

# EDA

August 9, 2021

## 1 Bibliotecas:

```
[406]: import pandas as pd
import numpy as np
import seaborn as sns

import matplotlib as plt
from matplotlib.pyplot import figure
plt.rcParams["figure.figsize"] = (30,15)

import plotly.graph_objects as go
from plotly.subplots import make_subplots
import plotly.express as px
```

## 2 Dicionário:

- Date -
- Android OS Version -
- Current Device Installs - \* Acumulado de instalações atuais ( independe se cada usuário tem mais de um dispositivo )
- Daily Device Installs -
- Daily Device Uninstalls -
- Daily Device Upgrades -
- Current User Installs -
- Total User Installs -
- Daily User Installs -
- Daily User Uninstalls -
- **Active Device Installs** - É o número de dispositivos ativos em que o app está instalado. Os dispositivos ativos são aqueles que foram ligados pelo menos uma vez nos últimos 30 dias.
- Install events - É o número de instalações, incluindo os dispositivos onde o app já havia sido instalado anteriormente. Isso não inclui pré-instalações nem reativação de dispositivos.

- Update events - É o número de dispositivos em que o app foi atualizado.
- Uninstall events - É o número de vezes que o app foi desinstalado. Isso não inclui dispositivos inativos.

**Fonte** = [https://support.google.com/googleplay/android-developer/answer/139628?ref\\_topic=7071935&\\_ga=2.1814854336.1623787579&\\_gac=1.180609877.1623787579.CjwKCAjwn6GGBhADEiwAruUcKhNyidcpyZoQhlzM7Arelacionadas-a-instala%C3%A7%C3%B5es%2Cfalhas-e-erros-o-app-n%C3%A3o-est%C3%A1-respondendo-anrs%2Cinstant-apps-android](https://support.google.com/googleplay/android-developer/answer/139628?ref_topic=7071935&_ga=2.1814854336.1623787579&_gac=1.180609877.1623787579.CjwKCAjwn6GGBhADEiwAruUcKhNyidcpyZoQhlzM7Arelacionadas-a-instala%C3%A7%C3%B5es%2Cfalhas-e-erros-o-app-n%C3%A3o-est%C3%A1-respondendo-anrs%2Cinstant-apps-android)

```
[407]: print('Instalações ativas atualmente no Android : ',df.loc[df.Date == df.Date.
        ↪max(),'Active Device Installs'].sum())
```

Instalações ativas atualmente no Android : 611475.0

### 3 Analisando - os\_\_version

Agrupados por versão de sistema operacional

```
[408]: df = pd.read_csv('Total_install_os_version_use.csv')
```

```
[409]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28646 entries, 0 to 28645
Data columns (total 15 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Date                                  28646 non-null  object
1   Package Name                         28646 non-null  object
2   Android OS Version                   25298 non-null  object
3   Current Device Installs              3649 non-null   float64
4   Daily Device Installs                28646 non-null  int64
5   Daily Device Uninstalls              28646 non-null  int64
6   Daily Device Upgrades                28646 non-null  int64
7   Current User Installs                3649 non-null   float64
8   Total User Installs                  28646 non-null  int64
9   Daily User Installs                  28646 non-null  int64
10  Daily User Uninstalls                28646 non-null  int64
11  Active Device Installs               25950 non-null  float64
12  Install events                       20835 non-null  float64
13  Update events                       20835 non-null  float64
14  Uninstall events                     20835 non-null  float64
dtypes: float64(6), int64(6), object(3)
memory usage: 3.3+ MB
```

```
[410]: df.Date = df.Date.astype('datetime64')
```

```
[411]: df.drop(columns=['Current Device Installs','Current User Installs','Package_↵
↵Name'],inplace=True)
```

```
[412]: df['Android OS Version'] = df['Android OS Version'].astype('category')
```

### 3.1 Distribuição de os por active device installs

```
[413]: df.loc[df.Date == df.Date.max()].sort_values('Active Device_↵
↵Installs',ascending=False)[['Android OS Version','Active Device Installs']]#_↵
↵Apenas devices mais recentes
```

```
[413]:
```

	Android OS Version	Active Device Installs
28643	Android 10	245632.0
28645	Android 11	126117.0
28642	Android 9	98384.0
28641	Android 8.1	48360.0
28640	Android 8.0	29745.0
28637	Android 6.0	25489.0
28639	Android 7.1	17303.0
28638	Android 7.0	14641.0
28636	Android 5.1	4214.0
28635	Android 5.0	1161.0
28634	Android 4.4	371.0
28631	Android 4.1	34.0
28632	Android 4.2	14.0
28633	Android 4.3	8.0
28644	Android 1.5	2.0
28630	NaN	0.0

```
[414]: df.loc[df.Date == df.Date.max(),'Active Device Installs'].sum() # Soma atual de_↵
↵active devices intalls.
```

```
[414]: 611475.0
```

#### 3.1.1 Visão de Devices atuais

Sobre todos devices dentro do último registro do google

```
[415]: df.Date.max() # último registro
```

```
[415]: Timestamp('2021-07-09 00:00:00')
```

```
[416]: df_os = df.loc[df.Date == df.Date.max()] # df_os contém apenas data mais atual
```

```
[417]: df_os.dropna(inplace=True);
```

```
<ipython-input-417-710e4b55d838>:1: SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
[418]: df_os.set_index('Android OS Version',inplace=True)
```

```
[419]: df_os.sort_values('Active Device Installs',ascending=False,inplace=True)
```

<ipython-input-419-bc3d7ce9d0dd>:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
[420]: df_os['Percentage_installs'] = df_os['Active Device Installs']/df_os['Active_
      ↪Device Installs'].sum()*100;
```

<ipython-input-420-71c513b6824d>:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
[421]: df_os
```

```
[421]:
```

	Date	Daily Device Installs	Daily Device Uninstalls	\
Android OS Version				
Android 10	2021-07-09	2219	0	
Android 11	2021-07-09	1133	0	
Android 9	2021-07-09	678	0	
Android 8.1	2021-07-09	399	0	
Android 8.0	2021-07-09	202	0	
Android 6.0	2021-07-09	255	0	
Android 7.1	2021-07-09	187	0	
Android 7.0	2021-07-09	109	0	
Android 5.1	2021-07-09	63	0	
Android 5.0	2021-07-09	9	0	
Android 4.4	2021-07-09	0	0	
Android 4.1	2021-07-09	0	0	
Android 4.2	2021-07-09	0	0	

Android 4.3	2021-07-09	0	0
Android 1.5	2021-07-09	0	0

	Daily Device Upgrades	Total User Installs \
Android OS Version		
Android 10	0	0
Android 11	0	0
Android 9	0	0
Android 8.1	0	0
Android 8.0	0	0
Android 6.0	0	0
Android 7.1	0	0
Android 7.0	0	0
Android 5.1	0	0
Android 5.0	0	0
Android 4.4	0	0
Android 4.1	0	0
Android 4.2	0	0
Android 4.3	0	0
Android 1.5	0	0

	Daily User Installs	Daily User Uninstalls \
Android OS Version		
Android 10	1708	2539
Android 11	687	1171
Android 9	517	845
Android 8.1	349	504
Android 8.0	160	232
Android 6.0	229	283
Android 7.1	139	177
Android 7.0	93	152
Android 5.1	52	89
Android 5.0	8	9
Android 4.4	0	0
Android 4.1	0	0
Android 4.2	0	0
Android 4.3	0	0
Android 1.5	0	0

	Active Device Installs	Install events	Update events \
Android OS Version			
Android 10	245632.0	3517.0	2172.0
Android 11	126117.0	1878.0	1242.0
Android 9	98384.0	1114.0	849.0
Android 8.1	48360.0	631.0	550.0
Android 8.0	29745.0	329.0	320.0
Android 6.0	25489.0	329.0	273.0

Android 7.1	17303.0	267.0	184.0
Android 7.0	14641.0	180.0	148.0
Android 5.1	4214.0	94.0	50.0
Android 5.0	1161.0	12.0	9.0
Android 4.4	371.0	1.0	0.0
Android 4.1	34.0	0.0	0.0
Android 4.2	14.0	0.0	0.0
Android 4.3	8.0	0.0	0.0
Android 1.5	2.0	0.0	0.0

