

# Um guia replicando 10 Formulas do EXCEL no Python



```
[49] print('Ola Mundão!')
```

Ola Mundão!



⋮	✕	✓	<i>fx</i>	
C	D	E	F	
		10		
10				
...				

# O que vamos replicar

## Replicando Excel no Python

Nesse guia vou utilizar um **script mais simples** para qualquer pessoa poder entender. Até porque a ideia é guiar como **são os conceitos** em ambas 'ferramentas'.

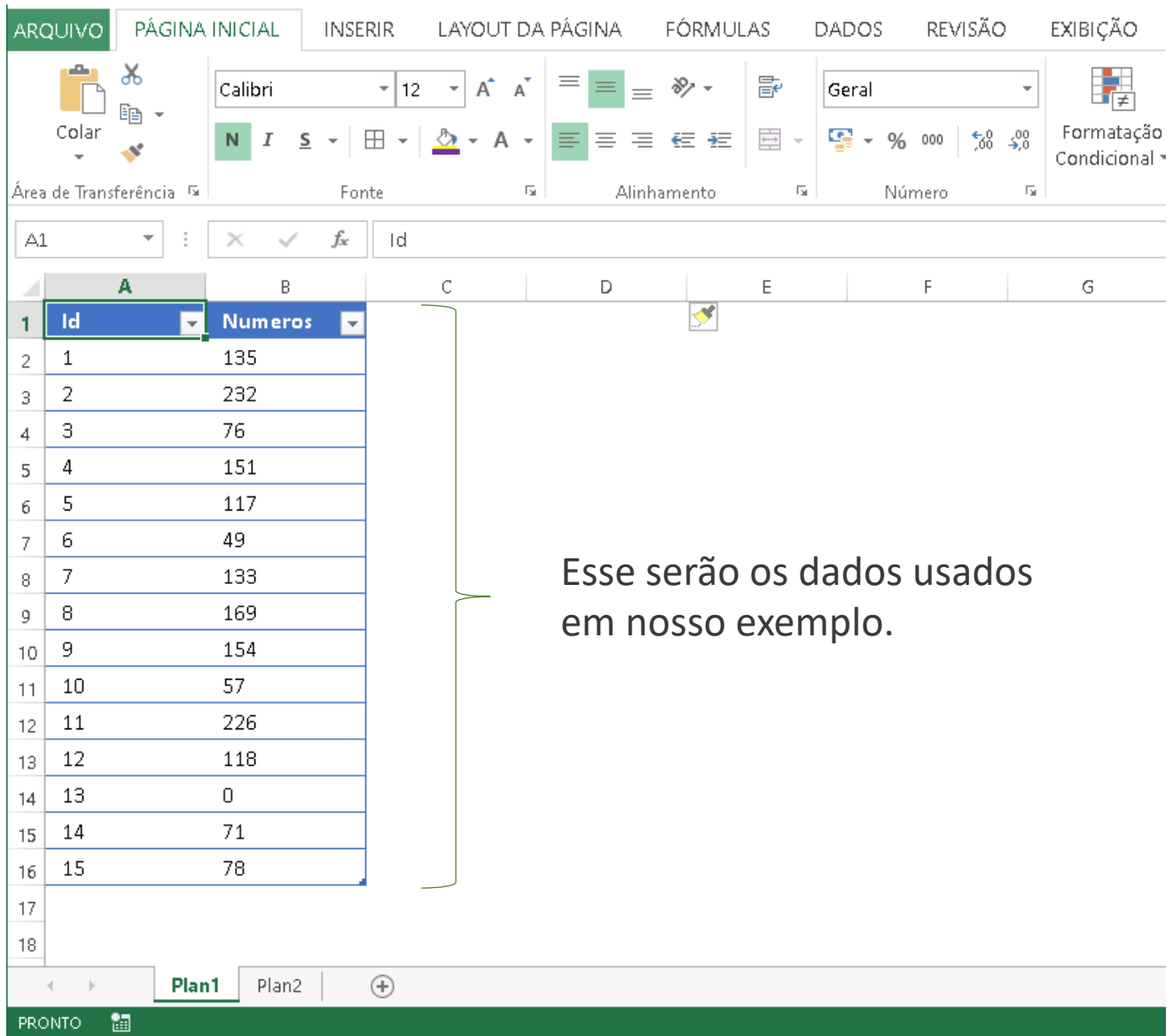
**Não vou utilizar exemplos complexos.**

Nesse guia vamos replicar as seguintes fórmulas do **Excel** no **Python**:

1. Soma
2. Média
3. Arredondar
4. Hoje
5. Cont.SES
6. Procv
7. Concatenar
8. Mínimo
9. Máximo
10. Se

# Mão na Massa

Vamos usar o seguinte conjunto de dados para esse guia.



