



# POWER BI – MÓDULO I

---

## SEMANA 06

*Prof. Kaio Mesquita*



# Objetivos do curso

---

- *Instalar e configurar a ferramenta Power BI;*
- *Diferenciar e compreender conceitos de Business Intelligence, Big Data e Data Science;*
- *Compreender e avaliar problemas de maneira analítica, estratégica e fenomenológica;*
- *Criar tabelas de relacionamento;*
- *Apresentar resultados interativos;*
- *Tratar, Transformar e Carregar(ETL – Extract, Transform e Load) dados de forma otimizada;*
- *Analisar Séries Temporais;*
- *Criar Dashboard práticos e objetivos.*

# Programa do curso

---

✓ 1. Primeiros passos com o Power BI (29/10)

✓ 2. Business Intelligence no Power BI (05/11)

✓ Modelagem, Relacionamento e DAX (12/11)

✓ 4. Limpeza, Transformação, Séries Temporais, Agregação e Filtros (19/11)

✓ Gráficos e Mapas (26/11)

✓ 6. Linguagem Python com Power BI (03/12)



# \* Projeto Prático

---

- ✓ Conclusão do Projeto prático

## 6. Linguagem Python e Power BI para Manipulação de Dados

---

- ✓ 6.1 Linguagem Python, R e DAX
- 6.2 Instalando o interpretador da linguagem python
- 6.3 Configurando a Integração entre o Power BI e a Linguagem Python
- 6.4 Projeto 3 – Power BI com Python (Introdução)
- 6.5 Projeto 3 – Análise Exploratória, Limpeza e Transformação de dados
- 6.6 Projeto 3 – Compreendendo o problema de desbalanceamento de Classe
- 6.7 Projeto 3 – Resolvendo o problema de negócios
- 6.8 Projeto 3 – Construindo o Dashboard analítico no Power BI

## 6.1 Linguagem Python, R e DAX

---

- ✓ A linguagem Python é amplamente difundida pela sua facilidade relacionado à outras linguagens, tendo uma curva de aprendizagem bem mais prática do que a linguagem R, por exemplo. Esta segunda cita é amplamente difundida para análises estatísticas e até mais robusta neste propósito, porém a linguagem Python é bem mais versátil e com um grande gama de funcionalidades e pacotes para tratamento e análise de dados, por isso vamos focar nela, neste capítulo.





## 6.1 Linguagem Python, R e DAX

---

- ✓ O Power BI oferece funcionalidades básicas de manipulação de dados e criação de gráficos. Ao usar Dax conseguimos customizar o processo de análise de dados e criar regras de negócio avançadas. Com Linguagem R trazemos para o Power BI uma poderosa plataforma de análise estatística, além de gráficos avançados. **Com a Linguagem Python temos manipulação de dados, modelos de Machine Learning, IA e gráficos avançados.**

Dessa forma a linguagem python não é diferente, e detém diversas bibliotecas. Aqui temos as principais bibliotecas do python para análise de dados:

- **Matplotlib:** Criação de gráficos
- **Seaborn:** Criação de gráficos estatísticos específicos
- **Pandas:** Trabalha com grandes volumes de dados para tratamento (um excel melhorado e com mais funcionalidades)
- **NumPy:** Manipulação matemática de dados, principalmente com matrizes.

## 6.2 Instalando o interpretador da linguagem Python

---

- ✓ O Power BI oferece funcionalidades básicas de manipulação de dados e criação de gráficos. Ao usar Dax conseguimos customizar o processo de análise de dados e criar regras de negócio avançadas. Com Linguagem R trazemos para o Power BI uma poderosa plataforma de análise estatística, além de gráficos avançados. **Com a Linguagem Python temos manipulação de dados, modelos de Machine Learning, IA e gráficos avançados.**

Dessa forma a linguagem python não é diferente, e detém diversas bibliotecas. Aqui temos as principais bibliotecas do python para análise de dados:

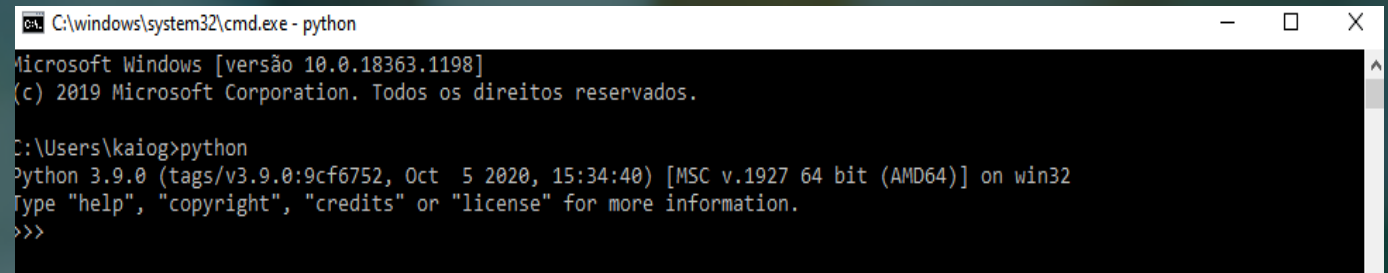
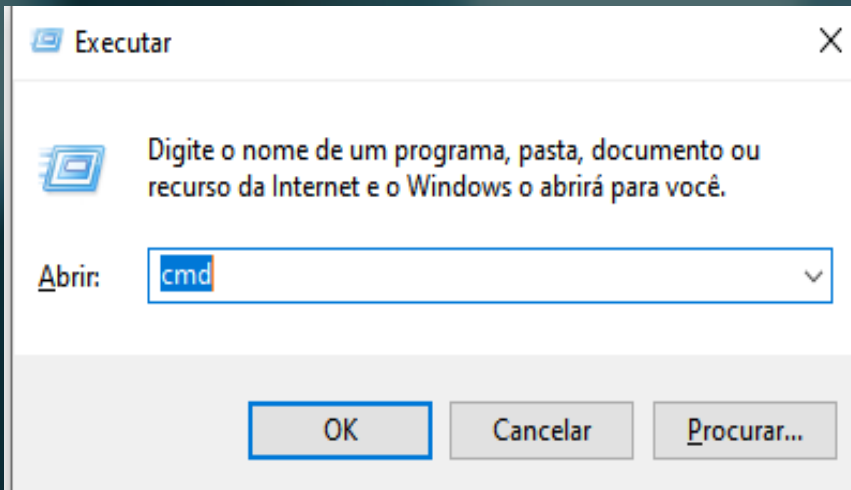
- **Matplotlib:** Criação de gráficos
- **Seaborn:** Criação de gráficos estatísticos específicos
- **Pandas:** Trabalha com grandes volumes de dados para tratamento (um excel melhorado e com mais funcionalidades)
- **NumPy:** Manipulação matemática de dados, principalmente com matrizes.



## 6.2 Instalando o interpretador da linguagem Python

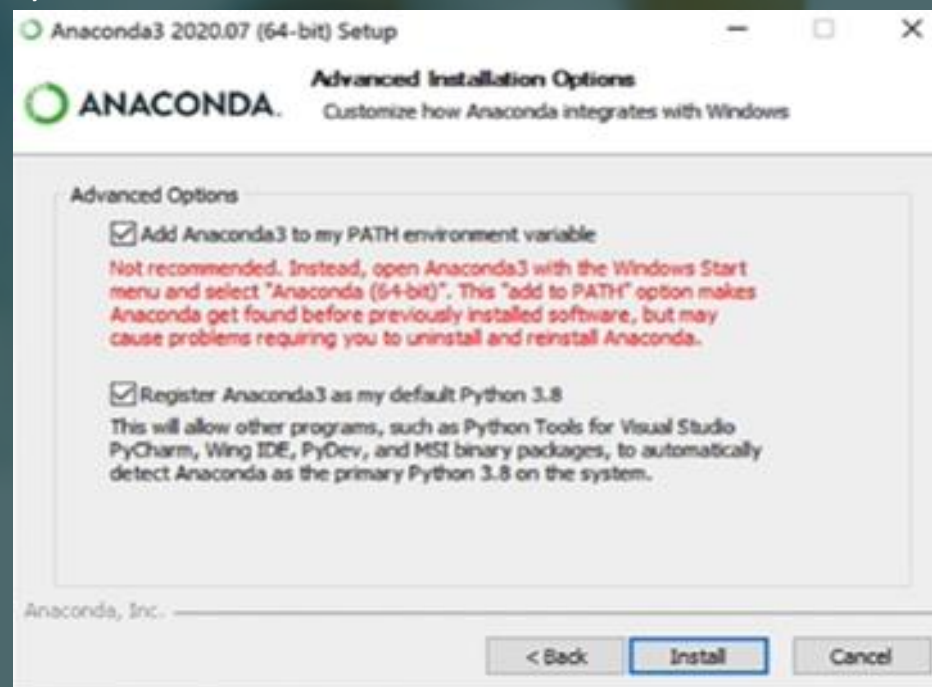
---

- ✓ Para verificar se sua máquina tem o interpretador do Python instalado, aperte **Windows + R** e na tela que irá aparecer digite **CMD**.