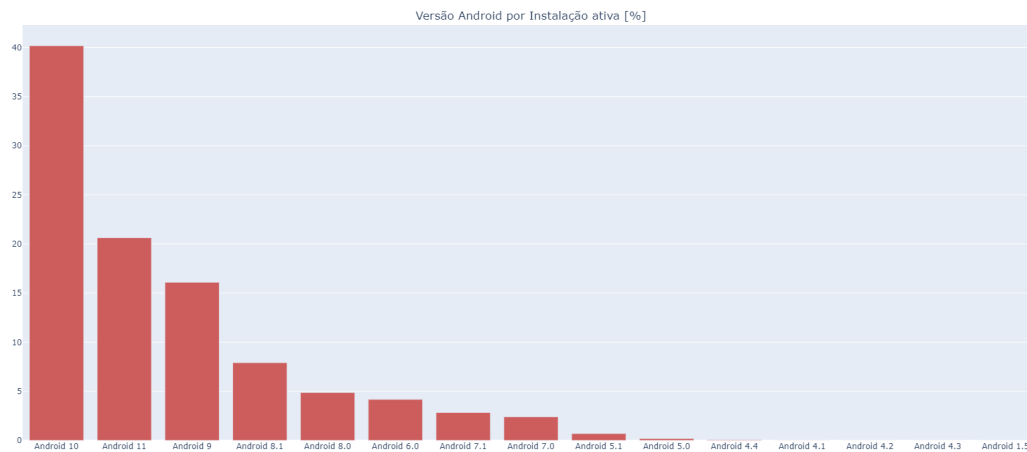
	Uninstall events	Percentage_installs
Android OS Version		
Android 10	2782.0	40.170408
Android 11	1361.0	20.625046
Android 9	938.0	16.089619
Android 8.1	544.0	7.908745
Android 8.0	269.0	4.864467
Android 6.0	307.0	4.168445
Android 7.1	209.0	2.829715
Android 7.0	166.0	2.394374
Android 5.1	97.0	0.689153
Android 5.0	10.0	0.189869
Android 4.4	0.0	0.060673
Android 4.1	0.0	0.005560
Android 4.2	0.0	0.002290
Android 4.3	0.0	0.001308
Android 1.5	0.0	0.000327

```
[422]: fig = go.Figure()

fig.add_trace(go.Bar(
    x=df_os.index,
    y=df_os['Percentage_installs'],
    marker_color='indianred'
))

fig.update_layout(
    title={
        'text': " Versão Android por Instalação ativa [%] ",
        'y':0.9,
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'},
    autosize=False,
    width=800,
    height=800,)
```

```
fig.show()
```



```
[423]: df_os['Active Device Installs']
```

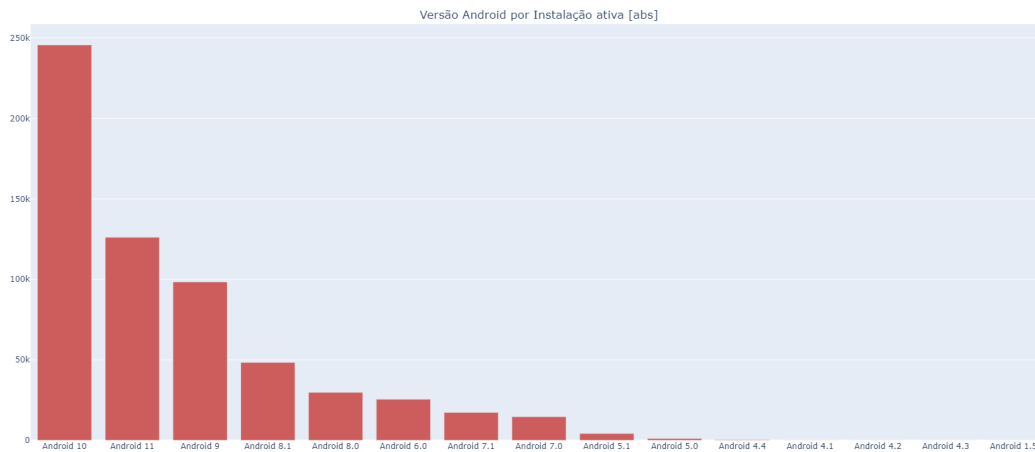
```
[423]: Android OS Version
Android 10      245632.0
Android 11      126117.0
Android 9       98384.0
Android 8.1     48360.0
Android 8.0     29745.0
Android 6.0     25489.0
Android 7.1     17303.0
Android 7.0     14641.0
Android 5.1      4214.0
Android 5.0      1161.0
Android 4.4       371.0
Android 4.1       34.0
Android 4.2       14.0
Android 4.3        8.0
Android 1.5        2.0
Name: Active Device Installs, dtype: float64
```

```
[424]: fig = go.Figure()

fig.add_trace(go.Bar(
    x=df_os.index,
    y=df_os['Active Device Installs'],
    marker_color='indianred'
))
```

```
fig.update_layout(
    title={
        'text': " Versão Android por Instalação ativa [abs] ",
        'y':0.9,
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'},
    autosize=False,
    width=800,
    height=800,)

fig.show()
```



## 4 Analisando - device

Cruzamento dos devices atuais conectados aos devices listados com google

Possíveis perguntas:

- Qual tipo de device temos mais ?
- Quais configurações de tela usamos mais ?
- Qual o tipo de dpi mais utilizado pelos clientes ?

Features Importantes:

- Model Name;
- Form Factor;
- Screen Sizes
- Screen Densities



#### 4.0.1 Datasets

```
[425]: df2 = pd.read_csv('Total_install_app_device_use.csv')
```

#### 4.0.2 Data mais recente

```
[426]: df2.Date.max()
```

```
[426]: '2021-07-09'
```

```
[427]: df2.Date = df2.Date.astype('datetime64')
df2_now = df2.loc[df2.Date == df2.Date.max(),['Device','Active Device_
↳Installs']] # Contém apenas atualizado dos devices 09/07/2021
```

```
[428]: df2_now.sort_values('Active Device Installs',ascending=False,inplace=True)
```

```
[429]: df2_now['Percentage_installs'] = df2_now['Active Device Installs']/
↳df2_now['Active Device Installs'].sum()*100
```

```
[430]: df2_now.groupby('Device').sum().sort_values('Active Device Installs')
```

```
[430]:
```

	Active Device Installs	Percentage_installs
Device		
1001-G	0.0	0.000000
serranolte	0.0	0.000000
A8	0.0	0.000000
BLU_STUDIO_G	0.0	0.000000
gauguinpro	0.0	0.000000
...	...	...
lavender	9882.0	1.616092
a10	10607.0	1.734658
on7xelte	11653.0	1.905720
on5xelte	12692.0	2.075637
ginkgo	17232.0	2.818104

[2580 rows x 2 columns]

#### 4.0.3 Filtro de mínimo 1 device instalados por modelo de device :

```
[431]: df2_now.loc[df2_now['Active Device Installs']>=1] # que tem pelo menos 1
↳devices
```

```
[431]:
```

	Device	Active Device Installs	Percentage_installs
7167168	ginkgo	17232.0	2.818104
7167658	on5xelte	12692.0	2.075637
7167660	on7xelte	11653.0	1.905720
7166846	a10	10607.0	1.734658
7167476	lavender	9882.0	1.616092

...	...	...	...
7166644	Sarah	1.0	0.000164
7166636	S0-04K	1.0	0.000164
7166634	S0-02K	1.0	0.000164
7166633	S0-02J	1.0	0.000164
7166643	SUPER	1.0	0.000164

[2566 rows x 3 columns]

```
[432]: df2_now = df2_now.loc[df2_now['Active Device Installs']>=1]
```

```
[433]: print('Devices com pelo menos 1 instalação :',df2_now.loc[df2_now['Active_
↪Device Installs']>=1].shape[0])
```

Devices com pelo menos 1 instalação : 2566

```
[434]: percentiles = list(np.arange(0,1,0.1)) # Faixas de percentil a serem analisadas
df2_now.describe(percentiles )
```

```
[434]:
```

	Active Device Installs	Percentage_installs
count	2566.000000	2566.000000
mean	238.298909	0.038971
std	1030.919395	0.168596
min	1.000000	0.000164
0%	1.000000	0.000164
10%	1.000000	0.000164
20%	1.000000	0.000164
30%	1.500000	0.000245
40%	2.000000	0.000327
50%	3.000000	0.000491
60%	6.000000	0.000981
70%	12.000000	0.001962
80%	38.000000	0.006214
90%	252.500000	0.041294
max	17232.000000	2.818104

- 25% dos modelos concentram a maior parte de instalações! Sendo que um dos modelos possui 17232 devices