ARQUIVO PÁGINA INICIAL INSERIR LAYOUT DA PÁGINA FÓRMULAS DADOS REVISÃO EXIBIÇÃO						
Colar						
Área de Transferência						
Fonte						
Alinhamento						
Número						
Formatação Condicional						
A1						
Id						
A B C D E F G						
1	Id	Numeros				
2	1	135				
3	2	232				
4	3	76				
5	4	151				
6	5	117				
7	6	49				
8	7	133				
9	8	169				
10	9	154				
11	10	57				
12	11	226				
13	12	118				
14	13	0				
15	14	71				
16	15	78				
17						
18						
Plan1 Plan2						
PRONTO						

Esse serão os dados usados em nosso exemplo.

Vamos fazer a mesma coisa no Python

1º Vamos importar algumas bibliotecas externas da linguagem. Basicamente estamos “chamando funções extras”.

```
[24] # Biblioteca para modelagem de dados
import pandas as pd

# Biblioteca para recursos de datas
import datetime
```

**Pandas** – é uma biblioteca com diversos recursos para trabalhamos com banco de dados. Podemos fazer toda a modelagem de dados através dele.

**Datetime** – é uma biblioteca com diversos recursos para trabalhamos com datas e horas de diferente tipos ( formato 12 horas, 24 horas, data américa, data britânica e outros ).

2º Vamos criar os mesmos números do Excel

```
[16] # Gerando os numeros igual ao do Excel
      Id = [ 1, 2, 3, 4, 5, 6, 7, 8, 9,
             10, 11, 12, 13, 14, 15 ]
      Valores = [ 135, 232, 76, 151, 117, 49, 133,
                  169, 154, 57, 226, 118, 0 , 71, 78 ]

      # Incluindo esses dados em um dicionario
      Dicionario = {
          'Id':Id,
          'Valores':Valores
      }

      # Incluindo esses dados em uma tabela
      Base = pd.DataFrame( Dicionario )
```

- 1º Criamos 2 listas para incluir todos os valores
- 2º Criamos um dicionário para organizar os dados
- 3º Chamamos a função do ‘Pandas’ para ler o dicionário.

Agora vamos verificar como ficou nossa tabela

```
[17] # Verificando os valores da base
      Base
```

	<b>Id</b>	<b>Valores</b>
<b>0</b>	1	135
<b>1</b>	2	232
<b>2</b>	3	76
<b>3</b>	4	151
<b>4</b>	5	117
<b>5</b>	6	49
<b>6</b>	7	133
<b>7</b>	8	169
<b>8</b>	9	154
<b>9</b>	10	57
<b>10</b>	11	226
<b>11</b>	12	118
<b>12</b>	13	0
<b>13</b>	14	71
<b>14</b>	15	78

Igualzinho ao Excel

# Formula ‘Soma()’

## Como é no Excel

	A	B	C	D	E	F
1	Id	Numeros				
2	1	135		Função	Retorno	Formula
3	2	232		Soma	1766	=SOMA(B2:B16)
4	3	76				
5	4	151				
6	5	117				
7	6	49				
8	7	133				
9	8	169				
10	9	154				
11	10	57				
12	11	226				
13	12	118				
14	13	0				
15	14	71				
16	15	78				

## Como posso fazer no Python

```
[19] # Replicando a função Soma

# Filtro a coluna a especifica e aplico a função [ sum() ]
Base['Valores'].sum()

1766
```

Aqui basicamente selecionamos a coluna ‘Valores’ e chamamos a função ‘sum()’.

Podemos fazer de diversas formas essa soma. Mas essa na minha opinião é mais simples e rápida.

Sum – seria soma em inglês.

