser usados no *Power BI* para criar visualizações personalizadas e avançadas.

Em resumo, o *Python* possui ferramentas poderosas para visualização de dados e pode ser integrado ao *Power BI* para criar visualizações personalizadas e avançadas. As bibliotecas de visualização de dados em *Python* oferecem uma ampla gama de gráficos que podem ser usados no *Power BI* para criar visualizações interativas e informativas.

Palavras-chaves: *Python*, *Pandas*, *Power BI*, Gráficos

ABSTRACT

Charts are an efficient way to visualize data and gain insights from it. Power BI is a powerful tool for data analysis that allows you to create interactive dashboards and reports with different types of charts.

Python is a popular programming language for analyzing data and visualizing graphs. Power BI has the ability to integrate Python to create custom and advanced visualizations.

To use Python in Power BI, you need to install the Python package in Power BI Desktop. Once installed, you can create a Python script to generate custom views. The Python script can be integrated with Power BI to create an interactive visualization.

There are several Python data visualization libraries that can be used in Power BI, such as Matplotlib, Plotly, and Bokeh. These libraries offer a wide range of charts such as bar charts, line charts, scatter charts, maps, and more.

In addition, Python also offers advanced data visualization features such as network graphs and 3D visualizations. These chart types can be used in Power BI to create custom and advanced visualizations.

In summary, Python has powerful tools for data visualization and can be integrated with Power BI to create custom and advanced visualizations. Python data visualization libraries offer a wide range of charts that can be used in Power BI to create interactive and informative visualizations.

Keywords: Python, Pandas, Power BI, Charts

AGRADECIMENTOS

Aqui, eu gostaria de expressar toda gratidão ao meu orientador Norton Guimarães, por toda paciência e por não ter desistido de mim, não só durante o processo de desenvolvimento do meu trabalho, mas também durante todo meu período de aulas na Instituição Federal Goiano - Campus Morrinhos. As suas orientações e conselhos foram fundamentais e de grande importância para que eu pudesse chegar até aqui. Novamente, obrigada por toda a paciência, dedicação e comprometimento que o senhor teve comigo.

Queria estender os meus agradecimentos ao meu esposo, Paulo Victor, por todo incentivo e paciência que teve comigo durante todo esse processo. O seu amor, compreensão e piadas foram essenciais para que eu pudesse terminar o trabalho.

Também não poderia deixar de agradecer ao meu amigo, Paulo Calanca, pela ajuda e pelos momentos de descontração que foram tão importantes para o meu equilíbrio emocional durante o desenvolvimento deste projeto.

Por fim, expresso minha gratidão a toda a minha família, principalmente meu pai, José Gregório e meus sogros, Keila e Alexandre que sempre estiveram presentes em todos os momentos, me apoiando e me incentivando a seguir em frente. Sem o amor e a compreensão de cada um de vocês, essa conquista não seria possível.

Muito obrigada a todos que contribuíram para o meu sucesso! Vocês são muito especiais em minha vida e sempre estarão em meu coração.

SUMÁRIO

RESUMO	5
ABSTRACT	6
AGRADECIMENTOS	7
SUMÁRIO	8
LISTA DE FIGURAS	9
LISTA DE CÓDIGOS	10
LISTA DE TABELAS	11
1 - INTRODUÇÃO	12
1.1 - Organização do relatório	12
2 - O POWER BUSINESS INTELLIGENCE (POWER BI)	13
2.1 - Origem do Power BI	14
2.2 - Instalação do Power BI - Step by Step	14
3 - O PYTHON	17
3.1 - Instalando o Python - step by step	17
3.2 - Introdução ao Pandas do Python	18
3.3 - Introdução ao Matplotlib do Python	19
4 - INTEGRAÇÃO DO PYTHON COM O POWER BI	21
4.1 - Por que usar o Python no Power BI?	21
4.2 - Instalando bibliotecas	21
4.3 - Utilizando o Python no Power BI	22
5 - ESTUDO DE CASO	24
5.1 - Análise Exploratória/ETL (tratamento de dados)	25
5.1.1 - Instalação das extensões - step by step	25
5.1.2 - Criando um novo arquivo - step by step	27
5.1.3 - Carregando os dados	27
5.2 - Análise Exploratória, Limpeza e Transformação de Dados	28
5.3 - Visuais no Power BI	30
6 - CONCLUSÃO	39
7 - REFERÊNCIAS BIBLIOGRÁFICAS	40

LISTA DE FIGURAS

Figura 1 - Tela de download do Power BI.	13
Figura 2 - Notificação da loja Microsoft Store.	13
Figura 3 - Tela de instalação do Power BI.	14
Figura 4 - Tela inicial do Power BI na loja da Microsoft.	14
Figura 5 - Tela do prompt de comando do Windows.	16
Figura 6 - Tela do prompt de comando do Windows.	18
Figura 7 - Tela do editor de scripts Python no Power BI.	29
Figura 8 - Tela do editor de scripts Python no Power BI com o código para gerar um gráfico de distribuição de tempo de serviço.	30
Figura 9 - Tela do campo de gráficos (Canvas) do Power BI.	31
Figura 10 - Tela do editor de scripts Python no Power BI com o código para gerar um histograma da idade dos colaboradores.	32
Figura 11 - Tela do editor de scripts Python no Power BI com o código para gerar um gráfico de violino para representar o número de treinamentos.	33
Figura 12 - Tela do editor de scripts Python no Power BI com o código para gerar um gráfico de barras para representar o número de treinamentos.	34
Figura 13 - Tela do editor de scripts Python no Power BI com o código para gerar um gráfico de barras para representar o número de colaboradores em cada nível de satisfação.	35
Figura 14 - Tela do editor de scripts Python no Power BI com o código para gerar um gráfico de barras para representar e ordenar os dados pelo salário de forma crescente.	36

LISTA DE CÓDIGOS

Trecho de código 1 - Comando de instalação da biblioteca Pandas	20
Trecho de código 2 - Comando de instalação da biblioteca Matplotlib	20
Trecho de código 3 - Importação das bibliotecas para o Python	25
Trecho de código 4 - Leitura do conjunto de dados	25
Trecho de código 5 - Visualização dos primeiros registros do conjunto de dados "dados_ilus".	26
Trecho de código 8 - Soma dos valores nulos em cada coluna	26
Trecho de código 9 - Limpeza de dados ausentes	27
Trecho de código 10 - Soma dos valores nulos em cada coluna.	27
Trecho de código 11 - Exportação do data frame para arquivo csv	28
Trecho de código 12 - Código para geração de gráfico de distribuição do tempo de serviço dos colaboradores utilizando Matplotlib e Seaborn	30
Trecho de código 13 - Código para geração de histograma da idade dos colaboradores utilizando Matplotlib e Seaborn	32
Trecho de código 14 - Código para geração de gráfico de violino utilizando Matplotlib e Seaborn para representar o número de treinamentos realizados por colaboradores.	33
Trecho de código 15 - Código para geração de gráfico de barras utilizando Matplotlib para representar a proporção de fonte de recrutamento por colaboradores	33
Trecho de código 16 - Código para geração de gráfico de barras utilizando Matplotlib para representar o número de colaboradores em cada nível de satisfação.	34
Trecho de código 17 - Código para geração de gráfico de barras utilizando Matplotlib para representar e ordenar os dados pelo salário de forma crescente	35

LISTA DE TABELAS

Tabela 1 - Dados dos colaboradores da empresa	27
Tabela 2 - Dados dos colaboradores da empresa	27

1 - INTRODUÇÃO

Os gráficos são uma forma eficiente de visualizar dados e extrair *insights* importantes deles. No entanto, para criar visualizações realmente poderosas, é necessário utilizar ferramentas avançadas de análise de dados. É aí que o *Power BI* entra em cena (MICROSOFT, 2022 b).

O *Power BI* é uma plataforma de análise de dados que permite criar relatórios e *dashboards* interativos com diversos tipos de gráficos. Além disso, o *Power BI* pode ser integrado com o *Python*, uma linguagem de programação popular para análise de dados e visualização de gráficos (CETAX, 2022).

Com a integração do *Python*, é possível criar visualizações personalizadas e avançadas no *Power BI*, utilizando bibliotecas de visualização de dados como Matplotlib, Plotly e Bokeh. Com isso, é possível criar visualizações interativas e informativas que ajudam a tomar decisões informadas com base nos *insights* extraídos dos dados (MICROSOFT, 2022 a).

Estudaremos o que é o *Power BI*, a linguagem de programação *Python* e suas bibliotecas e como foi feita a configuração do *Power BI* para utilizar o *Python*, a análise exploratória da base de dados e criação dos gráficos no *Power BI* para a Ilus Tecnologia.