#### 4.1 \*\*\* Discutir sobre nota de corte de devices \*\*\*

#### 4.2 Bricando com corte:

```
[435]: df2_now.loc[df2_now['Active Device Installs']>=50, 'Percentage_installs']
```

```
[435]: 7167168    2.818104
7167658    2.075637
7167660    1.905720
7166846    1.734658
```

```

7167476    1.616092
...
7167144    0.008177
7165935    0.008177
7166864    0.008177
7167898    0.008177
7166228    0.008177
Name: Percentage_installs, Length: 474, dtype: float64

```

```

[436]: df2_now.loc[df2_now['Active Device Installs']>=50,'Percentage_installs'].sum()
↳ # que tem pelo menos 19 device

```

```

[436]: 97.92289136923014

```

### 4.3 Conclusão

- 655 devices de um lista de 655(devices)/2582(total) 25% dos devices totais representam 98,84% das instalações ativas.
- Se pegarmos 283 devices listados acima representamos 95 % da população de instalações

```

[437]: df2_now['Percentage_installs_acc'] = df2_now['Percentage_installs'].cumsum() #
↳ Porcentagem acumulada para observar concentração de Device

```

```

[438]: df2_now.loc[df2_now['Percentage_installs_acc'] <= 95] # Número de device que
↳ detém 98% das instalações ativas

```

```

[438]:
      Device  Active Device Installs  Percentage_installs \
7167168    ginkgo             17232.0             2.818104
7167658  on5xelte             12692.0             2.075637
7167660  on7xelte             11653.0             1.905720
7166846     a10             10607.0             1.734658
7167476  lavender              9882.0             1.616092
...
7167245    gts6l               199.0             0.032544
7166307  OnePlus6T              197.0             0.032217
7167492     lmi               197.0             0.032217
7165556  ASUS_X00R_7            196.0             0.032054
7165507    A7_Pro              190.0             0.031072

      Percentage_installs_acc
7167168             2.818104
7167658             4.893741
7167660             6.799460
7166846             8.534118
7167476            10.150211
...
7167245            94.871745

```

```

7166307          94.903962
7167492          94.936179
7165556          94.968233
7165507          94.999305

```

[283 rows x 4 columns]

```
[439]: df2_now.loc[df2_now['Device']=='lavender']
```

```

[439]:          Device  Active Device  Installs  Percentage_installs  \
7167476  lavender                    9882.0             1.616092

          Percentage_installs_acc
7167476                10.150211

```

## 5 Características dos devices x Instalações:

Nome do device, resolução de tela, dpi e outras características

```
[473]: devices = pd.read_csv('devices.csv')
```

```
[474]: devices.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17098 entries, 0 to 17097
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Manufacturer           17098 non-null  object
1   Model Name             17098 non-null  object
2   Model Code             17098 non-null  object
3   RAM (TotalMem)         17095 non-null  object
4   Form Factor            17097 non-null  object
5   System on Chip         17098 non-null  object
6   Screen Sizes           17098 non-null  object
7   Screen Densities       17098 non-null  object
8   ABIs                   17098 non-null  object
9   Android SDK Versions   17098 non-null  object
10  OpenGL ES Versions     17098 non-null  object
dtypes: object(11)
memory usage: 1.4+ MB

```

### 5.1 Left Join: instalações + devices

```
[464]: devices.loc[devices['Model Code']=='lavender']
```

```
[464]:
```

	Manufacturer	Model Name	Model Code	RAM (TotalMem)	Form Factor	\
11296	Redmi	Redmi Note 7	lavender	2732-3736MB	Phone	
16446	ZTE	Z717VL	lavender	908MB	Phone	

	System on Chip	Screen Sizes	Screen Densities	\
11296	Qualcomm SDM660	1080x2340	440	
16446	Qualcomm MSM8909	480x800	240	

	ABIs	Android SDK Versions	OpenGL ES Versions
11296	arm64-v8a;armeabi;armeabi-v7a	28;29	3.2
16446	armeabi;armeabi-v7a	22	3.0

```
[478]: devices.drop_duplicates(subset='Model Code',inplace=True)
```

```
[479]: df2_now_complete = df2_now.  
↳merge(devices,how='left',left_on='Device',right_on='Model Code')
```

```
[481]: df2_now_complete
```

```
[481]:
```

	Device	Active Device	Installs	Percentage_installs	\
0	ginkgo		17232.0	2.818104	
1	on5xelte		12692.0	2.075637	
2	on7xelte		11653.0	1.905720	
3	a10		10607.0	1.734658	
4	lavender		9882.0	1.616092	
...	...		...	...	
2561	Sarah		1.0	0.000164	
2562	S0-04K		1.0	0.000164	
2563	S0-02K		1.0	0.000164	
2564	S0-02J		1.0	0.000164	
2565	SUPER		1.0	0.000164	

	Percentage_installs_acc	Manufacturer	Model Name	Model Code	\
0	2.818104	Redmi	Redmi Note 8	ginkgo	
1	4.893741	Samsung	Galaxy J5 Prime	on5xelte	
2	6.799460	Samsung	Galaxy J7 Prime	on7xelte	
3	8.534118	Samsung	Galaxy A10	a10	
4	10.150211	Redmi	Redmi Note 7	lavender	
...	...	...	...	...	
2561	99.999346	Sharp	AQUOS sense5G	Sarah	
2562	99.999509	Sony	Xperia XZ2 Premium	S0-04K	
2563	99.999673	Sony	Xperia XZ1 Compact	S0-02K	
2564	99.999836	Sony	Xperia X Compact	S0-02J	
2565	100.000000	NaN	NaN	NaN	

	RAM (TotalMem)	Form Factor	System on Chip	\
0	2634-5626MB	Phone	Qualcomm SM6125	

1	1837-1847MB	Phone	Samsung Exynos 7570
2	2815-2828MB	Phone	Samsung Exynos 7870
3	1720-1724MB	Phone	Samsung Exynos 7884B
4	2732-3736MB	Phone	Qualcomm SDM660
...	...	...	...
2561	3478MB	Phone	Qualcomm SM6350
2562	5678-5682MB	Phone	Qualcomm SDM845
2563	3728-3730MB	Phone	Qualcomm MSM8998
2564	2849MB	Phone	Qualcomm MSM8956
2565	NaN	NaN	NaN

	Screen Sizes	Screen Densities \
0	1080x2340	440
1	720x1280	320
2	1080x1920	480
3	1520x720;720x1339;720x1520	280
4	1080x2340	440
...	...	...
2561	1080x2280	440
2562	1080x1920	420
2563	720x1280	320
2564	720x1280	320
2565	NaN	NaN

	ABIs	Android SDK Versions	OpenGL ES Versions
0	arm64-v8a;armeabi;armeabi-v7a	28;29;30	3.2
1	armeabi;armeabi-v7a	23;24;26	3.1
2	armeabi;armeabi-v7a	23;24;27	3.0;3.2
3	armeabi;armeabi-v7a	28;29;30	3.2
4	arm64-v8a;armeabi;armeabi-v7a	28;29	3.2
...	...	...	...
2561	arm64-v8a;armeabi;armeabi-v7a	30	3.2
2562	arm64-v8a;armeabi;armeabi-v7a	26;28;29	3.2
2563	arm64-v8a;armeabi;armeabi-v7a	26;28	3.2
2564	arm64-v8a;armeabi;armeabi-v7a	23;24;26	3.1;3.2
2565	NaN	NaN	NaN

[2566 rows x 15 columns]

## 5.2 Análise de dpi:

```
[482]: dpi = df2_now_complete[['Screen Densities', 'Active Device Installs']].
        ↳groupby('Screen Densities').sum().sort_values('Active Device_
        ↳Installs', ascending=False)
```

```
[483]: dpi['Percentage_installs'] = dpi['Active Device Installs']/dpi['Active Device_
        ↳Installs'].sum()*100
```

```
[484]: dpi['% Acumulada'] = dpi['Percentage_installs'].cumsum()
```

```
[485]: dpi.head(10)
```

```
[485]:
```