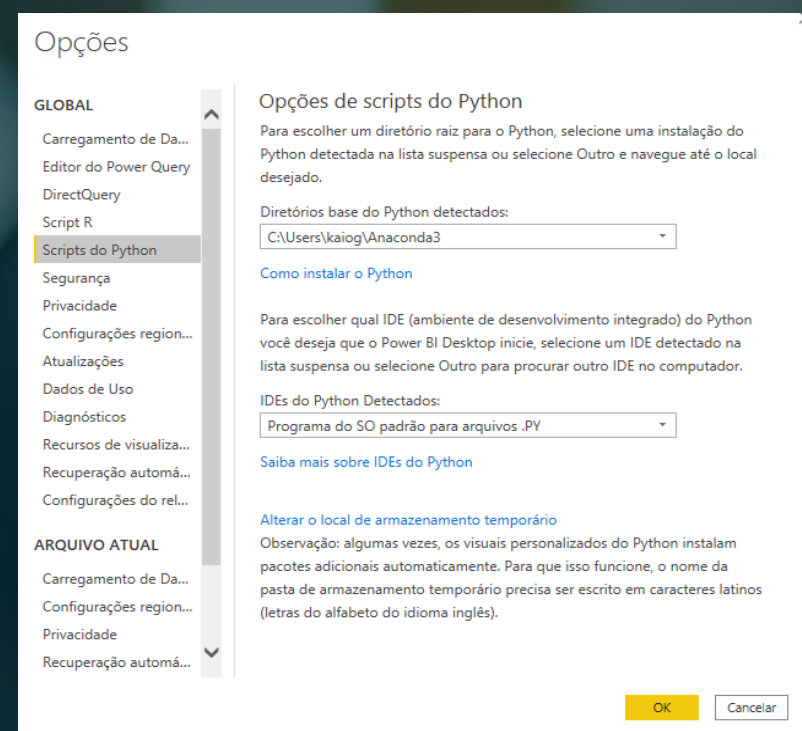
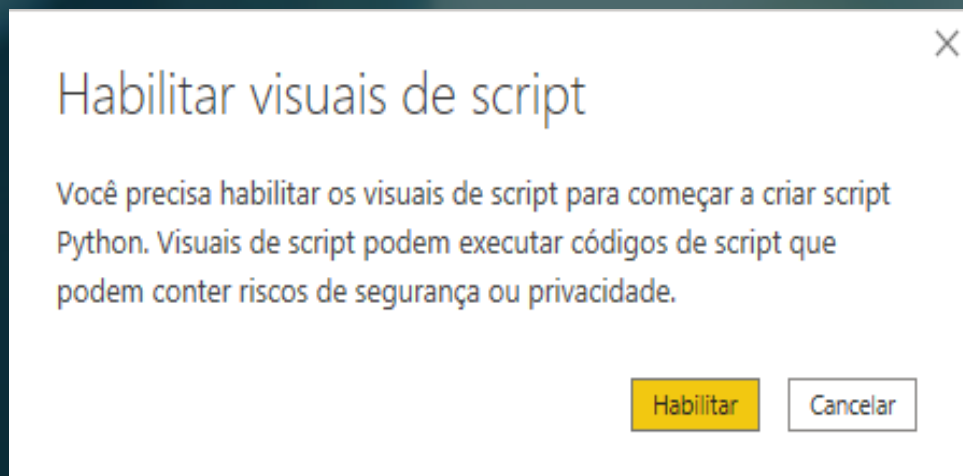
## 6.2 Instalando o interpretador da linguagem Python

- ✓ Após baixar o anaconda para sua máquina, clique em next e aceite os termos de usuário (Lembrando que é uma ferramenta opensource, gratuita).



## 6.3 Configurando a Integração entre o Power BI e Linguagem Python

- ✓ Abra o Power BI, e pesquise o ícone do python no menu de visualização. Será perguntado se você quer habilitar os visuais de script.
- ✓ Agora vá na guia de opções e clique sobre o menu de script do python. Configure para o diretório do anaconda. Você também tem a opção de escolher uma IDE. Clique em ok.

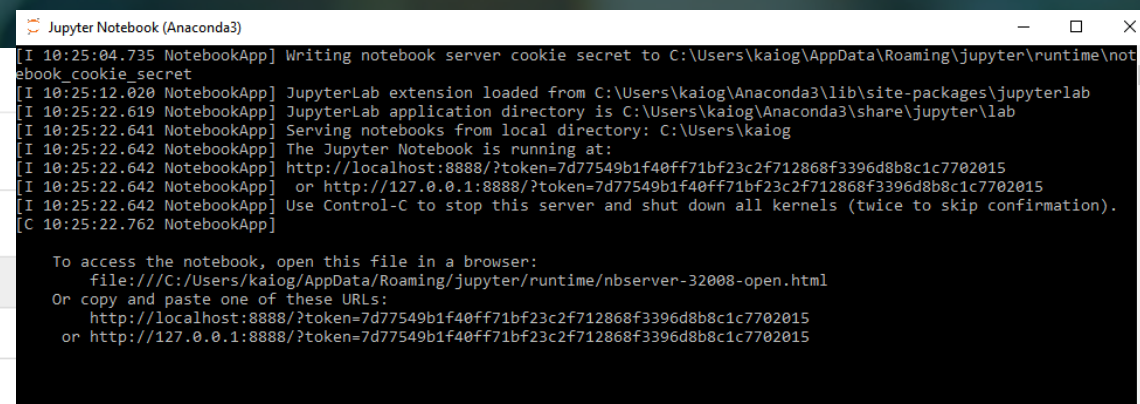




## 6.3 Configurando a Integração entre o Power BI e Linguagem Python

- ✓ Uma IDE é um ambiente integrado de desenvolvimento. Vamos mostrar como exemplo o Jupyter Notebook, que não é uma IDE, mas facilita muito a utilização e aplicação da linguagem via Browser. Baixe o arquivo referente a atividade.csv. Copie o endereço onde seu arquivo foi salvo e abra novamente o CMD. Dentro do CMD digite CD e o endereço copiado (basta clicar com o botão direito do mouse e será colado o que tiver sido copiado). Agora você está dentro do diretório onde está o seu arquivo. Digite agora Jupyter Notebook.

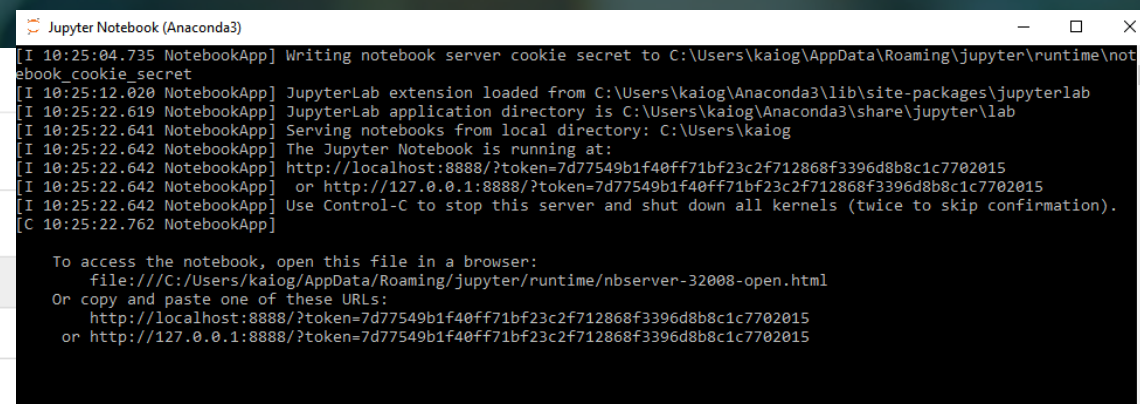
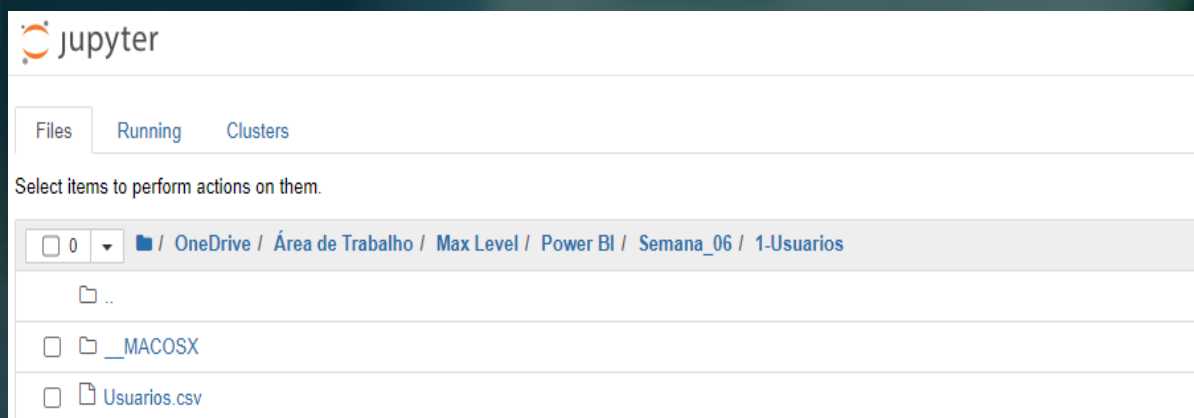
É importante ressaltar que somente conseguirá acesso ao jupyter notebook enquanto o prompt de comando estiver aberto.