1.1 - Organização do relatório

O texto do relatório está organizado da seguinte forma: No Capítulo 2 - O *Power Business Intelligence*: este capítulo conceitua o *Power BI*, sua origem e como fazer a instalação. No Capítulo 3 - O *Python*: Este capítulo conceitua a linguagem de programação e as bibliotecas que serão utilizadas e a instalação do *Python*. No Capítulo 4 - Integração do *Python* com o *Power BI*: Enumera os benefícios de utilizar o *Python* com o *Power BI*, instalação das bibliotecas Pandas e Matplotlib e a configuração do *Power BI* para utilizar o *Python*. No Capítulo 5 - Estudo de caso: Realiza a análise exploratória, configuração do *Visual Studio Code* e a criação de gráficos no *Power BI* com o *Python* e por fim no Capítulo 6 - Conclusão: Apresenta as conclusões do trabalho.

2 - O POWER BUSINESS INTELLIGENCE (POWER BI)

O *Power BI* é uma plataforma de análise de dados desenvolvida pela *Microsoft* (MONTENEGRO, 2023). Ele permite que os usuários conectem, analisem e visualizem dados de várias fontes em um único lugar (MICROSOFT, 2022). O *Power BI* é um dos líderes do mercado em termos de ferramentas de visualização de dados e é amplamente utilizado em empresas de todos os tamanhos (INFNET, 2022).

O *Power BI* é composto por vários componentes (MICROSOFT, 2022), incluindo:

1. *Power BI Desktop*: é o aplicativo de *desktop* do *Power BI* que permite criar relatórios, *dashboards* e visualizações de dados. Ele oferece uma interface de usuário intuitiva e fácil de usar para a criação de gráficos, tabelas e outras visualizações de dados.
2. *Power BI Service*: é uma plataforma baseada na web que permite compartilhar e colaborar em relatórios e *dashboards* criados no *Power BI Desktop*. Ele também oferece recursos adicionais, como programação de atualização de dados, compartilhamento de relatórios e segurança de dados.
3. *Power BI Mobile*: é um aplicativo móvel que permite acessar e interagir com relatórios e *dashboards* do *Power BI* em dispositivos móveis. Ele fornece uma experiência de usuário otimizada para dispositivos móveis e permite visualizar e compartilhar dados em movimento.

O *Power BI* suporta uma ampla variedade de fontes de dados, incluindo bancos de dados relacionais, planilhas, arquivos de texto, serviços em nuvem e muitos outros. Ele também oferece recursos de transformação e modelagem de dados que permitem limpar e preparar dados para análise (MICROSOFT, 2023).

O *Power BI* é uma ferramenta poderosa para análise de dados e visualização, permitindo que as empresas obtenham insights valiosos a partir de seus dados e tomem decisões informadas. Com uma ampla variedade de recursos e opções de personalização, o *Power BI* é uma escolha popular para análise de dados em muitas empresas em todo o mundo (MICROSOFT, 2022).

2.1 - Origem do *Power BI*

O *Power BI* foi criado pela Microsoft e lançado, em julho de 2015, como um serviço de nuvem empresarial. Antes disso, a Microsoft já tinha uma série de ferramentas de *Business Intelligence*, como o Excel e o *SQL Server Reporting Services*, mas o *Power BI* foi desenvolvido para fornecer uma plataforma mais moderna e abrangente para análise de dados (CETAX, 2022).

A ideia por trás do *Power BI* era oferecer uma ferramenta de análise de dados que fosse acessível para empresas de todos os tamanhos, sem a necessidade de uma grande equipe de TI ou recursos caros. Com o *Power BI*, os usuários podem conectar, analisar e visualizar dados de várias fontes em um único lugar, permitindo que tomem decisões informadas com base nos *insights* que extraem dos dados (MICROSOFT, 2022).

Desde o seu lançamento inicial, o *Power BI* evoluiu significativamente. A Microsoft continua a adicionar novos recursos e funcionalidades, incluindo suporte para análise avançada de dados, machine learning e visualizações personalizadas. O *Power BI* agora é uma das principais ferramentas de análise de dados disponíveis no mercado, utilizada por empresas de todos os setores para obter *insights* valiosos a partir de seus dados (DEUTSCH, 2021).

2.2 - Instalação do *Power BI* - Step by Step

Passo 1: Acesse a [página de *download*] (<https://powerbi.microsoft.com/pt-br/downloads/>) do *Power BI*.

Passo 2: Na página de *download*, você encontrará diversas opções. Procure pela opção **Microsoft Power BI Desktop** e clique em **Fazer download**:



Microsoft Power BI Desktop

Com o Power BI Desktop, você pode explorar visualmente seus dados por meio de uma tela do tipo "arrastar e soltar" de formato livre, uma ampla variedade de visualizações de dados modernas e uma experiência de criação de relatórios fácil de usar.

[Fazer download >](#)

[Opções de download avançadas >](#)

Figura 1 - Tela de download do Power BI.

Fonte: De autoria própria

Passo 3: Após essa ação, você será redirecionado para uma página em branco, onde será solicitada a abertura da loja da Microsoft. Clique em ****Abrir Microsoft Store****:

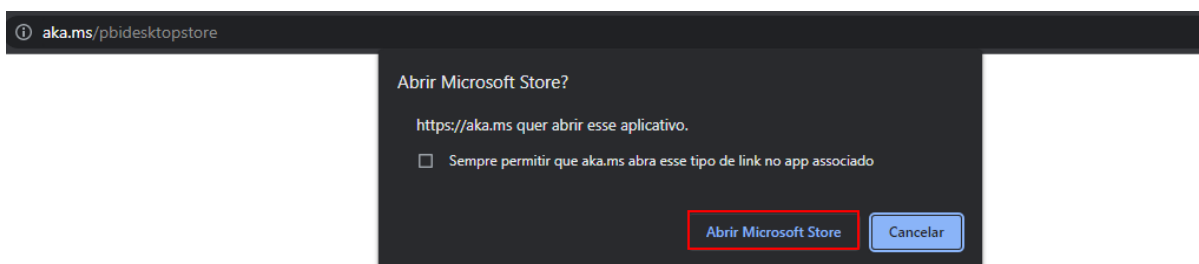


Figura 2 - Notificação da loja Microsoft Store.

Fonte: De autoria própria

Passo 4: Com a página inicial do *Power BI* Desktop na loja da Microsoft aberta, você pode clicar em ****Instalar****:

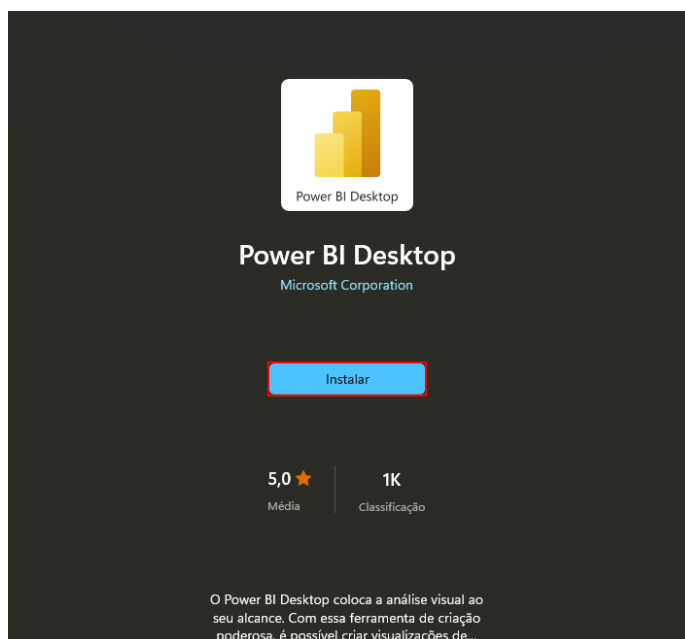


Figura 3 - Tela de instalação do Power BI.

Fonte: De autoria própria

Passo 5: Nessa etapa, é necessário aguardar a instalação ser finalizada. Após a instalação ser concluída, você pode clicar em ****Iniciar****:

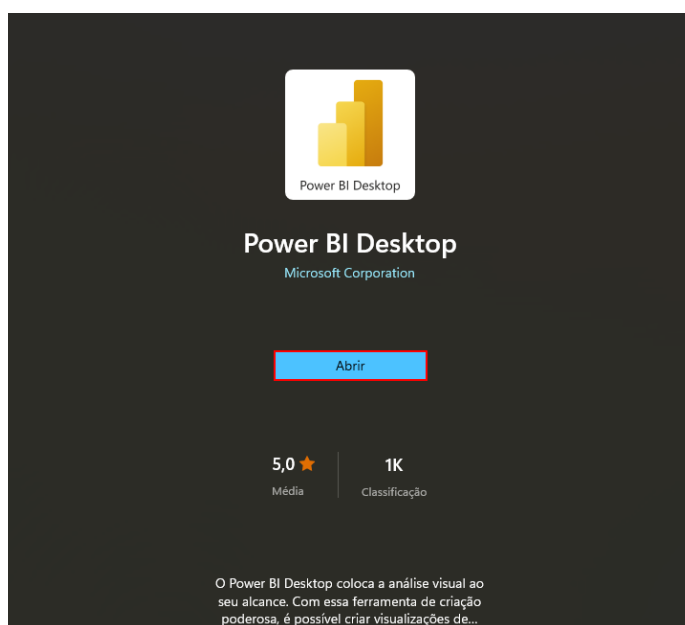


Figura 4 - Tela inicial do Power BI na loja da Microsoft.