# Formula ‘média()’

## Como é no Excel

	A	B	C	D	E	F	G
1	Id	Numeros					
2	1	135		Função	Retorno	Formula	
3	2	232		Média	117,73	=MÉDIA(B2:B16)	
4	3	76					
5	4	151					
6	5	117					
7	6	49					
8	7	133					
9	8	169					
10	9	154					
11	10	57					
12	11	226					
13	12	118					
14	13	0					
15	14	71					
16	15	78					
17							
18							

## Como posso fazer no Python

```
[21] # Replicando a função Média

# Filtro a coluna a especifica e aplico a função [ mean() ]
Base['Valores'].mean()

117.73333333333333
```

Muito similar ao conceito do soma.

Selecionamos a coluna em especifico e chamamos a função ‘mean’.

Mean – seria média em inglês .

# Formula ‘ARRED()’

## Como é no Excel

	A	B	C	D	E	F	G
1	Id	Numeros					
2	1	135		Função	Retorno	Formula	
3	2	232		Arredodar	118	=ARRED(E3;0)	
4	3	76					
5	4	151					
6	5	117					
7	6	49					
8	7	133					
9	8	169					
10	9	154					
11	10	57					
12	11	226					
13	12	118					
14	13	0					
15	14	71					
16	15	78					
17							
18							

## Como posso fazer no Python

```
[23] # Replicando a função arredodar

# Nesse caso chamaos a função [ round ]
# Definimos '0' ou seja, não queremos nenum valor dps da virgula
round( Base['Valores'].mean(), 0 )

118.0
```

Aproveitei o valor da média e arredondei.

Chamei a função ‘round’ para arredondar o valor.

Round – em inglês seria arredondar.

# Formula ‘Hoje()’

## Como é no Excel

	A	B	C	D	E	F	G
1	Id	Numeros					
2	1	135		Função	Retorno	Formula	
3	2	232		Hoje	28/04/2021	=HOJE()	
4	3	76					
5	4	151					
6	5	117					
7	6	49					
8	7	133					
9	8	169					
10	9	154					
11	10	57					
12	11	226					
13	12	118					
14	13	0					
15	14	71					
16	15	78					
17							
18							

## Como posso fazer no Python

```
[28] # Replicando a função hoje

# Chamo a função today da biblioteca datetime
print( datetime.datetime.today() )

2021-04-28 19:16:39.466819
```

Chamei a função ‘Today’ da biblioteca que importamos lá no começo ‘Datetime’.

Usei a função ‘print’ para mostrar o valor.



# Formula ‘Cont.SES()’

## Como é no Excel

	A	B	C	D	E	F	G
1	Id	Numeros					
2	1	135		Função	Retorno	Formula	
3	2	232		Cont.SES	9	=CONT.SE(B2:B16;">"&100)	
4	3	76					
5	4	151					
6	5	117					
7	6	49					
8	7	133					
9	8	169					
10	9	154					
11	10	57					
12	11	226					
13	12	118					
14	13	0					
15	14	71					
16	15	78					
17							
18							

## Como posso fazer no Python

```
[31] # Replicando a função cont.ses
      Base['Valores'].loc[ Base['Valores'] > 100 ].count()
```

9

Aqui usei a função ‘loc’ do Pandas.

A função ‘loc’ tem como objetivo localizar.

Assim localizei todos os registros maiores que 100 e depois apliquei a função ‘count’.

A função ‘count’ tem como objetivo contar os valores.

Count – em inglês seria contar.

# Formula ‘Procv()’

## Como é no Excel

	A	B	C	D	E	F	G
1	<b>Id</b>	<b>Numeros</b>					
2	1	135		Função	Retorno	Formula	
3	2	232		Procv	154	=PROCV(E6;A2:B16;2;0)	
4	3	76					
5	4	151					
6	5	117					
7	6	49					
8	7	133					
9	8	169					
10	9	154					
11	10	57					
12	11	226					
13	12	118					
14	13	0					
15	14	71					
16	15	78					
17							
18							

## Como posso fazer no Python

```
[34] # Replicando a função Procv

Base.loc[ Base['Id'] == 9 ]
```

Id Valores		
8	9	154

Aqui usei a função ‘loc’ do Pandas novamente.

A função ‘loc’ tem como objetivo localizar (apenas relembrando)

Passei para função localizar o ‘Id’ igual a 9.

Assim ele retornou inclusive a linha inteira.