## 6.3 Configurando a Integração entre o Power BI e Linguagem Python

- ✓ Uma IDE é um ambiente integrado de desenvolvimento. Vamos mostrar como exemplo o Jupyter Notebook, que não é uma IDE, mas facilita muito a utilização e aplicação da linguagem via Browser. Baixe o arquivo referente a atividade.csv. Copie o endereço onde seu arquivo foi salvo e abra novamente o CMD. Dentro do CMD digite CD e o endereço copiado (basta clicar com o botão direito do mouse e será colado o que tiver sido copiado). Agora você está dentro do diretório onde está o seu arquivo. Digite agora Jupyter Notebook.

É importante ressaltar que somente conseguirá acesso ao jupyter notebook enquanto o prompt de comando estiver aberto.



## 6.3 Configurando a Integração entre o Power BI e Linguagem Python

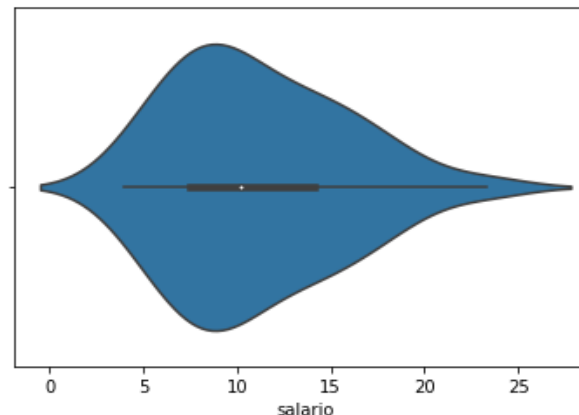
- ✓ Nesta tela digite o seguinte código. Nele estamos importando a biblioteca Pandas, salvando o nosso arquivo csv em uma variável, depois importamos matplotlib e o seaborn e geramos um gráfico de violino com a coluna salário.

```
In [1]: #Exemplo 1 - Violin Plot

#Carrega o Dataset
import pandas as pd
dataset = pd.read_csv("Usuarios.csv")

#Cria o gráfico

import matplotlib.pyplot as plt
import seaborn as sns
sns.violinplot(x="salario", data=dataset)
plt.show()
```





## 6.3 Configurando a Integração entre o Power BI e Linguagem Python

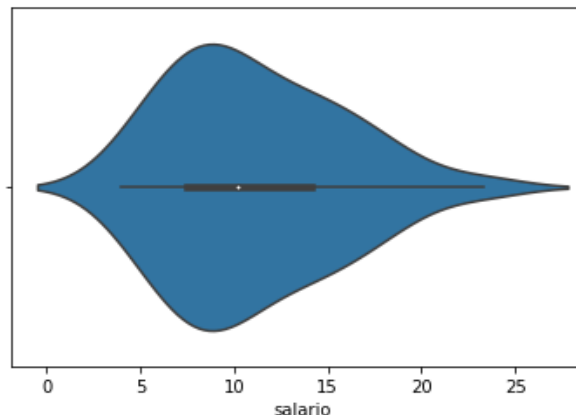
- ✓ Nesta tela digite o seguinte código. Nele estamos importando a biblioteca Pandas, salvando o nosso arquivo csv em uma variável, depois importamos matplotlib e o seaborn e geramos um gráfico de violino com a coluna salário.

```
In [1]: #Exemplo 1 - Violin Plot

#Carrega o Dataset
import pandas as pd
dataset = pd.read_csv("Usuarios.csv")

#Cria o gráfico

import matplotlib.pyplot as plt
import seaborn as sns
sns.violinplot(x="salario", data=dataset)
plt.show()
```



## 6.3 Configurando a Integração entre o Power BI e Linguagem Python

- ✓ Agora vamos no python e importe os dados. Não se esqueça de verificar a regionalidade e o tipo do arquivo. Salve o arquivo e clique sobre o menu de visualização para gráficos do python.

The screenshot displays the Power BI interface with the 'Dados' (Data) tab selected. The 'Área de Transferência' (Transfer Area) on the left shows a Python icon. The 'Consultas' (Queries) pane is empty. The 'Editor de scripts Python' (Python script editor) is open, showing a warning message: 'As linhas duplicadas serão removidas dos dados.' (Duplicate lines will be removed from the data). Below the warning, the script contains the following code:

```
1 # O código a seguir para criar um dataframe e remover as linhas duplicadas sempre é executado e age como um preâmbulo para o script:
2
3 # dataset = pandas.DataFrame(salario)
4 # dataset = dataset.drop_duplicates()
5
6 # Cole ou digite aqui seu código de script:
```

Below the script editor, a visualization titled 'salario' is shown, displaying a violin plot of the 'salario' variable. The x-axis is labeled 'salario' and ranges from 0 to 20. The plot shows a distribution of salaries with a peak around 10.

At the bottom of the screen, a message reads: 'Arraste os campos para a área Valores no painel Visualização para iniciar a criação de scripts.' (Drag the fields to the Values area in the Visualization pane to start creating scripts.)

## 6.3 Configurando a Integração entre o Power BI e Linguagem Python

---

- ✓ Obs.: Se por acaso não aparecer o gráfico ou ocorrer algum erro, provavelmente sua máquina não têm as bibliotecas instaladas. Dessa forma terá que instalar manualmente. Para isso abra o prompt de comando e digite **pip install seaborn**. Repita para todas as outras bibliotecas que formos usar.



## 6.4 Projeto 3 – Power BI com Python (Introdução)

---

- ✓ Primeiro vamos a definição do problema. Para ler a definição abra no jupyter notebook o arquivo referente ao projeto final.

### Definição do Problema

A empresa ZuZa Inc coletou dados do ano anterior com diversos atributos de funcionários de todos os setores. O objetivo é estudar a relação desses atributos e o impacto na promoção dos funcionários. 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 a Correlação Entre os Atributos dos Funcionários?
- Pergunta 2 - Qual o Tempo de Serviço da Maioria dos Funcionários?
- Pergunta 3 - Qual Avaliação do Ano Anterior Foi Mais Comum?
- Pergunta 4 - Qual a Distribuição das Idades dos Funcionários?
- Pergunta 5 - Qual o Número de Treinamentos Mais Frequente?
- Pergunta 6 - Qual a Proporção dos Funcionários Por Canal de Recrutamento?
- Pergunta 7 - Qual a Relação Entre a Promoção e a Avaliação do Ano Anterior?

Seu trabalho é limpar e preparar os dados e então construir um dashboard no Power BI para apresentar seus resultados.

## 6.4 Projeto 3 – Power BI com Python (Introdução)

---

- ✓ Primeiro vamos a definição do problema. Para ler a definição abra no jupyter notebook o arquivo referente ao projeto final.

### Definição do Problema

A empresa ZuZa Inc coletou dados do ano anterior com diversos atributos de funcionários de todos os setores. O objetivo é estudar a relação desses atributos e o impacto na promoção dos funcionários. 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 a Correlação Entre os Atributos dos Funcionários?
- Pergunta 2 - Qual o Tempo de Serviço da Maioria dos Funcionários?
- Pergunta 3 - Qual Avaliação do Ano Anterior Foi Mais Comum?
- Pergunta 4 - Qual a Distribuição das Idades dos Funcionários?
- Pergunta 5 - Qual o Número de Treinamentos Mais Frequente?
- Pergunta 6 - Qual a Proporção dos Funcionários Por Canal de Recrutamento?
- Pergunta 7 - Qual a Relação Entre a Promoção e a Avaliação do Ano Anterior?