	Active Device Installs	Percentage_installs	% Acumulada
Screen Densities			
320	128446.0	21.083819	21.083819
280	113970.0	18.707650	39.791470
420	110783.0	18.184519	57.975989
480	109739.0	18.013151	75.989140
440	69317.0	11.378066	87.367206
240	25245.0	4.143850	91.511057
440;480	12046.0	1.977295	93.488352
450	6099.0	1.001123	94.489475
420;480	4028.0	0.661178	95.150653
480;640	3752.0	0.615874	95.766526

```
[486]: fig = go.Figure()

# Create figure with secondary y-axis
fig = make_subplots(specs=[[{"secondary_y": True}]])

fig.add_trace(go.Bar(
    x=dpi.index,
    y=dpi['Percentage_installs'],
    marker_color='indianred',
    name = 'Frequência'
),secondary_y=False)

fig.add_trace(go.Scatter(
    x=dpi.index,
    y=dpi['% Acumulada'],
    marker_color='green',
    name = 'Acumulado'
),secondary_y=True)

fig.update_layout(
    title={
        'text': " Distribuição de 'DPI' nos devices ativos ",
        'y':0.9,
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'},
```

```

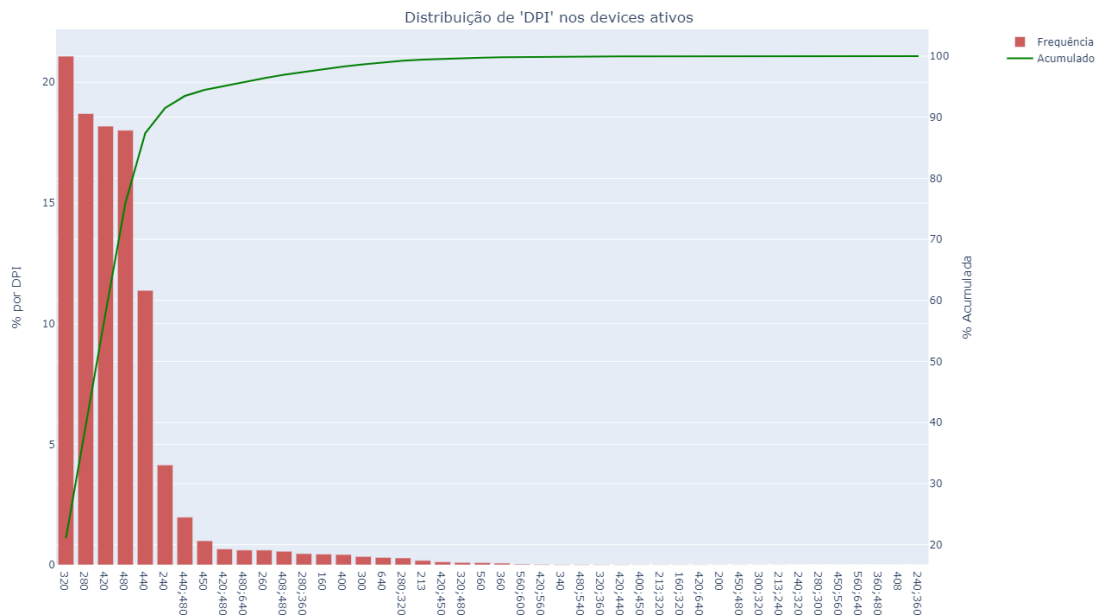
autosize=False,
width=1600,
height=800,)

# Set x-axis title
#fig.update_xaxes(title_text="Possíveis Resoluções")

### Set y-axes titles
fig.update_yaxes(title_text="% por DPI", secondary_y=False)
fig.update_yaxes(title_text="% Acumulada", secondary_y=True)

fig.show()

```



### 5.3 Análise de Resolução de Tela:

```

[487]: screen_size_dist = df2_now_complete[['Screen Sizes','Active Device Installs']].
        ↳groupby('Screen Sizes').sum().sort_values('Active Device_
        ↳Installs',ascending=False)

```

```

[488]: screen_size_dist.head(10)

```



[488]: Active Device Installs

Screen Sizes	
720x1280	62493.0
1080x2340	59052.0
1080x1920	51621.0
1080x2009;1080x2400	32044.0
720x1520	25274.0
1080x2400	24921.0
720x1600	23977.0
1080x2009;1080x2340	19824.0
720x1440	19011.0
1080x2009;1080x2220	17621.0

```
[489]: screen_size_dist['Percentage_installs'] = screen_size_dist['Active Device_Installs']/screen_size_dist['Active Device Installs'].sum()*100
```

```
[490]: screen_size_dist['% Acumulada'] = screen_size_dist['Percentage_installs'].cumsum()
```

```
[491]: fig = go.Figure()

# Create figure with secondary y-axis
fig = make_subplots(specs=[[{"secondary_y": True}]])

fig.add_trace(go.Bar(
    x=screen_size_dist.index,
    y=screen_size_dist['Percentage_installs'],
    marker_color='indianred',
    name = 'Frequência'
),secondary_y=False)

fig.add_trace(go.Scatter(
    x=screen_size_dist.index,
    y=screen_size_dist['% Acumulada'],
    marker_color='green',
    name = 'Acumulado'
),secondary_y=True)

fig.update_layout(
    title={
        'text': " Distribuição de 'resolução de tela' nos devices ativos ",
        'y':0.9,
        'x':0.5,
```

```

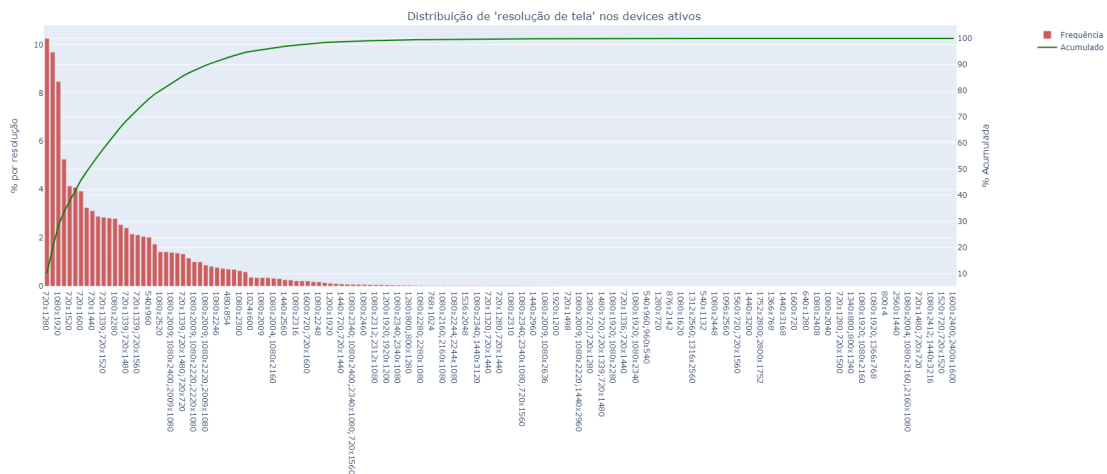
        'xanchor': 'center',
        'yanchor': 'top'},
    autosize=False,
    width=1600,
    height=800,)

# Set x-axis title
#fig.update_xaxes(title_text="Possíveis Resoluções")

### Set y-axes titles
fig.update_yaxes(title_text="% por resolução", secondary_y=False)
fig.update_yaxes(title_text="% Acumulada", secondary_y=True)

fig.show()

```



## 5.4 Análise de Marcas:

```

[492]: brands = df2_now_complete[['Active Device Installs', 'Manufacturer']].
        ↳groupby('Manufacturer').sum().sort_values('Active Device_
        ↳Installs', ascending=False)

```

```

[493]: brands.reset_index(inplace=True)

```

```

[494]: criar_peso_categoria(brands, 'Manufacturer')

```

Check de peso total : 0.9999999999999999

```

[495]: brands.set_index('Manufacturer', inplace=True)