Fonte: De autoria própria

Passo 6: Pronto! Finalizamos a instalação do *Power BI Desktop*.

3 - O PYTHON

Python é uma linguagem de programação de alto nível, interpretada, orientada a objetos e de uso geral. Foi criada em 1991 por Guido van Rossum e tem se tornado cada vez mais popular ao longo dos anos, sendo usada em diversas áreas, como ciência de dados, inteligência artificial, desenvolvimento web, automação, entre outras (VAN ROSSUM & DRAKE, 2011).

Uma das principais características do *Python* é a sua simplicidade e facilidade de aprendizado. O código em *Python* é muito legível e organizado, o que ajuda a tornar a programação mais fácil mesmo para quem está começando.

Além disso, a comunidade *Python* é muito ativa e existem diversas bibliotecas e frameworks disponíveis, o que ajuda a tornar o desenvolvimento mais rápido e eficiente (VAN ROSSUM & DRAKE, 2011).

3.1 - Instalando o *Python* - step by step

Passo 1: Acesse a página de *downloads* do site oficial do *Python*

(<https://www.python.org/downloads/release/python-3810/>)

Passo 2: Na página, verá que será utilizado a versão 3.8, é a versão mais estável, onde não trará problemas ao instalar bibliotecas que vamos precisar. Rolando a página, existem vários arquivos para *download*, vamos fazer o *download* apenas do instalador.

Passo 3: Aguarde o *download* do arquivo de instalação do *Python* (normalmente, ele será salvo na pasta de *Downloads* do seu computador).

Passo 4: Abra o arquivo de instalação do *Python* que você acabou de baixar e clique em "*Run*" para iniciar a instalação.

Passo 5: Na primeira tela do instalador, marque a opção "Add Python 3.8 to PATH" e clique em "*Customize installation*".

Passo 6: Na tela seguinte, escolha os recursos que deseja instalar (normalmente, é recomendado manter todas as opções selecionadas) e clique em "*Next*".

Passo 7: Escolha o diretório de instalação (normalmente, é recomendado manter o diretório padrão) e clique em *Install*.

Passo 8: Aguarde a instalação do *Python* ser concluída (isso pode levar alguns minutos).

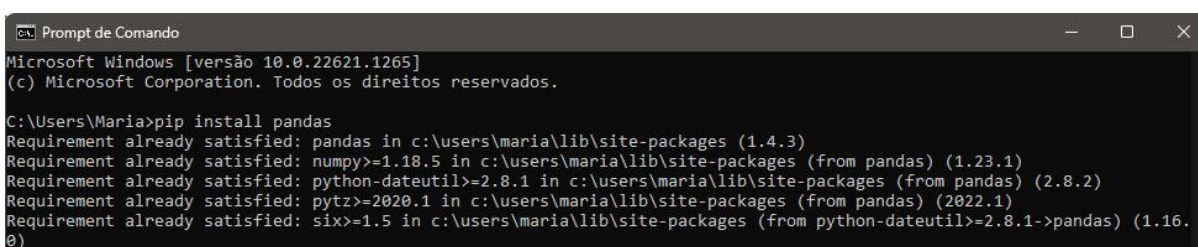
Passo 9: Após a conclusão da instalação, clique em *Close* para fechar o instalador.

Pronto! Agora o *Python* está instalado no seu computador.

3.2 - Introdução ao Pandas do *Python*

O Pandas é uma biblioteca de código aberto em *Python* para manipulação e análise de dados. Ele oferece estruturas de dados e funções de alta performance para trabalhar com dados estruturados, como tabelas e planilhas (MCKINNEY, 2017). Abaixo segue um passo a passo básico de como utilizar o Pandas em *Python*:

1. Instalação: O primeiro passo é instalar a biblioteca Pandas. Para isso, você pode utilizar o gerenciador de pacotes pip do *Python* digitando o seguinte comando no prompt de comando ou terminal:



```
Microsoft Windows [versão 10.0.22621.1265]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Maria>pip install pandas
Requirement already satisfied: pandas in c:\users\maria\lib\site-packages (1.4.3)
Requirement already satisfied: numpy>=1.18.5 in c:\users\maria\lib\site-packages (from pandas) (1.23.1)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\maria\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\maria\lib\site-packages (from pandas) (2022.1)
Requirement already satisfied: six>=1.5 in c:\users\maria\lib\site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
```

Figura 5 - Tela do prompt de comando do Windows.

Fonte: De autoria própria

2. Importação: Depois de instalado, é necessário importar o Pandas em seu script *Python* utilizando o comando *import pandas as pd*. É comum utilizar o apelido "pd" para se referir ao Pandas.
3. Leitura de dados: O Pandas permite a leitura de diversos tipos de arquivos, como CSV, Excel, SQL, JSON, entre outros. Para ler um arquivo CSV, por

exemplo, você pode utilizar a função `pd.read_csv("nome_do_arquivo.csv")` e atribuir o resultado a uma variável.

4. Manipulação de dados: O Pandas oferece diversas funções para manipular dados, como filtros, agregações, ordenações, preenchimento de valores faltantes, entre outros. Algumas das funções mais comuns são `df.head()` para visualizar as primeiras linhas de um *DataFrame*, `df.describe()` para gerar estatísticas descritivas de um *DataFrame* e `df.groupby()` para agrupar os dados por uma ou mais colunas.
5. Visualização de dados: O Pandas também oferece recursos para visualização de dados, como gráficos de barras, de linhas, de dispersão, entre outros. Para gerar um gráfico de barras, por exemplo, você pode utilizar o comando `df.plot(kind='bar', x='nome_da_coluna_x', y='nome_da_coluna_y')`.

Esses são alguns passos básicos para utilizar o Pandas em *Python*. O Pandas oferece muitas outras funções e recursos avançados para trabalhar com dados, mas esses passos já são suficientes para começar a manipular e analisar dados de forma eficiente em *Python* (MCKINNEY, 2017).

3.3 - Introdução ao Matplotlib do *Python*

Matplotlib é uma biblioteca de visualização de dados em *Python* que permite criar gráficos, diagramas, plots, mapas e outras visualizações. Foi criada por John D. Hunter em 2003 e é mantida atualmente pela comunidade (HUNTER, 2007).

Matplotlib é uma das bibliotecas mais utilizadas para visualização de dados em *Python*, por sua versatilidade e facilidade de uso. Ele permite criar gráficos de linha, de barras, de dispersão, histogramas, gráficos em 3D e muito mais (ALURA, 2018).

Para começar a usar o Matplotlib, é necessário instalar a biblioteca e importá-la no código *Python*. A maneira mais comum de instalar o Matplotlib é através do gerenciador de pacotes `pip`.

```
C:\Users\Maria>pip install matplotlib
Requirement already satisfied: matplotlib in c:\users\maria\lib\site-packages (3.5.2)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\maria\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\maria\lib\site-packages (from matplotlib) (4.34.4)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\maria\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: packaging>=20.0 in c:\users\maria\lib\site-packages (from matplotlib) (21.3)
Requirement already satisfied: pillow>=6.2.0 in c:\users\maria\lib\site-packages (from matplotlib) (9.2.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\maria\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: cycler>=0.10 in c:\users\maria\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: numpy>=1.17 in c:\users\maria\lib\site-packages (from matplotlib) (1.23.1)
Requirement already satisfied: six>=1.5 in c:\users\maria\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
C:\Users\Maria>
```

Figura 6 - Tela do prompt de comando do Windows.

Fonte: De autoria própria

Existem muitos outros recursos disponíveis no Matplotlib, como a personalização de cores, rótulos de eixo, legendas, tipos de linha e muito mais. Com prática e experimentação, você pode criar visualizações poderosas e informativas de seus dados com o Matplotlib.

4 - INTEGRAÇÃO DO PYTHON COM O POWER BI

4.1 - Por que usar o Python no Power BI?