Seu trabalho é limpar e preparar os dados e então construir um dashboard no Power BI para apresentar seus resultados.

## 6.4 Projeto 3 – Power BI com Python (Introdução)

- ✓ Para cada código é necessário sempre importar as bibliotecas. Perceba que cada célula detém uma linha de importação das mesmas bibliotecas. No caso do jupyter notebook só é necessário importar uma única vez cada biblioteca, porém como vamos executar no python é necessário importar novamente sempre que criarmos um novo gráfico.

```
In [5]: # Imports
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

```
In [6]: # Versões dos pacotes usados neste jupyter notebook
%reload_ext watermark
%watermark -a "Data Science Academy" --iversion
```

Author: Data Science Academy

```
pandas      : 1.0.5
seaborn      : 0.10.1
numpy        : 1.18.5
matplotlib   : 3.2.2
```



## 6.4 Projeto 3 – Power BI com Python (Introdução)

- ✓ Para cada código é necessário sempre importar as bibliotecas. Perceba que cada célula detém uma linha de importação das mesmas bibliotecas. No caso do jupyter notebook só é necessário importar uma única vez cada biblioteca, porém como vamos executar no python é necessário importar novamente sempre que criarmos um novo gráfico.

```
In [5]: # Imports
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

```
In [6]: # Versões dos pacotes usados neste jupyter notebook
%reload_ext watermark
%watermark -a "Data Science Academy" --iversion
```

Author: Data Science Academy

```
pandas      : 1.0.5
seaborn      : 0.10.1
numpy        : 1.18.5
matplotlib   : 3.2.2
```

## 6.4 Projeto 3 – Power BI com Python (Introdução)

- ✓ Agora carregue os dadosRH.csv e certifique-se que ambos estejam na mesma pasta. Depois chame a função head() que executa e exibe os primeiros dados do nosso dataset. Essa função é excelente para uma visualização resumida dos dados importados. No caso da última coluna, 1 significa sim, e 0 não. Quando utilizamos a função shape() ela retorna o número de linhas e de colunas.

```
Carregando os Dados

In [7]: dadosRH = pd.read_csv('dadosRH.csv')

In [8]: dadosRH.head()

Out[8]:
```

	id_funcionario	departamento	regiao	educacao	genero	canal_recrutamento	numero_treinamentos	idade	aval_ano_anterior	tempo_servico	promovido
0	65438	Marketing	Sudeste	Mestrado ou Doutorado	F	Outsourcing	1	35	5.0	8	0
1	65141	Logistica	Nordeste	Superior	M	Outro	1	30	5.0	4	0
2	7513	Marketing	Sudeste	Superior	M	Outsourcing	1	34	3.0	7	0
3	2542	Marketing	Norte	Superior	M	Outro	2	39	1.0	10	0
4	48945	Tecnologia	Nordeste	Superior	M	Outro	1	45	3.0	2	0

```
In [9]: dadosRH.shape

Out[9]: (54808, 11)
```

## 6.5 Projeto 3 – Análise exploratória, Limpeza e Transformação de Dados

---

- ✓ Agora vamos fazer um processo de tratamento dos dados utilizando a linguagem python. Primeiramente vamos verificar se existem dados nulos no nosso conjunto de dados e se existir quantos para cada coluna.

### Análise Exploratória, Limpeza e Transformação de Dados

Vamos manipular os dados e corrigir eventuais problemas.

```
In [8]: dadosRH.isnull().sum()
```

```
Out[8]: id_funcionario      0
         departamento      0
         regioa            0
         educacao          2409
         genero            0
         canal_recrutamento 0
         numero_treinamentos 0
         idade            0
         aval_ano_anterior  4124
         tempo_servico      0
         promovido         0
         dtype: int64
```



## 6.5 Projeto 3 – Análise exploratória, Limpeza e Transformação de Dados

- ✓ As colunas educação e aval\_ano\_anterior têm muitos dados em branco, dessa forma precisamos tratar. Antes de executar os tratamentos verifique essas colunas. Mas tome cuidado pois esse agrupamento não considera os valores nulos.

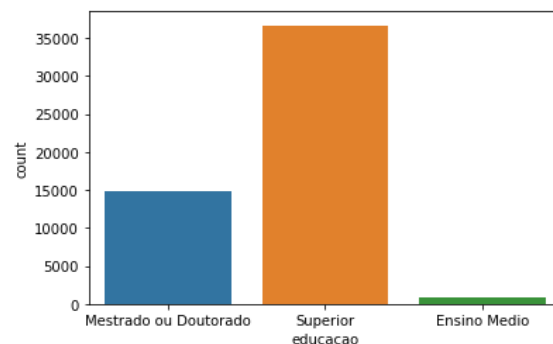
```
In [9]: dadosRH.groupby(['educacao']).count()
```

```
Out[9]:
```

	id_funcionario	departamento	regiao	genero	canal_recrutamento
educacao					
Ensino Medio	805	805	805	805	805
Mestrado ou Doutorado	14925	14925	14925	14925	14925
Superior	36669	36669	36669	36669	36669

```
In [10]: sns.countplot(dadosRH['educacao'])
```

```
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x1aa6bd8bbb0>
```



Vamos aplicar imputação e preencher os valores ausentes.

```
In [10]: dadosRH['educacao'].fillna(dadosRH['educacao'].mode()[0], inplace = True)
```

```
In [11]: dadosRH['aval_ano_anterior'].fillna(dadosRH['aval_ano_anterior'].median(), inplace = True)
```

```
In [12]: dadosRH.isnull().sum()
```

```
Out[12]: id_funcionario      0
departamento              0
regiao                    0
educacao                  0
genero                   0
canal_recrutamento       0
numero_treinamentos       0
idade                    0
aval_ano_anterior         0
tempo_servico             0
promovido                 0
dtype: int64
```

## 6.6 Projeto 3 – Compreendendo o problema de desbalanceamento de classes

- ✓ Para essa questão iremos utilizar a coluna de pessoas promovidas, primeiramente vamos verificar essas quantias com a seguinte função abaixo.

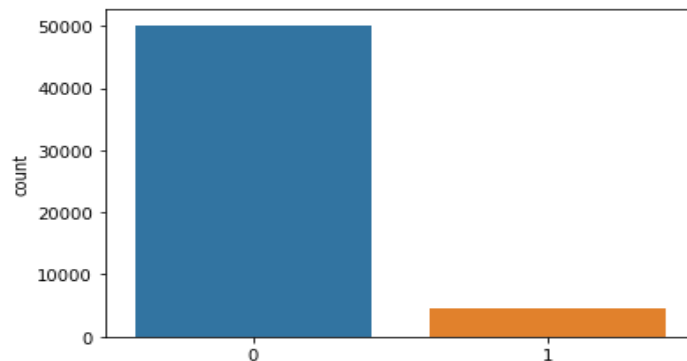
```
In [19]: dadosRH.groupby(['promovido']).count()
```

```
Out[19]:
```

	id_funcionario	departamento	regiao	educacao	genero	canal_recrutamento	numero_treinamentos	idade	aval_ano_anterior	tempo_servico
promovido										
0	50140	50140	50140	50140	50140	50140	50140	50140	50140	50140
1	4668	4668	4668	4668	4668	4668	4668	4668	4668	4668

```
In [20]: sns.countplot(dadosRH['promovido'])
```

```
Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x1aa6cc2d070>
```



## 6.6 Projeto 3 – Compreendendo o problema de desbalanceamento de classes

- ✓ Para essa questão iremos utilizar a coluna de pessoas promovidas, primeiramente vamos verificar essas quantias com a seguinte função abaixo.