```

```
[496]: brands['% Acumulada'] = brands['Manufacturer_weight'].cumsum()
```

```
[497]: fig = go.Figure()

# Create figure with secondary y-axis
fig = make_subplots(specs=[[{"secondary_y": True}]])

fig.add_trace(go.Bar(
    x=brands.index,
    y=brands['Manufacturer_weight'],
    marker_color='indianred',
    name = 'Frequência'
),secondary_y=False)

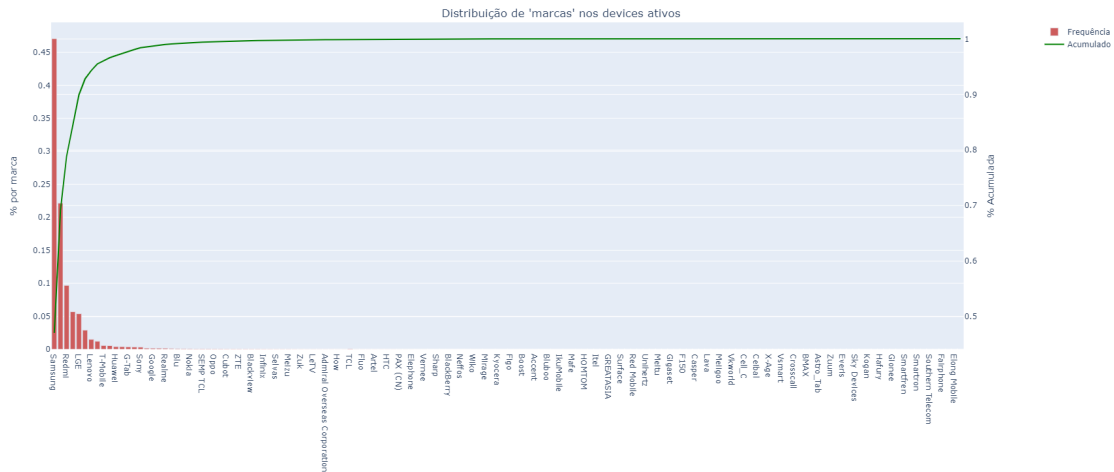
fig.add_trace(go.Scatter(
    x=brands.index,
    y=brands['% Acumulada'],
    marker_color='green',
    name = 'Acumulado'
),secondary_y=True)

fig.update_layout(
    title={
        'text': " Distribuição de 'marcas' nos devices ativos ",
        'y':0.9,
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'},
    autosize=False,
    width=1600,
    height=800,)

# Set x-axis title
#fig.update_xaxes(title_text="Possíveis Resoluções")

### Set y-axes titles
fig.update_yaxes(title_text="% por marca", secondary_y=False)
fig.update_yaxes(title_text="% Acumulada", secondary_y=True)

fig.show()
```



## 5.5 Análise de Aparelhos:

```
[498]: df2_now_complete.loc[df2_now_complete['Device']=='lavender']
```

```
[498]:      Device  Active Device Installs  Percentage_installs \
4  lavender                9882.0          1.616092

      Percentage_installs_acc Manufacturer      Model Name Model Code \
4                10.150211          Redmi  Redmi  Note 7  lavender

      RAM (TotalMem) Form Factor  System on Chip Screen Sizes Screen Densities \
4  2732-3736MB      Phone  Qualcomm SDM660   1080x2340          440

                                ABIs Android SDK Versions OpenGL ES Versions
4  arm64-v8a;armeabi;armeabi-v7a                28;29          3.2
```

```
[521]: df2_now_complete[['Active Device Installs','Model Name','Model Code']].
      ↳groupby(['Model Code','Model Name']).sum().sort_values('Active Device_
      ↳Installs',ascending=False)
```

```
[521]:      Active Device Installs
Model Code  Model Name
ginkgo      Redmi Note 8          17232.0
on5xelte     Galaxy J5 Prime       12692.0
on7xelte     Galaxy J7 Prime       11653.0
a10          Galaxy A10           10607.0
lavender     Redmi  Note 7          9882.0
...
HWCUN-U6582 Y5II                      1.0
```

```
HWCUN-L6735 Y5II 1.0
HWCPN-Q HUAWEI MediaPad M3 Lite 1.0
a7y18lteks Galaxy A7 (2018) 1.0
Studio_XL_2 Studio_XL_2 1.0
```

[2130 rows x 1 columns]

```
[512]: modelo = df2_now_complete[['Active Device Installs', 'Model Name', 'Model Code']].
        ↳groupby('Model Name').sum().sort_values('Active Device_
        ↳Installs', ascending=False)
```

```
# Existem devices com nomes iguais e especificações diferente, por isso existe_
↳o Model Name e Model Code(Individual)
```

```
[513]: modelo['%relativa'] = modelo['Active Device Installs'] / modelo['Active Device_
        ↳Installs'].sum()*100
```

```
[514]: modelo['% Acumulada'] = modelo['%relativa'].cumsum()
```

```
[515]: modelo
```

```
[515]:
```

	Active Device Installs	%relativa	% Acumulada
Model Name			
Redmi Note 8	17250.0	2.831508	2.831508
Galaxy J5 Prime	12692.0	2.083333	4.914841
Galaxy J7 Prime	11674.0	1.916233	6.831075
Galaxy A10	10607.0	1.741090	8.572165
Redmi Note 7	9882.0	1.622085	10.194250
...	...	...	...
HUAWEI MediaPad M3 Lite	1.0	0.000164	99.999343
C8	1.0	0.000164	99.999508
Xperia C5 Ultra Dual	1.0	0.000164	99.999672
Xperia Ace	1.0	0.000164	99.999836
PHANTOM X	1.0	0.000164	100.000000

[1706 rows x 3 columns]

```
[516]: fig = go.Figure()

# Create figure with secondary y-axis
fig = make_subplots(specs=[[{"secondary_y": True}]])

fig.add_trace(go.Bar(
    x=modelo.index,
    y=modelo['%relativa'],
```

```

        marker_color='indianred',
        name = 'Frequência'
    ),secondary_y=False)

fig.add_trace(go.Scatter(
    x=modelo.index,
    y=modelo['% Acumulada'],
    marker_color='green',
    name = 'Acumulado'
),secondary_y=True)

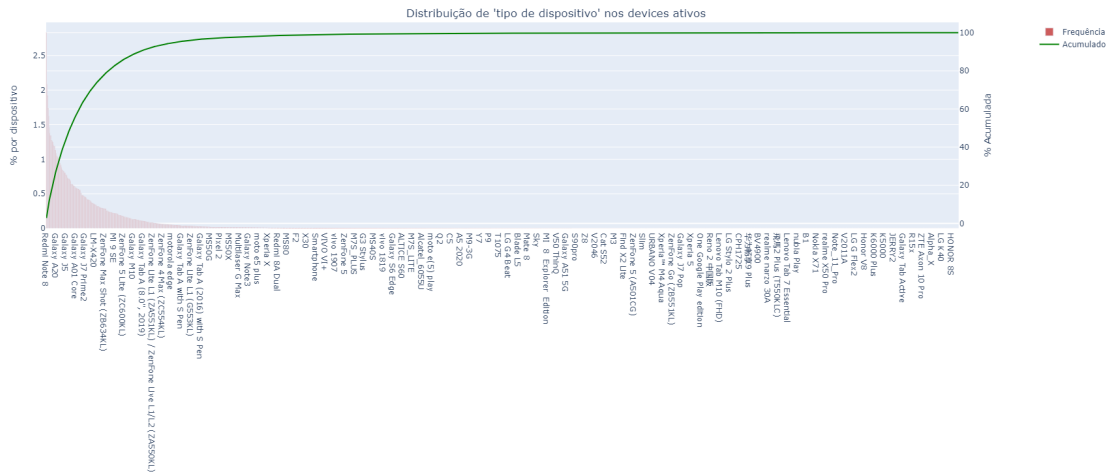
fig.update_layout(
    title={
        'text': " Distribuição de 'tipo de dispositivo' nos devices ativos ",
        'y':0.9,
        'x':0.5,
        'xanchor': 'center',
        'yanchor': 'top'},
    autosize=False,
    width=1600,
    height=800,)

# Set x-axis title
fig.update_xaxes(title_text="Possíveis Resoluções")

### Set y-axes titles
fig.update_yaxes(title_text="% por dispositivo", secondary_y=False)
fig.update_yaxes(title_text="% Acumulada", secondary_y=True)

fig.show()

```



## 6 Grupos de Análise

### 6.1 Visualização agrupada dos Dados:

#### 6.1.1 Marcas -> Aparelhos -> Resolução

```
[522]: df2_now_complete
```

```
[522]:
```

	Device	Active Device	Installs	Percentage_installs	\
0	ginkgo		17232.0	2.818104	
1	on5xelte		12692.0	2.075637	
2	on7xelte		11653.0	1.905720	
3	a10		10607.0	1.734658	
4	lavender		9882.0	1.616092	
...	...		...	...	
2561	Sarah		1.0	0.000164	
2562	S0-04K		1.0	0.000164	
2563	S0-02K		1.0	0.000164	
2564	S0-02J		1.0	0.000164	
2565	SUPER		1.0	0.000164	

	Percentage_installs_acc	Manufacturer	Model Name	Model Code	\
0	2.818104	Redmi	Redmi Note 8	ginkgo	
1	4.893741	Samsung	Galaxy J5 Prime	on5xelte	
2	6.799460	Samsung	Galaxy J7 Prime	on7xelte	
3	8.534118	Samsung	Galaxy A10	a10	
4	10.150211	Redmi	Redmi Note 7	lavender	
...	...	...	...	...	
2561	99.999346	Sharp	AQUOS sense5G	Sarah	