Existem algumas razões pelas quais você pode querer usar o *Python* no *Power BI* (REZNIK, 2021):

1. Análise avançada de dados: O *Python* é uma linguagem de programação poderosa para análise de dados e pode ser usado para executar tarefas mais avançadas do que o *Power BI* sozinho pode realizar. Por exemplo, você pode usar o *Python* para criar modelos de machine learning e realizar análises estatísticas complexas.
2. Integração com bibliotecas externas: O *Python* tem uma grande variedade de bibliotecas externas disponíveis que podem ser usadas para executar tarefas específicas. Com o *Python* no *Power BI*, você pode acessar essas bibliotecas diretamente, permitindo que você execute tarefas personalizadas que não seriam possíveis com o *Power BI* sozinho.
3. Visualização personalizada: O *Python* também pode ser usado para criar visualizações personalizadas no *Power BI*. Isso significa que você pode criar gráficos e visualizações personalizados que não estão disponíveis no *Power BI* padrão.
4. Automação de tarefas: O *Python* pode ser usado para automatizar tarefas no *Power BI*, o que pode economizar tempo e reduzir erros manuais. Por exemplo, você pode usar o *Python* para automatizar o processo de atualização de dados ou para executar uma série de análises em um conjunto de dados.

Em resumo, usar o *Python* no *Power BI* pode adicionar uma camada adicional de funcionalidade e personalização, permitindo que você execute tarefas mais avançadas e obtenha insights mais profundos a partir dos seus dados (REZNIK, 2021).

4.2 - Instalando bibliotecas

A integração do *Power BI* com *Python* requer a instalação de duas bibliotecas. No CMD (*prompt* de comando) ou shell, use a ferramenta de linha de

comando pip para instalar os pacotes. A ferramenta pip é empacotada com versões recentes do *Python*.

O **Pandas** é uma biblioteca de software para manipulação e análise de dados. Oferece estruturas de dados e operações para manipular tabelas numéricas e séries temporais (MCKINNEY, 2010).

Matplotlib é uma biblioteca de plotagem para *Python* e sua extensão matemática numérica NumPy. O Matplotlib fornece uma API orientada a objeto para inserir gráficos em aplicativos de GUI (interface gráfica do usuário) de uso geral para *Python* (HUNTER, 2007).

Então, na barra de pesquisa do Windows, vamos digitar **cmd** para abrir o prompt de comando e ali vamos usar os seguintes comandos:

```
pip install pandas
```

Trecho de código 1 - Comando de instalação da biblioteca Pandas

Fonte: Python

Depois de instalado o Pandas, vamos instalar o Matplotlib:

```
pip install matplotlib
```

Trecho de código 2 - Comando de instalação da biblioteca Matplotlib

Fonte: Python

4.3 - Utilizando o *Python* no *Power BI*

Para utilizar o *Python* no *Power BI*, é necessário seguir os seguintes passos:

1. No *Power BI* Desktop, selecione Arquivo>Opções e Configurações>Opções>Script do *Python*. A página Opções de scripts do *Python* é exibida.
2. Se necessário, forneça ou edite o caminho de instalação local do *Python* em Diretórios base do *Python* detectados. Na imagem acima, o caminho local de

instalação do *Python* é C:\Python. Se você tem mais de uma instalação local do *Python*, selecione a que deseja usar.

3. Criação de um script em *Python*: Depois de instalar o pacote *Python*, é possível criar um script em *Python* para gerar visualizações personalizadas. Por exemplo, é possível criar um gráfico de dispersão com uma linha de tendência usando a biblioteca Seaborn em *Python*. O script em *Python* pode ser integrado ao *Power BI* para criar uma visualização interativa.
4. Integração do script em *Python* ao *Power BI*: Para integrar o script em *Python* ao *Power BI*, é necessário criar uma visualização personalizada. Para criar uma visualização personalizada, selecione o tipo de visualização que deseja criar e selecione a opção "Personalizado" no painel "Visualizações". Em seguida, selecione o botão "Importar de arquivo" e selecione o arquivo com o script em *Python*.
5. Execução do script em *Python*: Depois de integrar o script em *Python* ao *Power BI*, é possível executá-lo clicando no botão "Atualizar" no painel "Visualizações". O script em *Python* será executado e a visualização personalizada será gerada

5 - ESTUDO DE CASO

A Illus Tecnologia é uma empresa que fornece soluções de serviços de computação em nuvem. Atualmente, a empresa tem passado por algumas mudanças de colaboradores que são difíceis de acompanhar. O objetivo é estudar a relação desses atributos e o impacto na promoção dos colaboradores. Esse trabalho de análise pode ser usado mais tarde para construir modelos de *Machine Learning* para prever se um colaborador será ou não promovido.

Essas são as perguntas que devem ser respondidas:

- Pergunta 1 - Qual o tempo de serviço da maioria dos colaboradores?
- Pergunta 2 - Qual a distribuição das idades dos colaboradores?
- Pergunta 3 - Qual o número de treinamentos mais frequente?
- Pergunta 4 - Qual a proporção dos colaboradores por fonte de recrutamento?
- Pergunta 5 - Quantos colaboradores têm o maior número de avaliações?
- Pergunta 6 - Qual o maior salário por departamento?

A base de dados consiste em dados sobre os colaboradores da empresa:

- Id do colaborador
- Salário
- Posição
- Estado
- Educação
- Idade
- Gênero
- Fonte Recrutamento
- Data Contratação
- Departamento
- Nível Satisfação
- Número de treinamentos
- Promovido

Usaremos o *Visual Studio Code* juntamente com a extensão do *Jupyter* para realizar o tratamento dos dados, se necessários. As bibliotecas *Pandas* e *Matplotlib* serão utilizadas para a criação dos gráficos para utilizarmos no *Power BI*.

Durante o desenvolvimento, foram feitas cinco simulações para chegar ao resultado final. Nessas simulações, foram feitos testes com as bibliotecas, como resolver o desbalanceamento de dados e testes com o melhor tipo de gráfico a ser utilizado para mostrar as informações.

5.1 - Análise Exploratória/ETL (tratamento de dados)

Aqui nessa parte, foi utilizado a extensão do *Jupyter Notebook* no *VS Code* para realizar a análise exploratória.

5.1.1 - Instalação das extensões - *step by step*

A extensão *Jupyter Notebook* para o *Visual Studio Code* permite que você trabalhe com *notebooks Jupyter* em seu ambiente de desenvolvimento integrado.

Neste tutorial, irei guiar você através do processo de instalação da extensão *Jupyter Notebook* no *Visual Studio Code*.

Antes de começar, certifique-se de que você tenha instalado o *Visual Studio Code* em seu computador. Caso ainda não tenha, você pode fazer o *download* do *Visual Studio Code* em seu site oficial.

Passo 1: Abra o *Visual Studio Code*

Abra o *Visual Studio Code* em seu computador. Se você estiver usando o *Windows*, pode encontrar o *Visual Studio Code* no menu *Iniciar* ou na barra de tarefas. Se você estiver usando um sistema operacional diferente, encontre o *Visual Studio Code* em seu menu de aplicativos.

Passo 2: Abra o gerenciador de extensões

No *Visual Studio Code*, abra o gerenciador de extensões clicando no ícone de extensões no menu lateral ou usando o atalho "Ctrl + Shift + X" no Windows ou "Cmd + Shift + X" no macOS.

Passo 3: Pesquise a extensão *Jupyter Notebook*

No campo de pesquisa do gerenciador de extensões, digite "Jupyter Notebook" e pressione Enter. A extensão *Jupyter Notebook* aparecerá nos resultados da pesquisa.

Passo 4: Instale a extensão *Jupyter Notebook*

Clique no botão "*Install*" para instalar a extensão *Jupyter Notebook*. Após alguns segundos, a extensão será baixada e instalada em seu *Visual Studio Code*.

Passo 5: Reinicie o *Visual Studio Code*

Após a instalação da extensão *Jupyter Notebook*, você precisará reiniciar o *Visual Studio Code* para que as alterações entrem em vigor. Clique em "Reload" na notificação que aparece na parte inferior da tela para reiniciar o *Visual Studio Code*.

Além disso, precisamos da extensão da linguagem *Python* para podermos compilar os códigos. O *Visual Studio Code* é um editor de código e não um compilador.

A instalação da extensão do *Python* no *Visual Studio Code* é um processo simples que pode ser feito em poucos passos. Siga as instruções abaixo para instalar a extensão do *Python* no *Visual Studio Code*:

Passo 1: Abra o *Visual Studio Code*

Passo 2: Abra o gerenciador de extensões

Passo 3: Abra o Gerenciador de Extensões. Procure pela Extensão do *Python* No Gerenciador de Extensões, digite "*Python*" na barra de pesquisa. A extensão do *Python* aparecerá nos resultados da pesquisa. Clique em "Instalar" para começar a instalação.