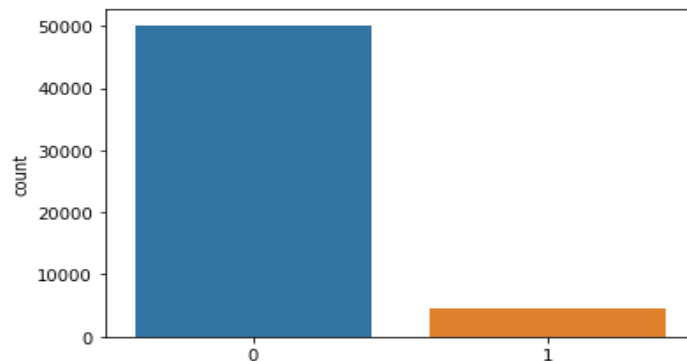
```
In [19]: dadosRH.groupby(['promovido']).count()
```

```
Out[19]:
```

	id_funcionario	departamento	regiao	educacao	genero	canal_recrutamento	numero_treinamentos	idade	aval_ano_anterior	tempo_servico
promovido										
0	50140	50140	50140	50140	50140	50140	50140	50140	50140	50140
1	4668	4668	4668	4668	4668	4668	4668	4668	4668	4668

```
In [20]: sns.countplot(dadosRH['promovido'])
```

```
Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x1aa6cc2d070>
```





## 6.6 Projeto 3 – Compreendendo o problema de desbalanceamento de classes

- ✓ Nessa técnica de amostragem iremos utilizar a biblioteca do sklearn. E transformaremos a classe minoritária (a de menor quantidade de dados) para que ela tenha a mesma quantidade da classe majoritária (parâmetro de comparação). Dessa forma utilizamos a função `resample` para realizar esta técnica e após executar verificaremos se está tudo ok com os novos dados.

```
In [21]: df_classe_majoritaria = dadosRH[dadosRH.promovido == 0]
df_classe_minoritaria = dadosRH[dadosRH.promovido == 1]
```

```
In [22]: df_classe_majoritaria.shape
```

```
Out[22]: (50140, 11)
```

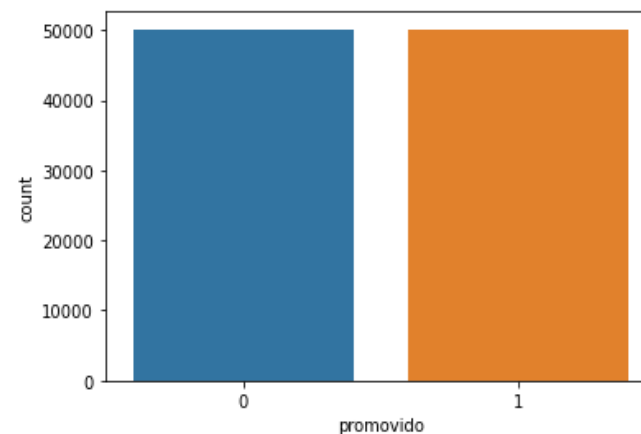
```
In [23]: df_classe_minoritaria.shape
```

```
Out[23]: (4668, 11)
```

```
In [24]: # Upsample da classe minoritária
from sklearn.utils import resample
df_classe_minoritaria_upsampled = resample(df_classe_minoritaria,
                                             replace=True,
                                             n_samples=50140,
                                             random_state=42)
```

```
In [28]: sns.countplot(dadosRH_balanceados['promovido'])
```

```
Out[28]: <matplotlib.axes._subplots.AxesSubplot at 0x1aa6d0c9940>
```

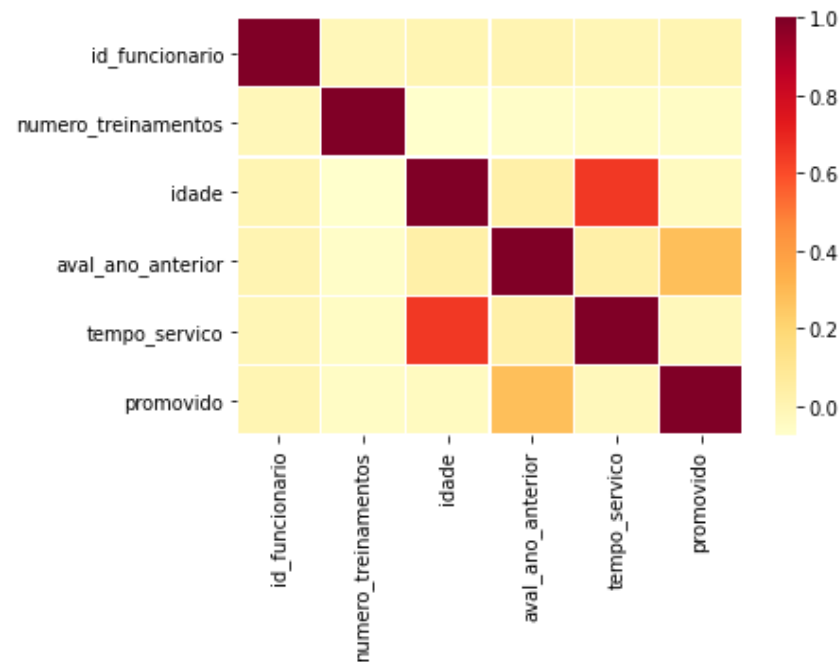


Os dados estão balanceados. Vamos salvar o dataset com os dados manipulados.

## 6.7 Projeto 3 – Resolvendo o Problema de Negócios

✓ Qual a correlação entre os atributos dos funcionários?

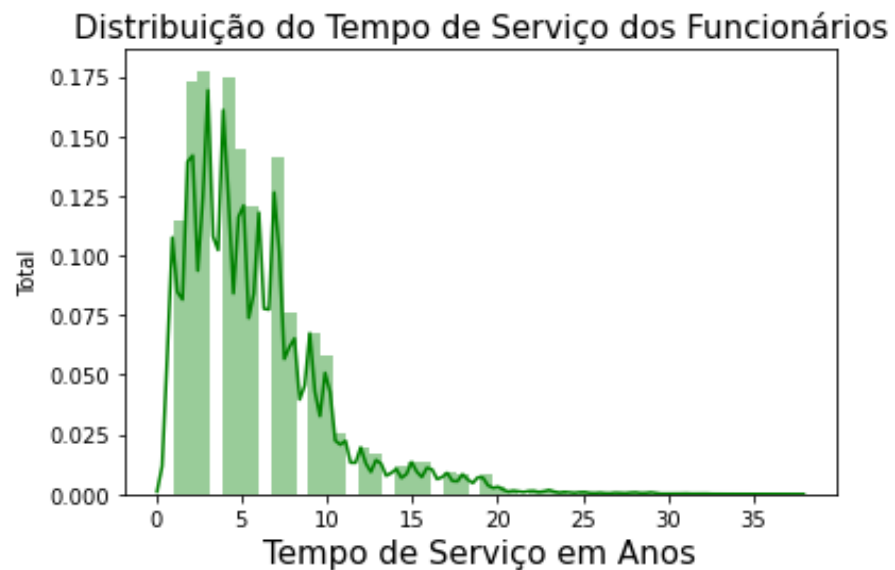
```
In [32]: import matplotlib.pyplot as plt
import seaborn as sns
corr = dataset.corr()
sns.heatmap(corr, cmap = "YlOrRd", linewidths = 0.1)
plt.show()
```



## 6.7 Projeto 3 – Resolvendo o Problema de Negócios

✓ Qual o tempo de Serviço da Maioria dos funcionários?

```
In [33]: import matplotlib.pyplot as plt
import seaborn as sns
sns.distplot(dataset['tempo_servico'], color = 'green')
plt.title('Distribuição do Tempo de Serviço dos Funcionários', fontsize = 15)
plt.xlabel('Tempo de Serviço em Anos', fontsize = 15)
plt.ylabel('Total')
plt.show()
```

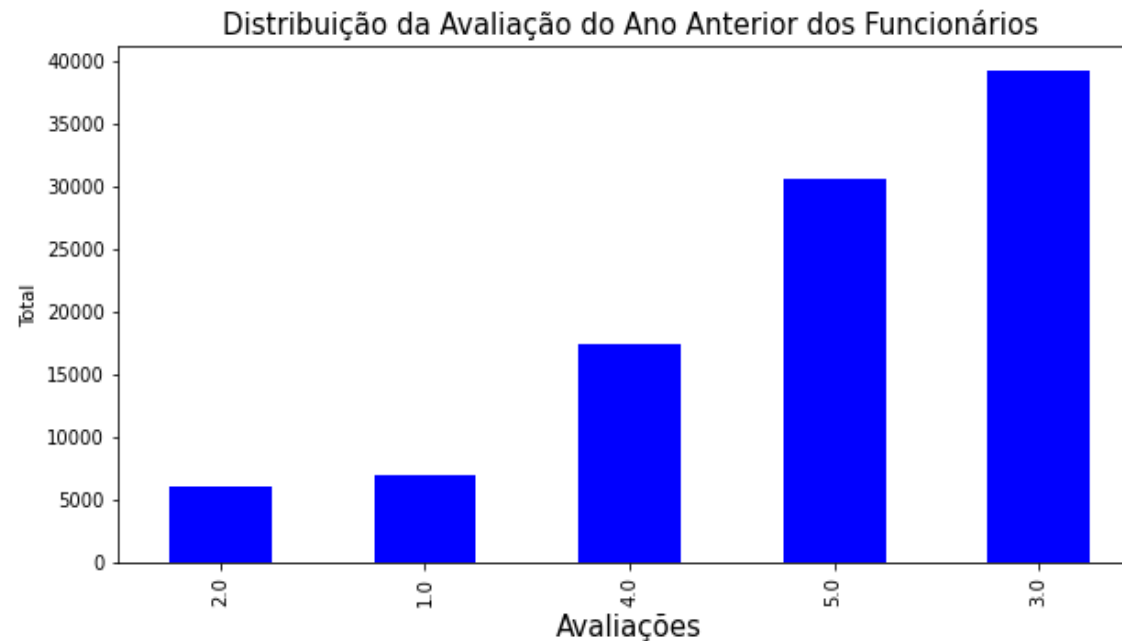




## 6.7 Projeto 3 – Resolvendo o Problema de Negócios

✓ Qual a avaliação do Ano anterior foi mais comum?