2562	99.999509	Sony	Xperia XZ2 Premium	S0-04K
2563	99.999673	Sony	Xperia XZ1 Compact	S0-02K
2564	99.999836	Sony	Xperia X Compact	S0-02J
2565	100.000000	NaN	NaN	NaN

	RAM (TotalMem)	Form Factor	System on Chip \
0	2634-5626MB	Phone	Qualcomm SM6125
1	1837-1847MB	Phone	Samsung Exynos 7570
2	2815-2828MB	Phone	Samsung Exynos 7870
3	1720-1724MB	Phone	Samsung Exynos 7884B
4	2732-3736MB	Phone	Qualcomm SDM660
...	...	...	...
2561	3478MB	Phone	Qualcomm SM6350
2562	5678-5682MB	Phone	Qualcomm SDM845
2563	3728-3730MB	Phone	Qualcomm MSM8998
2564	2849MB	Phone	Qualcomm MSM8956
2565	NaN	NaN	NaN

	Screen Sizes	Screen Densities \
0	1080x2340	440
1	720x1280	320
2	1080x1920	480
3	1520x720;720x1339;720x1520	280
4	1080x2340	440
...	...	...
2561	1080x2280	440
2562	1080x1920	420
2563	720x1280	320
2564	720x1280	320
2565	NaN	NaN

	ABIs	Android SDK Versions	OpenGL ES Versions
0	arm64-v8a;armeabi;armeabi-v7a	28;29;30	3.2
1	armeabi;armeabi-v7a	23;24;26	3.1
2	armeabi;armeabi-v7a	23;24;27	3.0;3.2
3	armeabi;armeabi-v7a	28;29;30	3.2
4	arm64-v8a;armeabi;armeabi-v7a	28;29	3.2
...	...	...	...
2561	arm64-v8a;armeabi;armeabi-v7a	30	3.2
2562	arm64-v8a;armeabi;armeabi-v7a	26;28;29	3.2
2563	arm64-v8a;armeabi;armeabi-v7a	26;28	3.2
2564	arm64-v8a;armeabi;armeabi-v7a	23;24;26	3.1;3.2
2565	NaN	NaN	NaN

[2566 rows x 15 columns]



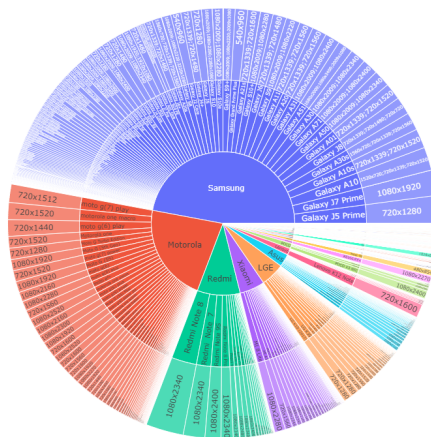
```
[523]: ##### Ajuda o time de Android a olhar possíveis dpis e telas
```

```
fig = px.sunburst(df2_now_complete[['Screen Sizes','Screen Densities','Active_
↳Device Installs']].dropna(), path=['Screen Densities','Screen Sizes'],
                  values='Active Device Installs',height = 800)

fig.write_html("resolucao_dpi.html")
```

```
[526]: fig = px.sunburst(df2_now_complete[['Screen Sizes','Manufacturer','Model_
↳Name','Active Device Installs']].dropna(), path=['Manufacturer','Model_
↳Name','Screen Sizes'],
                  values='Active Device Installs',height = 800)

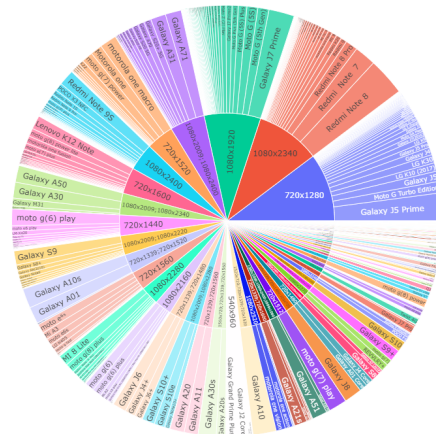
fig.show()
```



### 6.1.2 Resolução - > Aparelhos

```
[527]: fig = px.sunburst(df2_now_complete[['Screen Densities','Screen_
↳Sizes','Manufacturer','Model Name','Active Device Installs']].dropna(),
↳path=['Screen Sizes','Model Name'],
                  values='Active Device Installs',height = 800)

fig.show()
```



[ ]:

### 6.1.3 Dpi - > Aparelhos

```
[141]: fig = px.sunburst(df2_now_complete[['Screen Densities','Screen_
    ↳ Sizes','Manufacturer','Model Name','Active Device Installs']].dropna(),
    ↳ path=['Screen Densities','Model Name'],
        values='Active Device Installs',height = 800)
```

## 7 Modelo Proposto para Indicação dos Celulares a buscar

Criação de Pesos para features importantes:

Manufacturer / Model Name / Form Factor / Screen Sizes / Screen Densities

Manufacturer / Model Name Weight / Form Factor Weight / Screen Sizes Weight / Screen Densities

```
[405]: def criar_peso_categoria(dataframe,nome_coluna):

    nome_coluna = str(nome_coluna)

    # Calcula primeiramente qual o peso dado o total de devices instalados:
    ↳ Cria df_temp

    df_temp = dataframe[['Active Device Installs',f'{nome_coluna}']].
    ↳ groupby(f'{nome_coluna}').sum().sort_values('Active Device
    ↳ Installs',ascending=False)
    df_temp[['weight']] = df_temp[['Active Device Installs']] /
    ↳ df_temp[['Active Device Installs']].sum()
```

```

print('Check de peso total :',df_temp['weight'].sum())

# Criando o nove nome da coluna com pesos

dataframe[f'{nome_coluna}_weight'] = dataframe[f'{nome_coluna}'].map(lambda
→x: df_temp.filter(like=x, axis=0)['weight'][0])

#print(dataframe)

```

```
[528]: df2_weights = df2_now_complete.copy()
```

```
[529]: df2_weights
```

```
[529]:
```

	Device	Active Device	Installs	Percentage_installs	\
0	ginkgo		17232.0	2.818104	
1	on5xelte		12692.0	2.075637	
2	on7xelte		11653.0	1.905720	
3	a10		10607.0	1.734658	
4	lavender		9882.0	1.616092	
...	...		...	...	
2561	Sarah		1.0	0.000164	
2562	S0-04K		1.0	0.000164	
2563	S0-02K		1.0	0.000164	
2564	S0-02J		1.0	0.000164	
2565	SUPER		1.0	0.000164	

	Percentage_installs_acc	Manufacturer	Model Name	Model Code	\
0	2.818104	Redmi	Redmi Note 8	ginkgo	
1	4.893741	Samsung	Galaxy J5 Prime	on5xelte	
2	6.799460	Samsung	Galaxy J7 Prime	on7xelte	
3	8.534118	Samsung	Galaxy A10	a10	
4	10.150211	Redmi	Redmi Note 7	lavender	
...	...	...	...	...	
2561	99.999346	Sharp	AQUOS sense5G	Sarah	
2562	99.999509	Sony	Xperia XZ2 Premium	S0-04K	
2563	99.999673	Sony	Xperia XZ1 Compact	S0-02K	
2564	99.999836	Sony	Xperia X Compact	S0-02J	
2565	100.000000	NaN	NaN	NaN	

	RAM (TotalMem)	Form Factor	System on Chip	\
0	2634-5626MB	Phone	Qualcomm SM6125	
1	1837-1847MB	Phone	Samsung Exynos 7570	
2	2815-2828MB	Phone	Samsung Exynos 7870	

3	1720-1724MB	Phone	Samsung Exynos 7884B
4	2732-3736MB	Phone	Qualcomm SDM660
...	...	...	...
2561	3478MB	Phone	Qualcomm SM6350
2562	5678-5682MB	Phone	Qualcomm SDM845
2563	3728-3730MB	Phone	Qualcomm MSM8998
2564	2849MB	Phone	Qualcomm MSM8956
2565	NaN	NaN	NaN

	Screen Sizes	Screen Densities	\
0	1080x2340	440	
1	720x1280	320	
2	1080x1920	480	
3	1520x720;720x1339;720x1520	280	
4	1080x2340	440	
...	...	...	
2561	1080x2280	440	
2562	1080x1920	420	
2563	720x1280	320	
2564	720x1280	320	
2565	NaN	NaN	