Passo 4: Aguarde a Instalação Após clicar em "Instalar", aguarde até que a extensão seja baixada e instalada em seu computador.

Passo 5: Reinicie o *Visual Studio Code* Após a instalação, o *Visual Studio Code* irá solicitar que você reinicie a aplicação para ativar a extensão do *Python*. Clique em "Reiniciar" para permitir que o *Visual Studio Code* seja reiniciado.

5.1.2 - Criando um novo arquivo - *step by step*

Passo 1: Com o *Visual Studio Code* aberto, clique em "Arquivo" no menu superior.

Passo 2: Na barra superior, digite o nome do arquivo que você deseja criar, incluindo a extensão de arquivo apropriada. Em seguida, selecione a linguagem de programação desejada para o arquivo. Se você não selecionar uma linguagem de programação específica, o *Visual Studio Code* tentará adivinhar a linguagem do arquivo com base na extensão do arquivo.

A partir daqui começa a descrição de como foi feita a análise exploratória.

Antes de tudo, precisamos importar as bibliotecas que vamos utilizar:

```
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
```

Trecho de código 3 - Importação das bibliotecas para o Python

Fonte: De autoria própria

5.1.3 - Carregando os dados

Foi utilizada a biblioteca Pandas para realizar a leitura dos dados do arquivo 'dp_ilus.csv', por meio do seguinte código em *Python*: `dados_ilus = pd.read_csv('dp_ilus.csv')`

```
dados_ilus = pd.read_csv('dp_ilus.csv')
```

Trecho de código 4 - Leitura do conjunto de dados

Fonte: De autoria própria

Caso queira ver o conteúdo do csv, pode usar essa função:

```
dados_ilus.head()
```

Trecho de código 5 - Visualização dos primeiros registros do conjunto de dados "dados_ilus".

Fonte: De autoria própria

5.2 - Análise Exploratória, Limpeza e Transformação de Dados

Provavelmente pode ser que tenha dados que estão nulos/vazios ou até mesmo dados desbalanceados. E aqui, vamos ajustar tudo isso.

Para verificar a quantidade de valores nulos em cada coluna do conjunto de dados "dados_ilus", foi utilizado o seguinte código em *Python*:

```
dados_ilus.isnull().sum()
```

Trecho de código 8 - Soma dos valores nulos em cada coluna

Fonte: De autoria própria

`isnull()` é uma função que mostra o que está nulo e o `sum()` faz essa somatória dos registros nulos.

O resultado dessa operação indica a quantidade de valores nulos em cada coluna do conjunto de dados.

Quadro 1 - Dados dos colaboradores da empresa

IdColaborador	12
Salario	12
Posicao	12
Estado	12
Educacao	12
Idade	12
Genero	12
FonteRecrutamento	12
DataContratacao	12
tempoServico	0
Departamento	12
NivelSatisfacao	12

```
numero_treinamentos    12
Promovido              12
```

A tabela acima apresenta informações sobre salário, posição, estado, educação, idade, gênero, fonte de recrutamento, data de contratação, tempo de serviço, departamento, nível de satisfação, número de treinamentos e se foram promovidos ou não. O campo "tempoServico" não apresenta dados nulos.

Vamos usar a função `dropna()` para deletar todas as linhas que estão com valores nulos.

```
dadosilus = dados_ilus.dropna()
```

Trecho de código 9 - Limpeza de dados ausentes

Fonte: De autoria própria

Se rodarmos o código de novo, não teremos mais registros nulos.

```
dadosilus.isnull().sum()
```

Trecho de código 10 - Soma dos valores nulos em cada coluna.

Fonte: De autoria própria

Quadro 2 - Dados dos colaboradores da empresa

Fonte: De autoria própria

```
IdColaborador    0
Salario          0
Posicao           0
Estado           0
Educacao         0
Idade            0
Genero           0
FonteRecrutamento 0
DataContratacao  0
tempoServico     0
Departamento    0
NivelSatisfacao  0
```

```
numero_treinamentos    0
Promovido                0
```

A tabela apresenta informações sobre os colaboradores, incluindo seu ID, salário, posição, estado, educação, idade, gênero, fonte de recrutamento, data de contratação, tempo de serviço, departamento, nível de satisfação, número de treinamentos realizados e se foram ou não promovidos. As informações da tabela não apresentam valores ausentes (0 em todas as colunas).

Agora, vamos salvar um novo *dataset*, que agora está com os dados tratados. A partir daí, podemos iniciar no *Power BI*.

```
dadosilus.to_csv('C:/Users/maria/Área de
Trabalho/tcc/dados_ilus_tratados.csv', encoding = 'utf-8', index =
False)
```

Trecho de código 11 - Exportação do data frame para arquivo csv

Fonte: De autoria própria

Código criado para tratamento de dados e exportação em formato CSV para o arquivo "dados_ilus_tratados.csv". A codificação utilizada foi UTF-8 e o índice não foi incluído.

5.3 - Visuais no *Power BI*

Aqui, já temos o *Power BI* configurado para podermos utilizar o *Python*. Com isso, podemos começar a inserir os visuais.

No campo de visuais, vamos adicionar um visual de *Python*, que no caso, é o visual que tem o símbolo 'Py'.

Ao adicionar, uma mensagem dizendo que é necessário habilitar os visuais *python* é mostrada, então clicamos no botão que diz 'habilitar visuais'. Tudo pronto, podemos começar a digitar nossos códigos para criar os visuais.

Um campo é aberto para poder digitar os códigos, mas ainda não carregamos a base de dados. Na parte superior, tem um botão de 'obter dados' e clicamos na opção 'Texto/CSV'.

Vamos carregar o *dataset* que está tratado, no caso, 'dados_ilus_tratados.csv'. Ao carregar, não vai ser preciso transformar os dados, já que isso foi feito no *Python*, apenas adicionamos os dados e ao fazer isso, ao lado direito a lista com os campos do *dataset* será mostrada.

Feito isso, podemos começar a responder as perguntas com os gráficos:

- Pergunta 1: Qual o tempo de serviço da maioria dos colaboradores?

Antes de começarmos a digitar os códigos, precisamos adicionar o campo que vamos utilizar dentro do visual no *Power BI*.

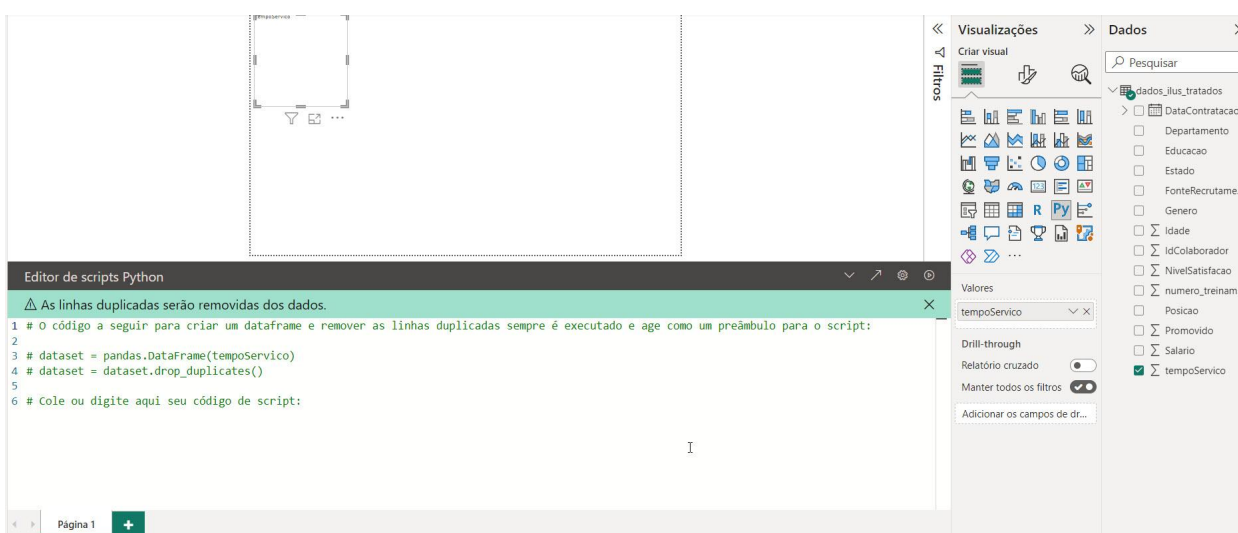


Figura 7 - Tela do editor de scripts Python no Power BI.