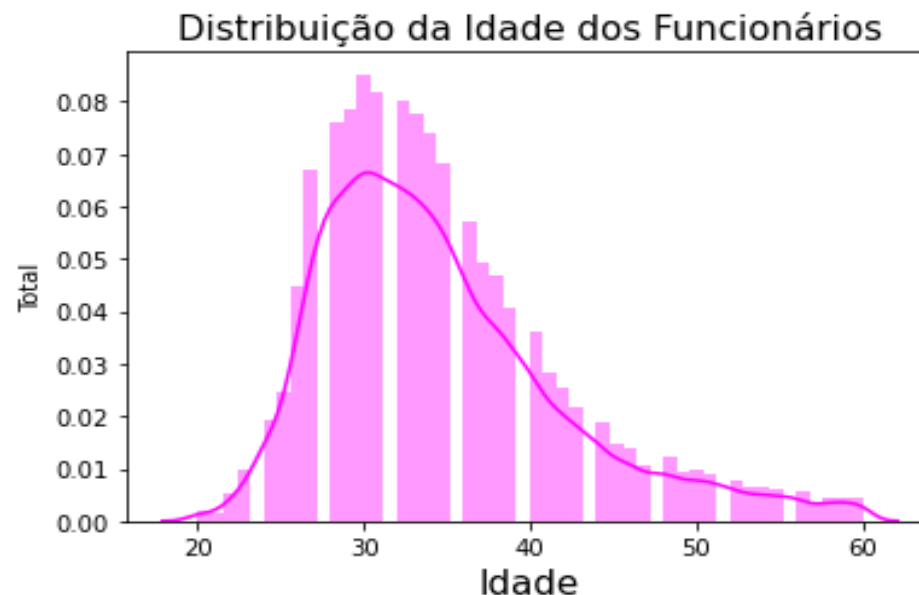
```
In [34]: import matplotlib.pyplot as plt
import seaborn as sns
dataset['aval_ano_anterior'].value_counts().sort_values().plot.bar(color = 'blue', figsize = (10, 5))
plt.title('Distribuição da Avaliação do Ano Anterior dos Funcionários', fontsize = 15)
plt.xlabel('Avaliações', fontsize = 15)
plt.ylabel('Total')
plt.show()
```



## 6.7 Projeto 3 – Resolvendo o Problema de Negócios

✓ Qual a distribuição das idades dos funcionários?

```
In [35]: import matplotlib.pyplot as plt
import seaborn as sns
sns.distplot(dataset['idade'], color = 'magenta')
plt.title('Distribuição da Idade dos Funcionários', fontsize = 15)
plt.xlabel('Idade', fontsize = 15)
plt.ylabel('Total')
plt.show()
```

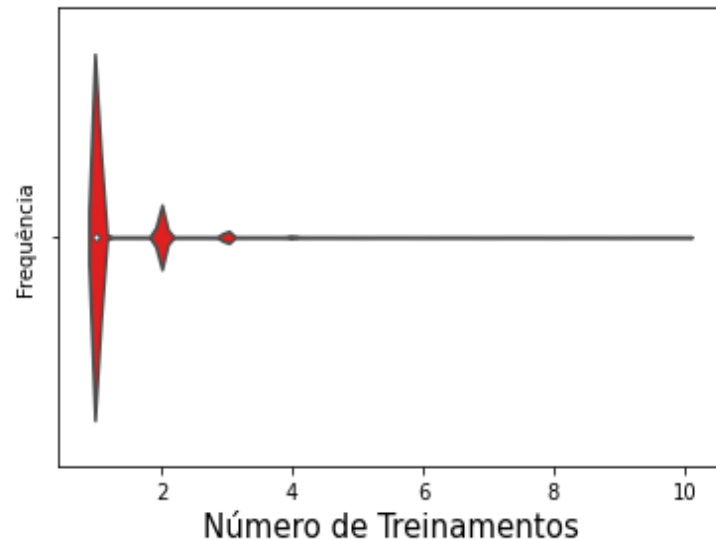


## 6.7 Projeto 3 – Resolvendo o Problema de Negócios

✓ Qual o Número de treinamentos mais frequente?

```
In [5]: import matplotlib.pyplot as plt
import seaborn as sns
sns.violinplot(dataset['numero_treinamentos'], color = 'red')
plt.title('Número de Treinamentos Feitos Pelos Funcionários', fontsize = 15)
plt.xlabel('Número de Treinamentos', fontsize = 15)
plt.ylabel('Frequência')
plt.show()
```

Número de Treinamentos Feitos Pelos Funcionários



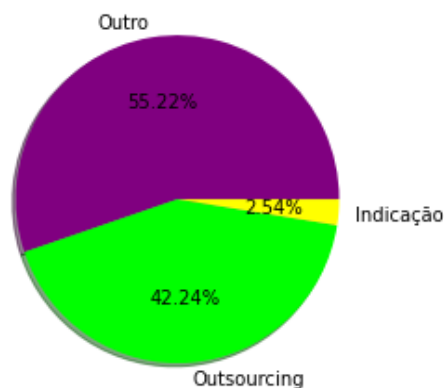


## 6.7 Projeto 3 – Resolvendo o Problema de Negócios

✓ Qual a Proporção dos funcionários por canal de recrutamento?

```
In [38]: import matplotlib.pyplot as plt
import seaborn as sns
dataset['canal_recrutamento'].value_counts()
fatias = [55375, 42358, 2547]
labels = "Outro", "Outsourcing", "Indicação"
colors = ['purple', 'lime', 'yellow']
explode = [0, 0, 0]
plt.pie(fatias, labels = labels, colors = colors, explode = explode, shadow = True, autopct = "%.2f%%")
plt.title('Percentual de Funcionários Por Canal de Recrutamento', fontsize = 15)
plt.axis('off')
#plt.legend()
plt.show()
```