	ABIs	Android SDK Versions	OpenGL ES Versions
0	arm64-v8a;armeabi;armeabi-v7a	28;29;30	3.2
1	armeabi;armeabi-v7a	23;24;26	3.1
2	armeabi;armeabi-v7a	23;24;27	3.0;3.2
3	armeabi;armeabi-v7a	28;29;30	3.2
4	arm64-v8a;armeabi;armeabi-v7a	28;29	3.2
...	...	...	...
2561	arm64-v8a;armeabi;armeabi-v7a	30	3.2
2562	arm64-v8a;armeabi;armeabi-v7a	26;28;29	3.2
2563	arm64-v8a;armeabi;armeabi-v7a	26;28	3.2
2564	arm64-v8a;armeabi;armeabi-v7a	23;24;26	3.1;3.2
2565	NaN	NaN	NaN

[2566 rows x 15 columns]

```
[530]: df2_weights.dropna(inplace=True)
```

```
[542]: lista = ['Model Code', 'Manufacturer', 'Model Name', 'Form Factor', 'Screen_
↪Sizes', 'Screen Densities']

for i in lista:

    criar_peso_categoria(df2_weights,i)
```

Check de peso total : 1.0

Check de peso total : 0.9999999999999999

```

Check de peso total : 1.0
Check de peso total : 1.0
Check de peso total : 0.9999999999999999
Check de peso total : 1.0

```

```
[543]: df2_weights.columns
```

```
[543]: Index(['Device', 'Active Device Installs', 'Percentage_installs',
        'Percentage_installs_acc', 'Manufacturer', 'Model Name', 'Model Code',
        'RAM (TotalMem)', 'Form Factor', 'System on Chip', 'Screen Sizes',
        'Screen Densities', 'ABIs', 'Android SDK Versions',
        'OpenGL ES Versions', 'Manufacturer_weight', 'Model Name_weight',
        'Form Factor_weight', 'Screen Sizes_weight', 'Screen Densities_weight',
        'Result', 'Model Code_weight'],
        dtype='object')
```

```
[544]: # Calculo global de pesos !
```

```
''' Talvez não faça sentido pois deve ser olhado de forma individual ou talvez
    ↳scaling dos dados por coluna e somar pra fazer sentido
```

```
Podemos olhar principalmente pra Screen sizes weight e Screen Densities
    ↳weight'''
```

```
df2_weights['Result'] = df2_weights['Manufacturer_weight'] +df2_weights['Model_
    ↳Name_weight']+df2_weights['Screen Sizes_weight']+df2_weights['Screen_
    ↳Densities_weight'] #+df2_weights['Form Factor_weight']
```

```
df2_weights['Result'] = df2_weights['Manufacturer_weight'] *df2_weights['Model_
    ↳Name_weight']*df2_weights['Screen Sizes_weight']*df2_weights['Screen_
    ↳Densities_weight']*100 #*df2_weights['Form Factor_weight']
```

```
[548]: df2_weights.sort_values('Model Name_weight',ascending=False).head(10)
```

```
[548]:
```

	Device	Active Device Installs	Percentage_installs	\
0	ginkgo	17232.0	2.818104	
2032	Note_8	1.0	0.000164	
665	biloba	18.0	0.002944	
35	j5lte	4702.0	0.768960	
1923	j5nlte	1.0	0.000164	
1262	J5	3.0	0.000491	
1	on5xelte	12692.0	2.075637	
779	j53g	12.0	0.001962	
97	j7elte	1846.0	0.301893	
565	j7e3g	30.0	0.004906	

	Percentage_installs_acc	Manufacturer	Model Name	Model Code	\
0	2.818104	Redmi	Redmi Note 8	ginkgo	
2032	99.912834	Ulefone	Note 8	Note_8	
665	98.880576	Redmi	Redmi Note 8	biloba	
35	42.648023	Samsung	Galaxy J5	j5lte	
1923	99.895008	Samsung	Galaxy J5	j5nlte	
1262	99.668997	Cubot	J5	J5	
1	4.893741	Samsung	Galaxy J5 Prime	on5xelte	
779	99.147308	Samsung	Galaxy J5	j53g	
97	73.302424	Samsung	Galaxy J7	j7elte	
565	98.501165	Samsung	Galaxy J7	j7e3g	

	RAM (TotalMem)	Form Factor	System on Chip	...	\
0	2634-5626MB	Phone	Qualcomm SM6125	...	
2032	1943MB	Phone	Mediatek MT6580	...	
665	3736MB	Phone	Mediatek MT6769	...	
35	1383-1388MB	Phone	Qualcomm MSM8916	...	
1923	1383-1388MB	Phone	Qualcomm MSM8916	...	
1262	1959MB	Phone	Mediatek MT6580	...	
1	1837-1847MB	Phone	Samsung Exynos 7570	...	
779	1390-1395MB	Phone	Qualcomm MSM8216	...	
97	1370-1373MB	Phone	Samsung Exynos 7580	...	
565	1370-1372MB	Phone	Samsung Exynos 7580	...	

	ABIs	Android SDK Versions	OpenGL ES Versions	\
0	arm64-v8a;armeabi;armeabi-v7a	28;29;30	3.2	
2032	armeabi;armeabi-v7a	29	2.0	
665	arm64-v8a;armeabi;armeabi-v7a	30	3.2	
35	armeabi;armeabi-v7a	22;23	3.0	
1923	armeabi;armeabi-v7a	22;23	3.0	
1262	armeabi;armeabi-v7a	28	2.0	
1	armeabi;armeabi-v7a	23;24;26	3.1	
779	armeabi;armeabi-v7a	22;23	3.0	
97	armeabi;armeabi-v7a	22;23	3.0;3.1	
565	armeabi;armeabi-v7a	22;23	3.0;3.1	

	Manufacturer_weight	Model Name_weight	Form Factor_weight	\
0	0.096737	0.028315	0.988464	
2032	0.000269	0.028315	0.988464	
665	0.096737	0.028315	0.988464	
35	0.470654	0.020833	0.988464	
1923	0.470654	0.020833	0.988464	
1262	0.000312	0.020833	0.988464	
1	0.470654	0.020833	0.988464	
779	0.470654	0.020833	0.988464	
97	0.470654	0.019162	0.988464	

565	0.470654	0.019162	0.988464
-----	----------	----------	----------

	Screen Sizes_weight	Screen Densities_weight	Result \
0	0.096931	0.113781	3.020955e-03
2032	0.000007	0.041439	2.073878e-10
665	0.096931	0.180132	4.782615e-03
35	0.102579	0.210838	2.120655e-02
1923	0.102579	0.210838	2.120655e-02
1262	0.006937	0.041439	1.867684e-07
1	0.102579	0.210838	2.120655e-02
779	0.102579	0.210838	2.120655e-02
97	0.102579	0.210838	1.950561e-02
565	0.102579	0.210838	1.950561e-02

	Model Code_weight
0	0.028286
2032	0.000002
665	0.000030
35	0.007718
1923	0.000002
1262	0.000005
1	0.020833
779	0.000020
97	0.003030
565	0.000049

[10 rows x 22 columns]

```
[549]: df2_weights.to_csv('results_device_weights.csv',index=False)
```

Podemos ponderar os pesos a fim de destacar o que é mais relevante. Temos problema com device ? Temos problema com dpi ? ou com Resolução ? Caso contrário tudo tem o mesmo peso!