Fonte: De autoria própria

O que acontece quando adicionamos um campo dentro do visual de *Python* no *Power BI*? O Pandas, ele cria um novo *data frame* a partir do campo inserido no visual, ele cria um índice novo, no entanto, o índice do Pandas é criado a partir do 0 e não do 1. Isso pode trazer problemas para o resultado final do gráfico.

Como resolvemos isso? Usando um campo que pode servir como um índice, no caso, o campo 'id_colaborador'. Então, adicionar esse campo junto com o 'tempoServico' para o gráfico vir com o resultado correto.

Agora vou adicionar o código e executar:

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
##grafico distribuição, tempo
sns.displot(dataset['tempoServico'], color = 'blue')
plt.title('Tempo de serviço dos colaboradores', fontsize = 15)
plt.xlabel('Tempo de serviço em anos', fontsize = 15)
plt.ylabel('Total')
plt.show()
```

Trecho de código 12 - Código para geração de gráfico de distribuição do tempo de serviço dos colaboradores utilizando Matplotlib e Seaborn

Fonte: De autoria própria

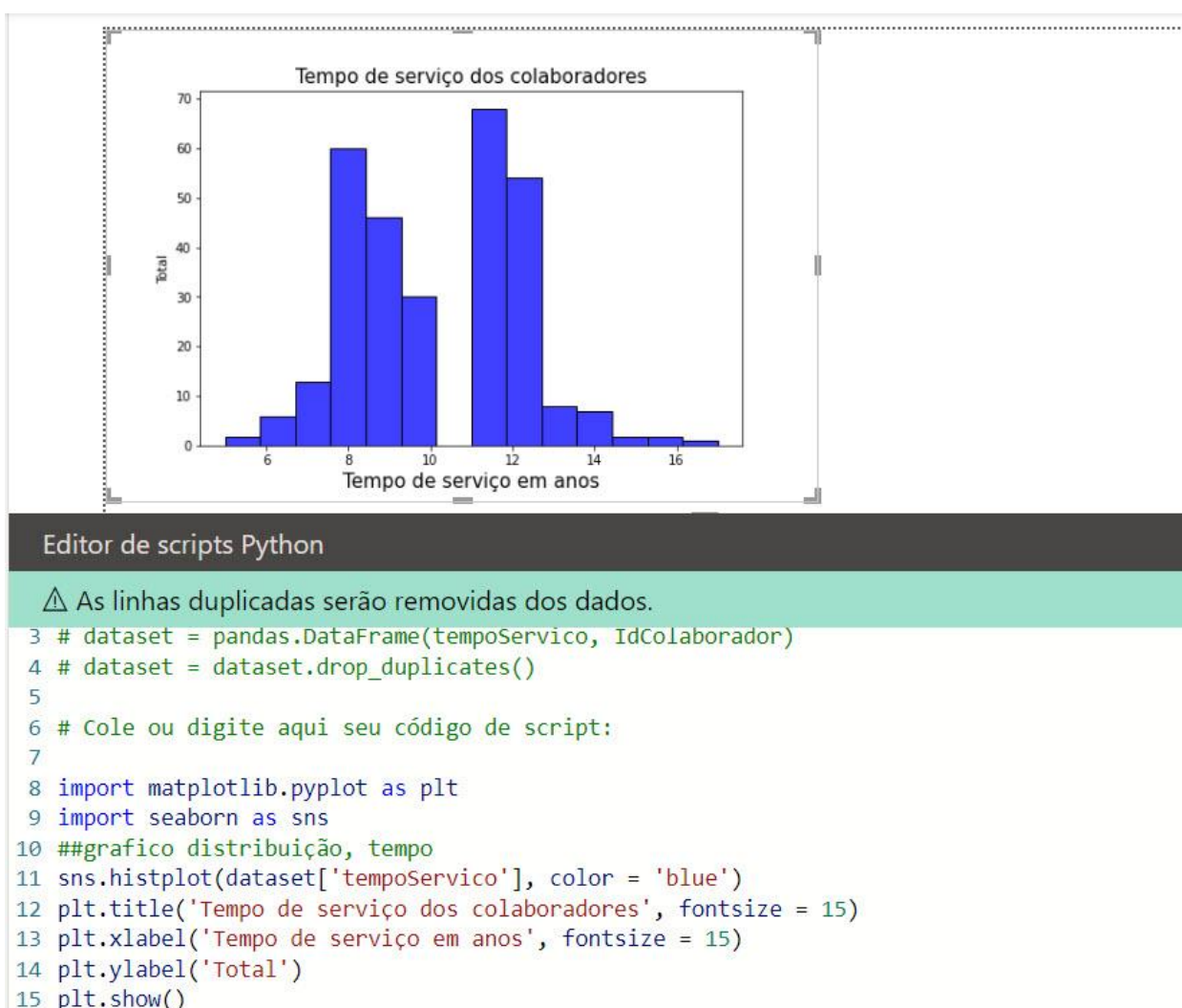


Figura 8 - Tela do editor de scripts Python no Power BI com o código para gerar um gráfico de distribuição de tempo de serviço.

Fonte: De autoria própria

De início, o gráfico está pequeno. Para resolver, é necessário recolher o campo de editor de scripts *Python*.

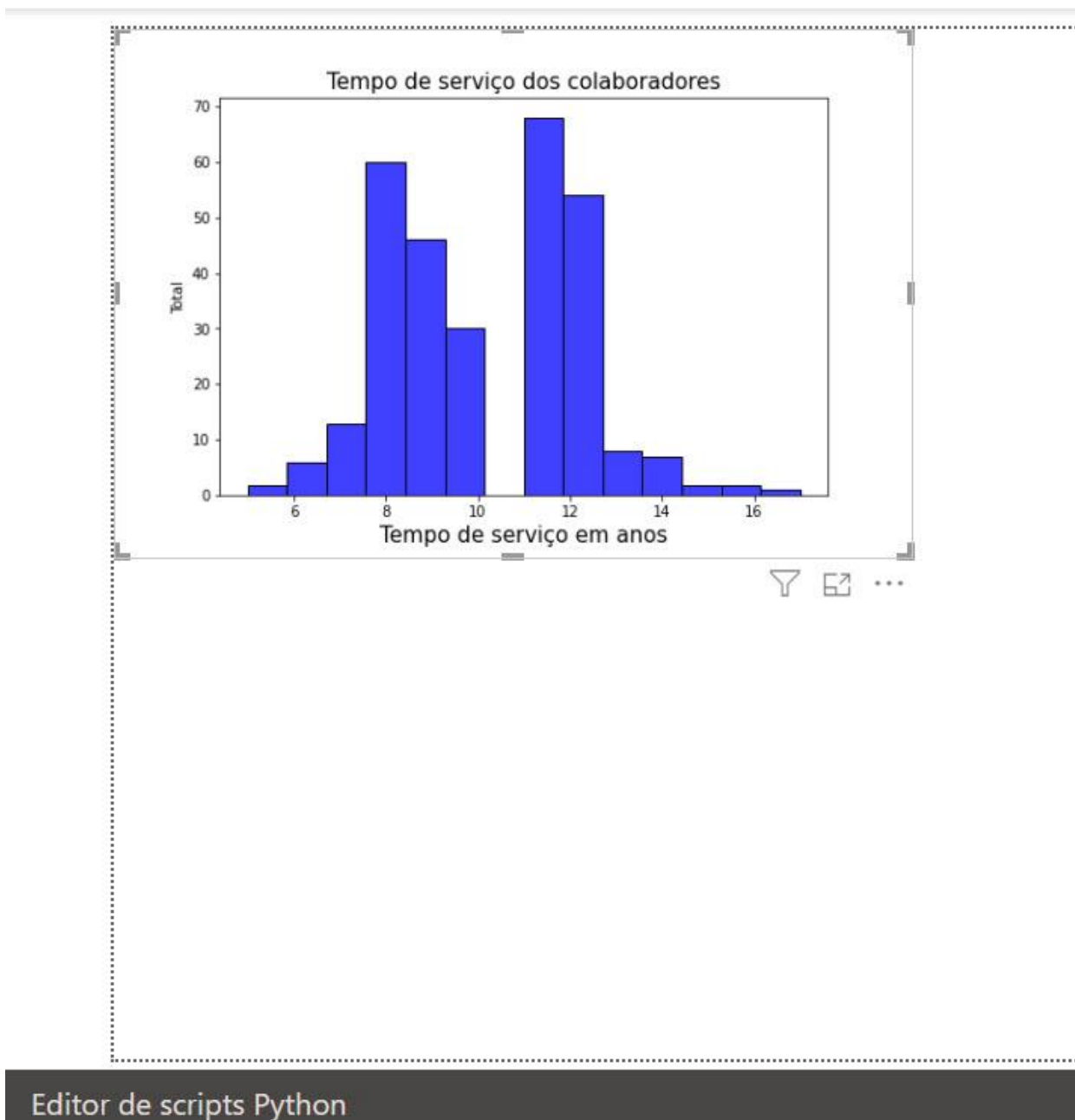


Figura 9 - Tela do campo de gráficos (Canvas) do Power BI.

Fonte: De autoria própria

A partir daqui, podemos responder o restante das perguntas.

- Pergunta 2: Qual a distribuição das idades dos colaboradores?

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.histplot(dataset['Idade'], color = 'cornflowerblue')
plt.title('Idade dos Colaboradores', fontsize = 15)
plt.xlabel('Idade', fontsize = 15)
plt.ylabel('Total')
plt.show()
```