Percentual de Funcionários Por Canal de Recrutamento

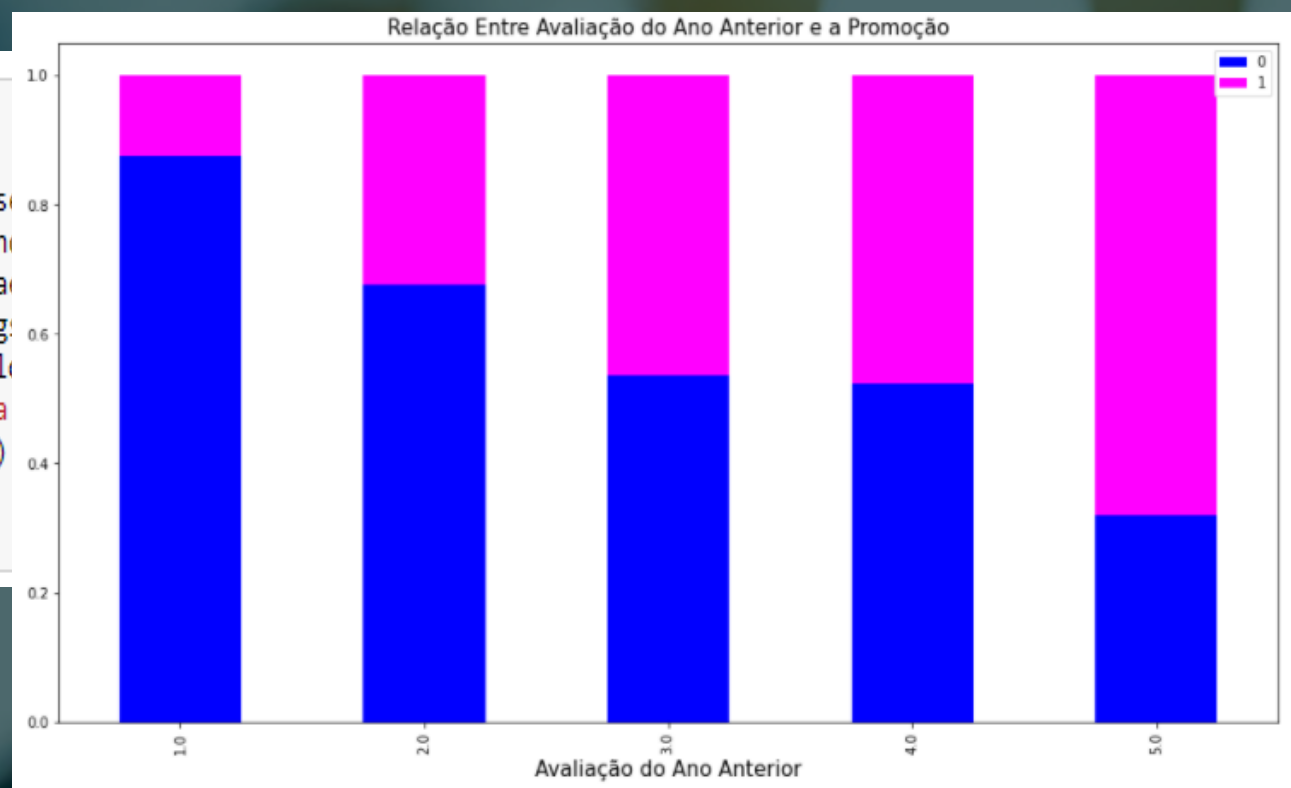


## 6.7 Projeto 3 – Resolvendo o Problema de Negócios

### ✓ Qual a Relação Entre a Promoção e a Avaliação do Ano Anterior?

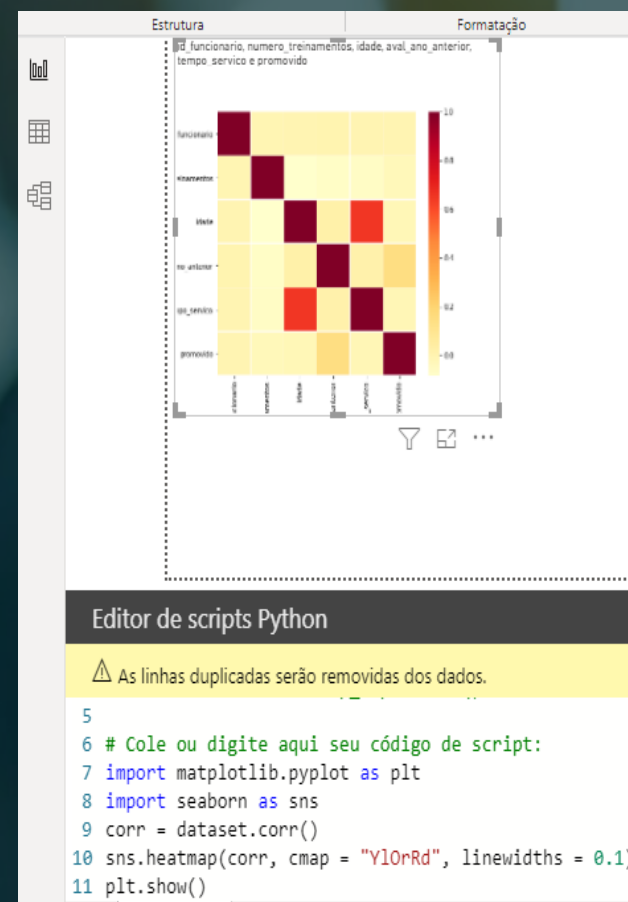
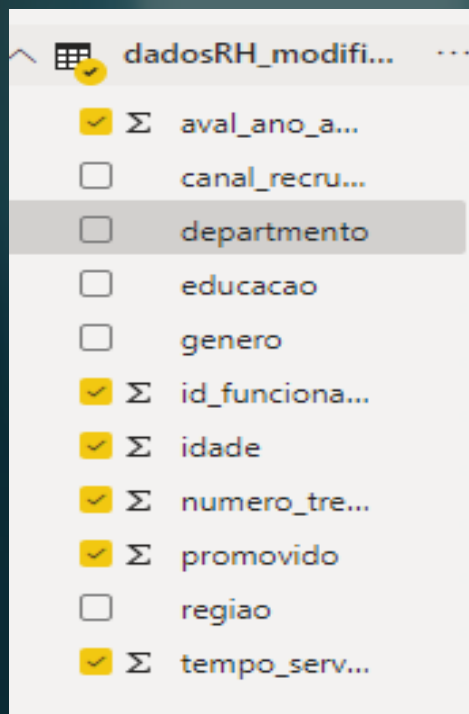
```
In [39]: import matplotlib.pyplot as plt
import seaborn as sns
data = pd.crosstab(dataset['aval_ano_anterior'], data['promoção'],
                    data.div(data.sum(1).astype(float), axis = 0).plot(kind='stacked',
                    figsize=(10, 5))
col = 'Avaliação do Ano Anterior'

plt.title('Relação Entre Avaliação do Ano Anterior e a Promoção')
plt.xlabel('Avaliação do Ano Anterior', fontsize = 15)
plt.legend()
plt.show()
```



## 6.8 Projeto 3 – Construindo o DashBoard analítico

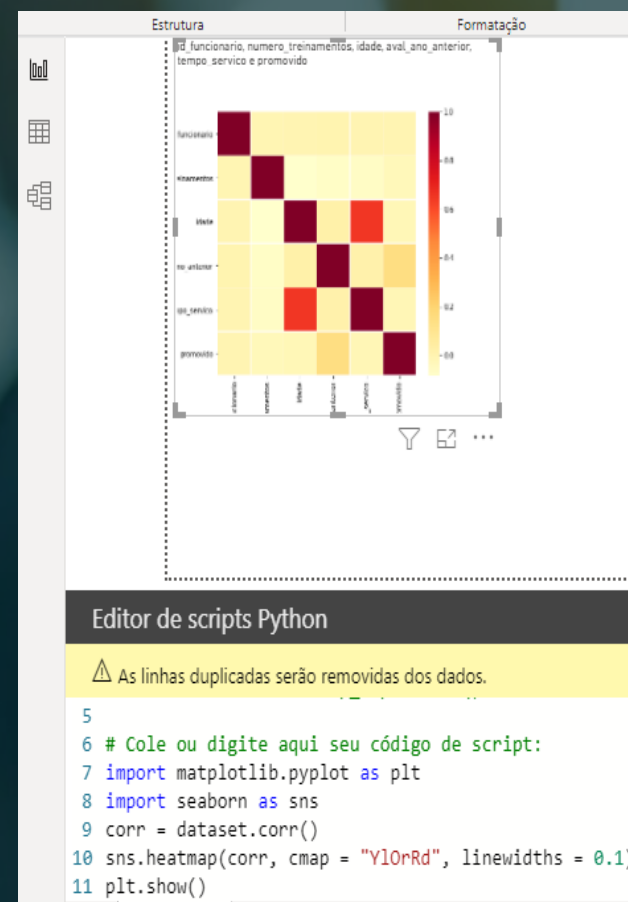
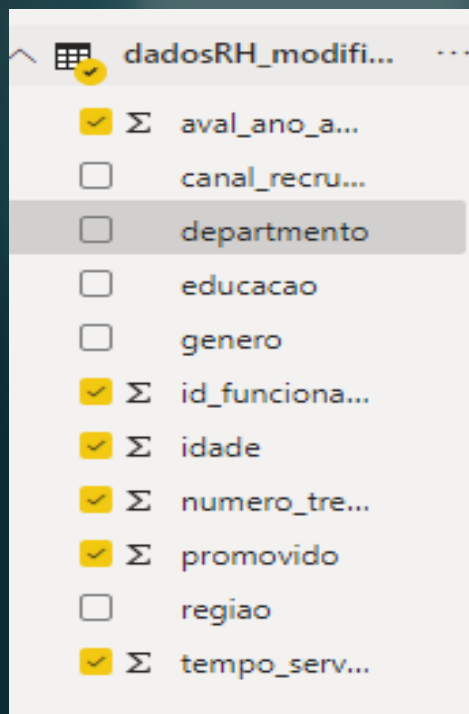
- ✓ Abra um novo projeto e verifique as configurações regionais. E importe o arquivo criado dadosRH\_modificado.csv. Após importar o arquivo selecione o campo de script do python e vamos criar o primeiro gráfico.