# Formula ‘Concatenar()’

## Como é no Excel

	A	B	C	D	E	F	G
1	<b>Id</b>	<b>Numeros</b>					
2	1	135		Função	Retorno	Formula	
3	2	232		Concatenar	Concatenar	=CONCATENAR(A2;B2)	
4	3	76					
5	4	151					
6	5	117					
7	6	49					
8	7	133					
9	8	169					
10	9	154					
11	10	57					
12	11	226					
13	12	118					
14	13	0					
15	14	71					
16	15	78					
17							
18							

## Como posso fazer no Python

```
[45] # Replicando a função Concatenar
      Id = Base.loc[ Base['Id'] == 1 ]['Id'].values
      Valores = Base.loc[ Base['Id'] == 1 ]['Valores'].values

      print( 'Concatenando:', str(Id[0]) + str(Valores[0]) )
```

Concatenando: 1135

No nosso exemplo do Excel, estamos concatenando o ‘A2’+’B2’.

Formatos de campos em linguagens de programação é pouco ‘chatinho’ para trabalhar.

- 1º Filtrei o ‘id’ 1 de ambas colunas e defini eles em variáveis diferentes;
- 2º Tive que converter o campo para ‘string’ ( string = Texto ). O campo anteriormente estava em formato numero, assim caso tentasse fazer isso sem converter iria gerar um erro;
- 3º Depois que converter para texto o campo, usei o símbolo de adição para juntar ambos.

# Formula ‘Mínimo()’

## Como é no Excel

	A	B	C	D	E	F	G
1	<b>Id</b>	<b>Numeros</b>					
2	1	135		Função	Retorno	Formula	
3	2	232		Minimo	0	=MENOR(B2:B16;1)	
4	3	76					
5	4	151					
6	5	117					
7	6	49					
8	7	133					
9	8	169					
10	9	154					
11	10	57					
12	11	226					
13	12	118					
14	13	0					
15	14	71					
16	15	78					
17							
18							

## Como posso fazer no Python

```
[46] # Replicando a função Mínimo

# Vamos filtrar a coluna e chamar a função [ min() ]
Base['Valores'].min()
```

0

Muito similar a função ‘sum’.

Seleciona a coluna e aplica a função ‘min’

# Formula ‘Maior()’

## Como é no Excel

	A	B	C	D	E	F	G
1	Id	Numeros					
2	1	135		Função	Retorno	Formula	
3	2	232		Maior	232	=MAIOR(B2:B16;1)	
4	3	76					
5	4	151					
6	5	117					
7	6	49					
8	7	133					
9	8	169					
10	9	154					
11	10	57					
12	11	226					
13	12	118					
14	13	0					
15	14	71					
16	15	78					
17							
18							

## Como posso fazer no Python

```
[47] # Replicando a função Maximo

# Vamos filtrar a coluna e chamar a função [ max() ]
Base['Valores'].max()
```

232

Muito similar a função ‘min’.

Seleciona a coluna e aplica a função ‘max’

# Formula ‘SE()’

## Como é no Excel

	A	B	C	D	E	F
1	Id	Numeros				
2	1	135				
3	2	232				
4	3	76				
5	4	151				
6	5	117				
7	6	49				
8	7	133				
9	8	169				
10	9	154				
11	10	57				
12	11	226				
13	12	118				
14	13	0				
15	14	71				
16	15	78				
17						
18						

Função	Retorno	Formula
Se	Ele é maior	=SE(E10<100;"Ele é menor";"Ele é maior")

## Como posso fazer no Python

```
[48] # Replicando a função SE

# Vamos definir em uma variavel o valor máximo
Valor_Maximo = Base['Valores'].max()

# Agora vamos aplicar uma condição [ if ]
# o IF é semelhante ao SE
if Valor_Maximo < 100:
    print('Ele é menor')
else:
    print('Ele é maior')
```

Ele é maior

- 1º Gravei o valor do maior em uma variável.
- 2º Chamei o método ‘if’ que é igual ao ‘se’ do Excel.
- 3º Passei uma condição de ‘Menor’
- 4º Caso não seja menor que 100, vá para a segunda condição, que no caso seria o ‘else’.

## Final

Esse guia é mais uma ilustração de como funcionaria em ambas 'ferramentas'.

Aqui usei um script bem simples com o intuito de demonstração.



**Odemir Depieri Jr**

Software Engineer Sr  
Tech Lead  
Specialization AI