Trecho de código 13 - Código para geração de histograma da idade dos colaboradores utilizando Matplotlib e Seaborn

Fonte: De autoria própria

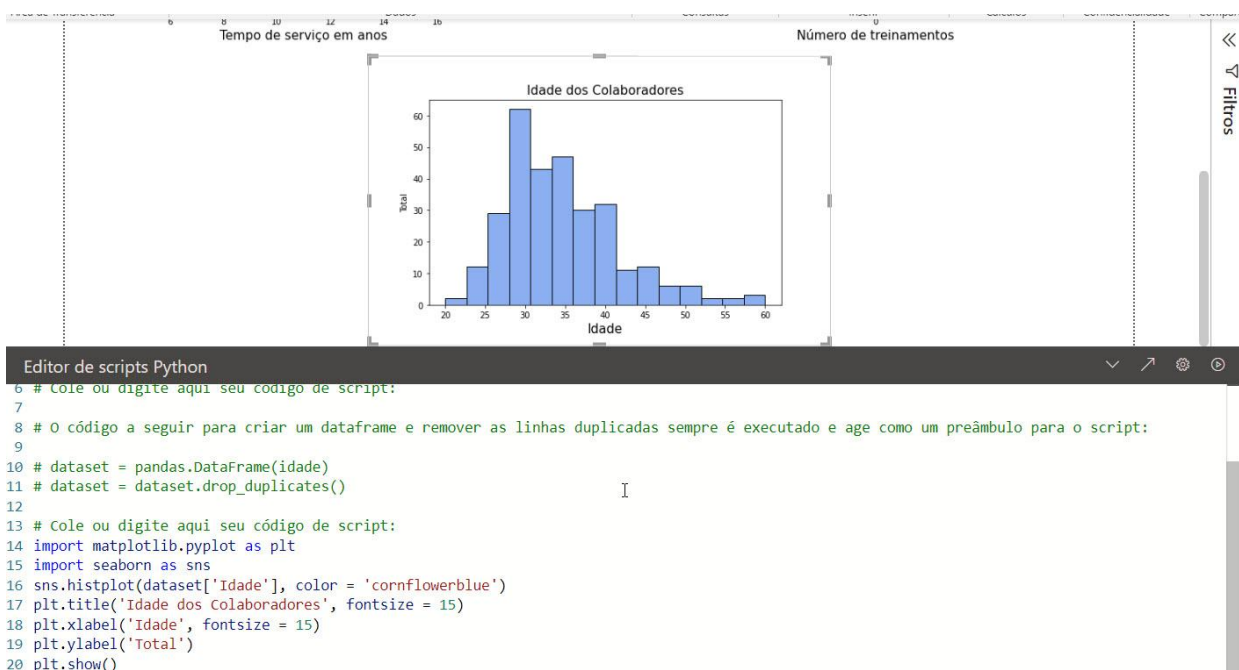


Figura 10 - Tela do editor de scripts Python no Power BI com o código para gerar um histograma da idade dos colaboradores.

Fonte: De autoria própria

- Pergunta 3 - Qual o número de treinamentos mais frequente?

```
import matplotlib.pyplot as plt
import seaborn as sns
##grafico violino
sns.violinplot(dataset['numero_treinamentos'], color =
'dodgerblue')
plt.title('Treinamentos feitos pelos colaboradores', fontsize = 15)
plt.xlabel('Número de treinamentos', fontsize = 15)
```

```
plt.ylabel('Frequência')
plt.show()
```

Trecho de código 14 - Código para geração de gráfico de violino utilizando Matplotlib e Seaborn para representar o número de treinamentos realizados por colaboradores.

Fonte: De autoria própria

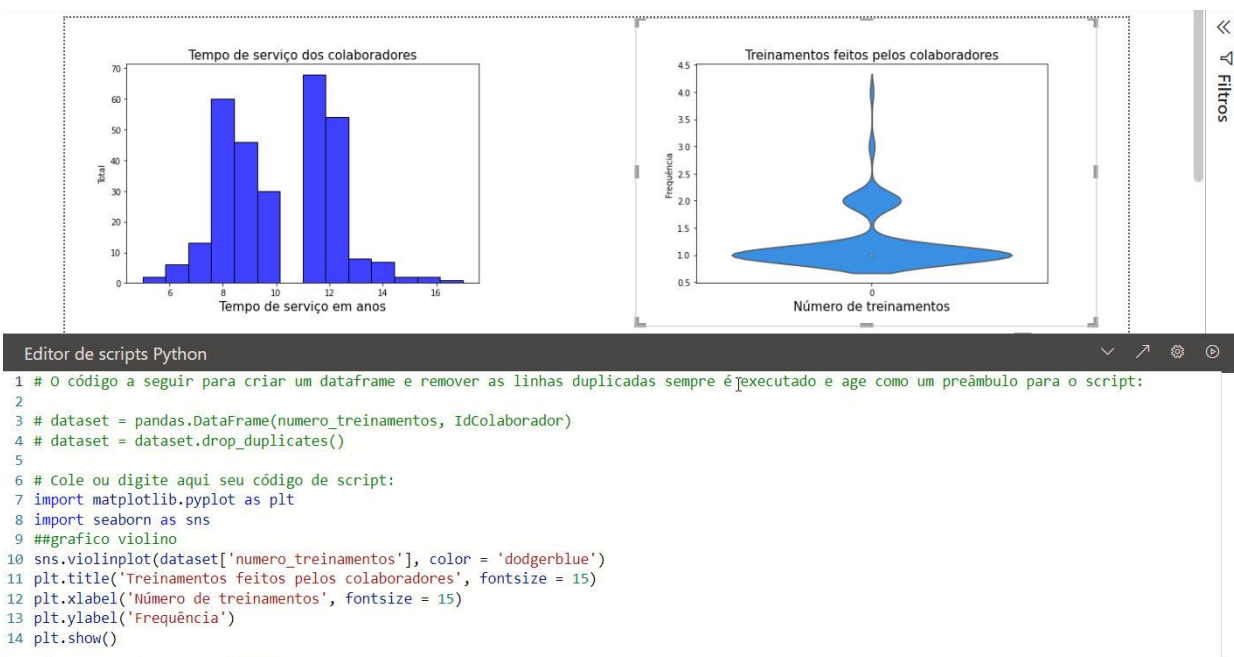


Figura 11 - Tela do editor de scripts Python no Power BI com o código para gerar um gráfico de violino para representar o número de treinamentos.

Fonte: De autoria própria

- Pergunta 4 - Qual a proporção dos colaboradores por fonte de recrutamento?

```
import matplotlib.pyplot as plt
dataset['FonteRecrutamento'].value_counts().sort_values().plot.bar(
color = 'steelblue', figsize = (10,6))
plt.title('Fonte de recrutamento', fontsize = 15)
plt.tight_layout()
plt.xticks(rotation = 45)
plt.ylabel('Total')
plt.show()
```

Trecho de código 15 - Código para geração de gráfico de barras utilizando Matplotlib para representar a proporção de fonte de recrutamento por colaboradores