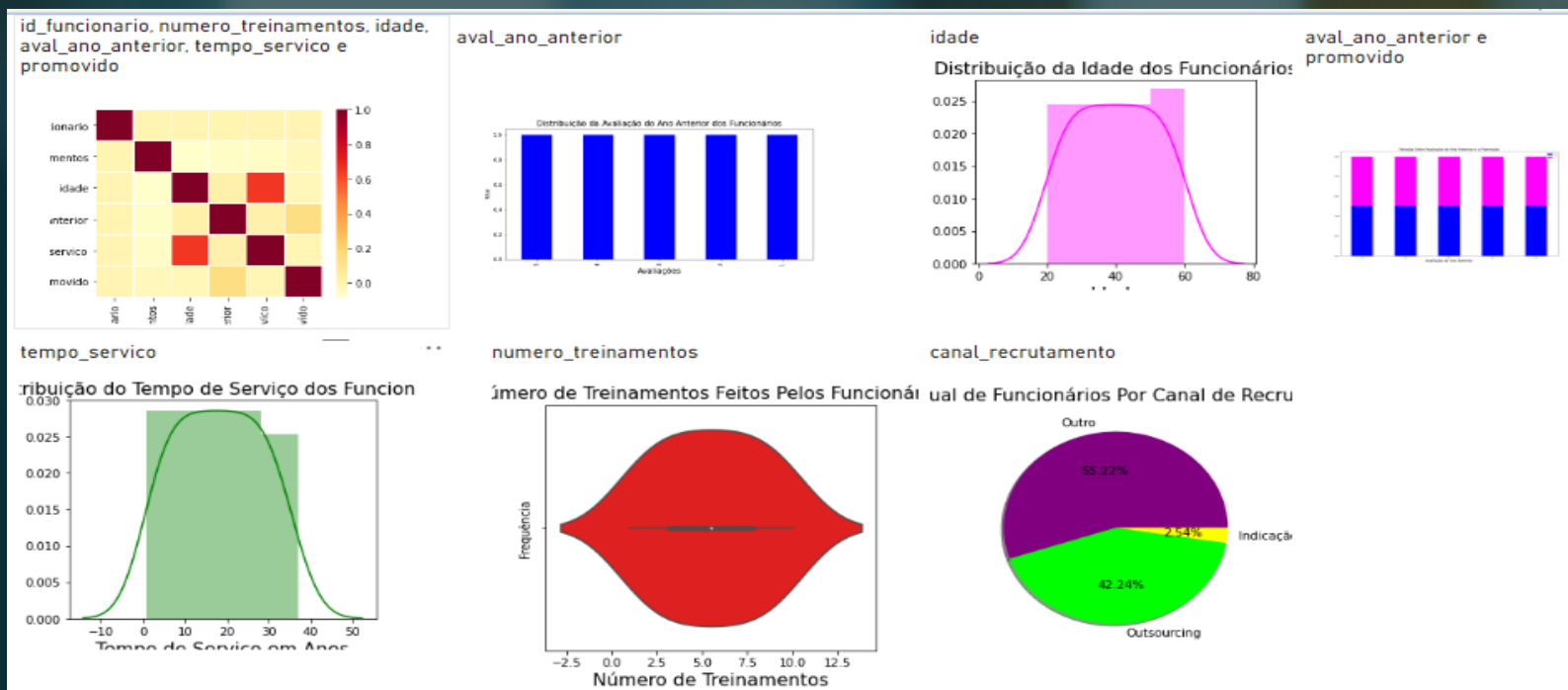
## 6.8 Projeto 3 – Construindo o DashBoard analítico

- ✓ Abra um novo projeto e verifique as configurações regionais. E importe o arquivo criado dadosRH\_modificado.csv. Após importar o arquivo selecione o campo de script do python e vamos criar o primeiro gráfico.



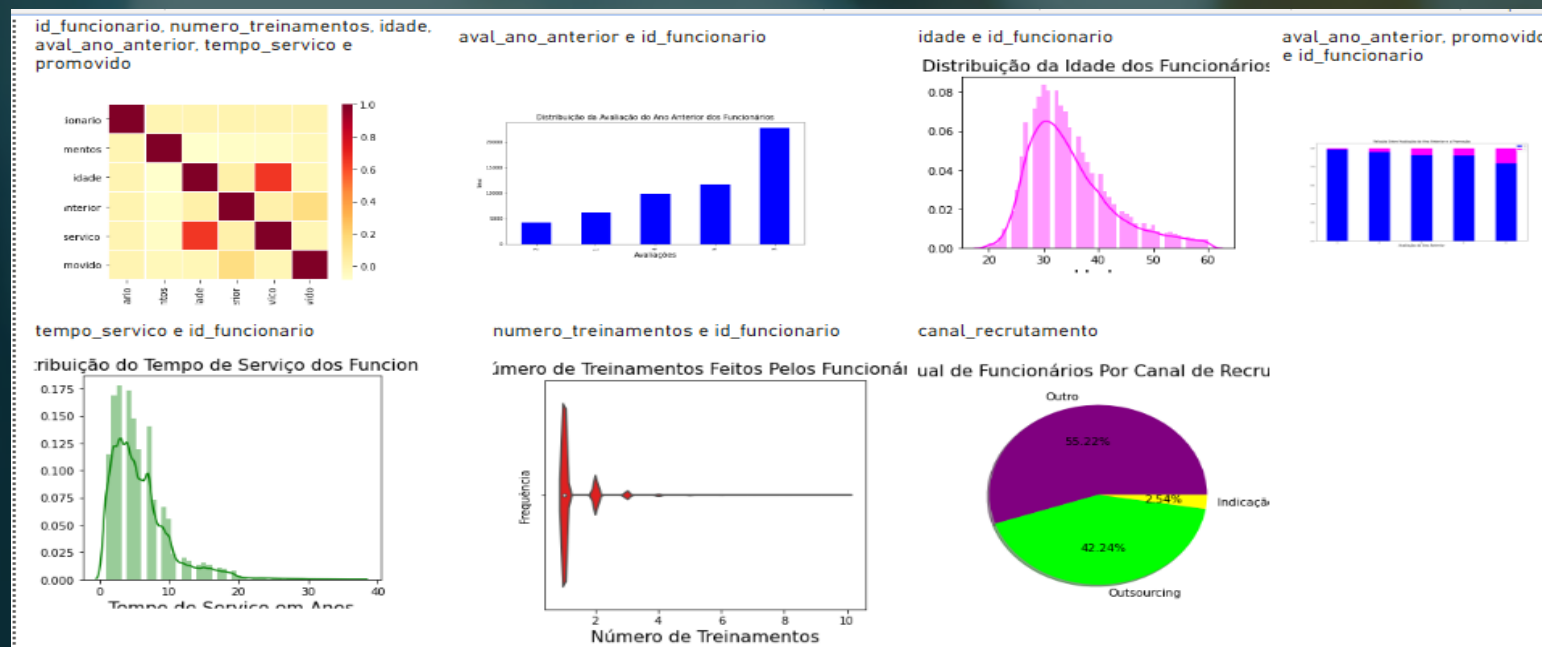
## 6.8 Projeto 3 – Construindo o DashBoard analítico

- ✓ De maneira análoga crie os outros 6 gráficos dentro do Power BI. Perceba que alguns gráficos ficaram diferentes do que foram gerados no Jupyter Notebook. Na realidade eles deveriam estar idênticos, pois é o mesmo código. A causa desse problema é a manipulação de dados feita pelo Power BI.



## 6.8 Projeto 3 – Construindo o DashBoard analítico

- ✓ Quando você utiliza um Script em Python dentro do Power BI, o mesmo cria automaticamente um código para indexação dos valores, ou seja, ele cria um código para identificar cada valor. Então para corrigir devemos dizer ao Power BI, qual coluna ele deve utilizar como index. Então basta selecionar um gráfico de cada vez e selecionar o campo de `id_funcionario` no menu de campos.





# Objetivos do curso

---

- ✓ • *Instalar e configurar a ferramenta Power BI;*
- ✓ • *Diferenciar e compreender conceitos de Business Intelligence, Big Data e Data Science;*
- ✓ • *Compreender e avaliar problemas de maneira analítica, estratégica e fenomenológica;*
- ✓ • *Criar tabelas de relacionamento;*
- ✓ • *Apresentar resultados interativos;*
- ✓ • *Tratar, Transformar e Carregar(ETL – Extract, Transform e Load) dados de forma otimizada;*
- ✓ • *Analisar Séries Temporais;*
- ✓ • *Criar Dashboard práticos e objetivos.*
- ✓ • *Utilizar ferramenta python para criar dashboards mais dinâmicos e otimizar o processamento de dados.*



# OBRIGADO!

---



*Kaio Mesquita*



*kaio@det.ufc.br*



*kaio\_mesquitaa*