Fonte: De autoria própria

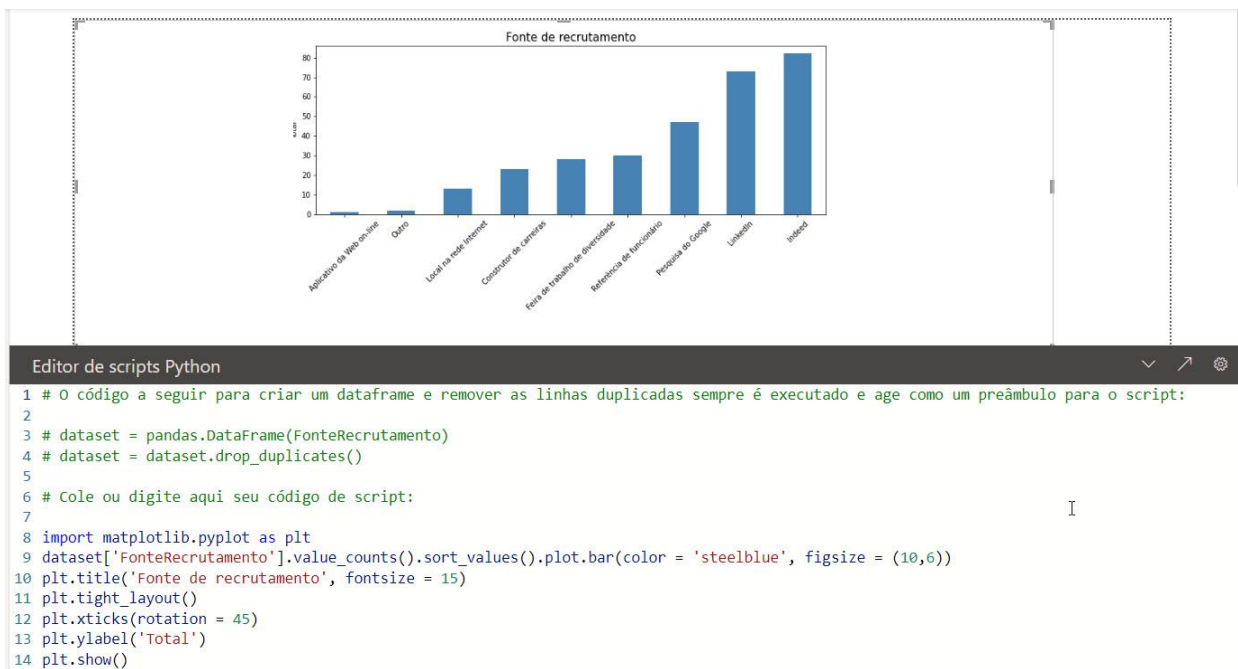


Figura 12 - Tela do editor de scripts Python no Power BI com o código para gerar um gráfico de barras para representar o número de treinamentos.

Fonte: De autoria própria

- Pergunta 5 - Quantos colaboradores têm o maior número de avaliações?

```

import matplotlib.pyplot as plt
dataset_ordenado = dataset.sort_values(by='NivelSatisfacao',
ascending=False)
num_colaboradores_por_nivel =
dataset_ordenado['NivelSatisfacao'].value_counts().sort_index()
# Plotar o gráfico de barras com o eixo x em ordem crescente
num_colaboradores_por_nivel.sort_index().plot(kind='bar',
figsize=(8, 6), color = 'navy')
plt.xlabel('Nível de Satisfação')
plt.ylabel('Número de Colaboradores')
plt.title('Número de Colaboradores por Nível de Satisfação')
plt.show()

```

Trecho de código 16 - Código para geração de gráfico de barras utilizando Matplotlib para representar o número de colaboradores em cada nível de satisfação.

Fonte: De autoria própria



Figura 13 - Tela do editor de scripts Python no Power BI com o código para gerar um gráfico de barras para representar o número de colaboradores em cada nível de satisfação.

Fonte: De autoria própria

- Pergunta 6 - Qual o maior salário por departamento?

```

import matplotlib.pyplot as plt
dados_salario = dataset.sort_values(by='Salario', ascending=True,
ignore_index=True)
plt.figure(figsize=(10,6))
plt.bar(dados_salario['Departamento'], dados_salario['Salario'],
color = 'cadetblue')
plt.title('Maior Salário por Departamento')
plt.ylabel('Maior Salário')
plt.xticks(rotation = 45)
plt.tight_layout()
plt.show()

```

Trecho de código 17 - Código para geração de gráfico de barras utilizando Matplotlib para representar e ordenar os dados pelo salário de forma crescente

Fonte: De autoria própria



Figura 14 - Tela do editor de scripts Python no Power BI com o código para gerar um gráfico de barras para representar e ordenar os dados pelo salário de forma crescente.

Fonte: De autoria própria

6 - CONCLUSÃO

Este trabalho teve como objetivo mostrar de forma simplificada, o funcionamento do uso de scripts *Python* dentro do *Power BI* e demonstrar como é feita a configuração do *Python* e do *Power BI*, para facilitar trabalhos futuros.

Em conclusão, o uso de gráficos é fundamental para a visualização e interpretação de dados em diversas áreas. Tanto *Python* quanto *Power BI* oferecem recursos poderosos para criar visualizações de dados impactantes e informativas.

Ao combinar as habilidades de programação e visualização de dados do *Python* com a facilidade de uso e interatividade do *Power BI*, é possível criar gráficos e dashboards eficazes e personalizados.

No entanto, é importante ter em mente que a escolha da ferramenta certa depende das necessidades e objetivos específicos do projeto. É essencial avaliar cuidadosamente as opções disponíveis e selecionar a ferramenta mais adequada para as necessidades do usuário.

Em suma, a combinação de *Python* e *Power BI* oferece uma ampla gama de recursos para criar gráficos e visualizações de dados poderosos e impactantes. Essa abordagem pode ajudar a melhorar a tomada de decisões e a compreensão de dados em diversas áreas, desde negócios até ciência e pesquisa.

7 - REFERÊNCIAS BIBLIOGRÁFICAS

ALURA, Power BI: o que é?. 2022. Disponível em: <<https://www.alura.com.br/artigos/power-bi-o-que-e/>>. Acesso em: 21 de set. 2022.

ALURA, Matplotlib uma biblioteca Python para gerar gráficos interessantes. 2023. Disponível em: <<https://www.alura.com.br/artigos/criando-graficos-no-python-com-a-matplotlib/>>. Acesso em: 10 de abr. 2023.

CETAX. O que é o Microsoft Power BI?. 2022. Disponível em: <<https://cetax.com.br/o-que-e-o-microsoft-power-bi/>>. Acesso em: 10 de abr. 2023.

DEUTSCH, P., Microsoft Power BI Complete Reference: Bringing data to life. Disponível em: <<https://www.worldcat.org/pt/title/microsoft-power-bi-complete-reference-bring-your-data-to-life-with-the-powerful-features-of-microsoft-power-bi/oclc/1081000272/>>. Acesso em: 10 de abr. 2023

HUNTER, J. D., Matplotlib: A 2D graphics environment. 9. ed. IEEE Annals of the History of Computing. 2007. 90-95 p.

INFNET, Instituto. O que são indicadores no Power BI e como criá-los?. 2022. Disponível em: <<https://blog.infnet.com.br/businessintelligence/o-que-sao-indicadores-no-power-bi-como-cria-los/>>. Acesso em: 10 de abr. 2023.

MATPLOTLIB, Matplotlib 3.7.1 documentation, Installation. Disponível em: <<https://matplotlib.org/stable/index.html/>>. Acesso em: 01 de nov. 2022.

MCKINNEY, Wes. Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 2010.

MCKINNEY, Wes, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. 2017. O'Reilly Media.

MICROSOFT, Instalar o Visual Studio. 2022. Disponível em: <<https://learn.microsoft.com/pt-br/visualstudio/install/install-visual-studio?view=vs-2022/>>. Acesso em: 15 de nov. 2022.

MICROSOFT, Executar scripts do Python no Power BI Desktop. 2022. Disponível em: <<https://learn.microsoft.com/pt-br/power-bi/connect-data/desktop-python-scripts/>>. Acesso em: 30 de set. 2022.

MICROSOFT, Obter Power BI Desktop. 2023. Disponível em: <<https://learn.microsoft.com/pt-br/power-bi/fundamentals/desktop-get-the-desktop#install-as-an-app-from-the-microsoft-store/>>. Acesso em: 20 de fev. 2023.

MONTENEGRO, Bruna. Power BI: o que é, para o que é usado e como funciona?. 2023. Disponível em: <<https://ebaonline.com.br/blog/o-que-e-power-bi#:~:text=O%20Power%20BI%20%C3%A9%20uma%20ferramenta%20de%20avalia%C3%A7%C3%A3o%20e%20visualiza%C3%A7%C3%A3o,de%20maneira%20simples%20e%20intuitiva>>. Acesso em: 10 de abr. 2023.

PANDAS, pandas.DataFrame.plot.bar. Disponível em: <<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.bar.html/>>. Acesso em: 06 de abr. 2023.

PANDAS, pandas.DataFrame.plot.hist. Disponível em: <<https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.hist.html/>>. Acesso em: 06 de abr. 2023.

PYTHON, Instalando Módulos Python. Disponível em: <<https://docs.python.org/pt-br/3.7/installing/index.html/>>. Acesso em: 01 de nov. 2022.

REZNIK, A., Python for Power BI: A Comprehensive Guide to Integrating Python and Power BI for Business Intelligence and Data Analysis. 2021. Packt Publishing.

VAN ROSSUM, G.; DRAKE, F. L. Python 3 : reference manual. United States: Sohobooks, 2009.