## 7.1 Exemplo de utilização:

```
[567]: #Considerando o peso do Modelo + 2*(Tamanho de tela e 2*Screen Size Weight) - dados como mais criticos
```

```
df2_weights['Result'] = (df2_weights['Model Code_weight'] +
    +2*df2_weights['Screen Densities_weight'] + 2*df2_weights['Screen
    Sizes_weight'])*100
```

```
[562]: pd.set_option('display.max_columns',30)
```

```
[579]: df2_weights.sort_values(by=['Model Code_weight','Screen
    Densities_weight','Screen Sizes_weight'],ascending=False).
    sort_values('Result',ascending=False).head(1000)
```

[579]:

	Device	Active Device	Installs	Percentage_installs	\
1	on5xelte		12692.0	2.075637	
28	merlin		5219.0	0.853510	
35	j5lte		4702.0	0.768960	
46	mlv5		3862.0	0.631588	
1354	cv3		3.0	0.000491	
...	...		...	...	
2519	Tokyo_Lite_4G		1.0	0.000164	
102	mdh15lm		1787.0	0.292244	
152	meh15lm		932.0	0.152418	
179	malta		644.0	0.105319	
191	capri		562.0	0.091909	

	Percentage_installs_acc	Manufacturer	Model Name	\
1	4.893741	Samsung	Galaxy J5 Prime	
28	37.052700	Motorola	Moto G Turbo Edition	
35	42.648023	Samsung	Galaxy J5	
46	50.258800	LGE	LG K10 (2017)	
1354	99.714134	LGE	LG Premier Pro	
...	...	...	...	
2519	99.992477	T-Mobile	REVL 4	
102	74.783106	LGE	LG-K41S	
152	85.621162	LGE	LM-K420	
179	89.025389	Motorola	Lenovo K12	
191	90.200908	Lenovo	Lenovo K13 Note	

	Model Code	RAM (TotalMem)	Form Factor	System on Chip	\
1	on5xelte	1837-1847MB	Phone	Samsung Exynos 7570	
28	merlin	1914MB	Phone	Qualcomm MSM8939	
35	j5lte	1383-1388MB	Phone	Qualcomm MSM8916	
46	mlv5	1859-1867MB	Phone	Mediatek MT6755	
1354	cv3	1863-1871MB	Phone	Qualcomm MSM8917	
...	...	...	...	...	
2519	Tokyo_Lite_4G	1826MB	Phone	Mediatek MT6761	
102	mdh15lm	2839-2841MB	Phone	Mediatek MT6762	
152	meh15lm	2843MB	Phone	Mediatek MT6762	
179	malta	1781-3774MB	Phone	Mediatek MT6762	
191	capri	3726MB	Phone	Qualcomm SM4350	

	Screen Sizes	Screen Densities	ABIs	\
1	720x1280	320	armeabi;armeabi-v7a	
28	720x1280	320	armeabi;armeabi-v7a	
35	720x1280	320	armeabi;armeabi-v7a	
46	720x1280	320	armeabi;armeabi-v7a	
1354	720x1280	320	armeabi;armeabi-v7a	
...	...	...	...	
2519	720x1520	280	armeabi;armeabi-v7a	



102	720x1600	280	armeabi;armeabi-v7a
152	720x1600	280	armeabi;armeabi-v7a
179	720x1600	280	arm64-v8a;armeabi;armeabi-v7a
191	720x1600	280	arm64-v8a;armeabi;armeabi-v7a

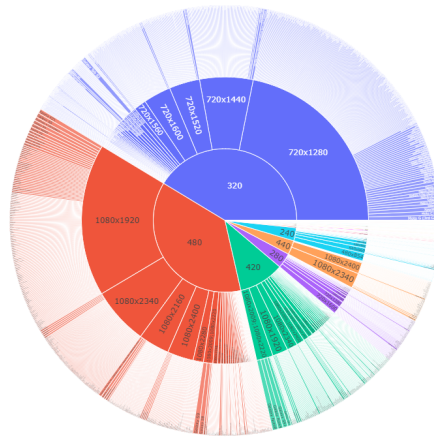
	Android SDK Versions	OpenGL ES Versions	Manufacturer_weight \
1	23;24;26	3.1	0.470654
28	22;23	3.0;3.1	0.221376
35	22;23	3.0	0.470654
46	24;27	3.2	0.053754
1354	25;27;28	3.0	0.053754
...	...	...	...
2519	29	3.2	0.005665
102	28;29	3.2	0.053754
152	29	3.2	0.053754
179	29	3.2	0.221376
191	30	3.2	0.014922

	Model Name_weight	Form Factor_weight	Screen Sizes_weight \
1	0.020833	0.988464	0.102579
28	0.008567	0.988464	0.102579
35	0.020833	0.988464	0.102579
46	0.006339	0.988464	0.102579
1354	0.000005	0.988464	0.102579
...	...	...	...
2519	0.000002	0.988464	0.041486
102	0.002933	0.988464	0.039357
152	0.001530	0.988464	0.039357
179	0.010310	0.988464	0.039357
191	0.000922	0.988464	0.039357

	Screen Densities_weight	Result	Model Code_weight
1	0.210838	64.766848	0.020833
28	0.210838	63.540189	0.008567
35	0.210838	63.455326	0.007718
46	0.210838	63.317444	0.006339
1354	0.210838	63.314818	0.006313
...	...	...	...
2519	0.187077	45.712686	0.000002
102	0.187077	45.580057	0.002933
152	0.187077	45.439713	0.001530
179	0.187077	45.392439	0.001057
191	0.187077	45.388992	0.001023

[1000 rows x 22 columns]

```
[577]: fig = px.sunburst(df2_weights.sort_values(by=['Model Code_weight','Screen_
↳Densities_weight','Screen Sizes_weight'],ascending=False).
↳sort_values('Result',ascending=False), path=['Screen Densities','Screen_
↳Sizes','Model Name'],
values='Result',height = 800)
fig.show()
```



```
[555]: df2_weights.sort_values(by=['Result'],ascending=False).head(10)
```

```
[555]:
```

	Device	Active Device	Installs	Percentage_installs	\
1	on5xelte		12692.0	2.075637	
28	merlin		5219.0	0.853510	
35	j5lte		4702.0	0.768960	
46	mlv5		3862.0	0.631588	
1354	cv3		3.0	0.000491	
47	mcv3		3846.0	0.628971	
64	j5y17lte		2931.0	0.479333	
74	cv109		2521.0	0.412282	
943	cv1		7.0	0.001145	
84	j7xelte		2244.0	0.366981	

	Percentage_installs_acc	Manufacturer	Model Name	Model Code	\
1	4.893741	Samsung	Galaxy J5 Prime	on5xelte	
28	37.052700	Motorola	Moto G Turbo Edition	merlin	
35	42.648023	Samsung	Galaxy J5	j5lte	
46	50.258800	LGE	LG K10 (2017)	mlv5	
1354	99.714134	LGE	LG Premier Pro	cv3	
47	50.887771	LGE	LG K30	mcv3	
64	60.597408	Samsung	Galaxy J5 Pro	j5y17lte	

74	65.054172	LGE	LG K9	cv109
943	99.399812	LGE	LG it	cv1
84	68.994153	Samsung	Galaxy J7(2016)	j7xelte

	RAM (TotalMem)	Form Factor	System on Chip	...	\
1	1837-1847MB	Phone	Samsung Exynos 7570	...	
28	1914MB	Phone	Qualcomm MSM8939	...	
35	1383-1388MB	Phone	Qualcomm MSM8916	...	
46	1859-1867MB	Phone	Mediatek MT6755	...	
1354	1863-1871MB	Phone	Qualcomm MSM8917	...	
47	1846-2879MB	Phone	Mediatek MT6750	...	
64	1816-1819MB	Phone	Samsung Exynos 7870	...	
74	1907MB	Phone	Qualcomm MSM8909	...	
943	1873-2892MB	Phone	Qualcomm MSM8917	...	
84	1820-1822MB	Phone	Samsung Exynos 7870	...	

	ABIs	Android SDK Versions	OpenGL ES Versions	\
1	armeabi;armeabi-v7a	23;24;26	3.1	
28	armeabi;armeabi-v7a	22;23	3.0;3.1	
35	armeabi;armeabi-v7a	22;23	3.0	
46	armeabi;armeabi-v7a	24;27	3.2	
1354	armeabi;armeabi-v7a	25;27;28	3.0	
47	armeabi;armeabi-v7a	25;27	3.2	
64	armeabi;armeabi-v7a	24;27;28	3.2	
74	armeabi;armeabi-v7a	25	3.0	
943	armeabi;armeabi-v7a	25;27;28	3.0	
84	armeabi;armeabi-v7a	23;24;27	3.0;3.2	

	Manufacturer_weight	Model Name_weight	Form Factor_weight	\
1	0.470654	0.020833	0.988464	
28	0.221376	0.008567	0.988464	
35	0.470654	0.020833	0.988464	
46	0.053754	0.006339	0.988464	
1354	0.053754	0.000005	0.988464	
47	0.053754	0.006313	0.988464	
64	0.470654	0.004813	0.988464	
74	0.053754	0.004138	0.988464	
943	0.053754	0.000011	0.988464	
84	0.470654	0.003690	0.988464	

	Screen Sizes_weight	Screen Densities_weight	Result	\
1	0.102579	0.210838	0.334251	
28	0.102579	0.210838	0.321984	
35	0.102579	0.210838	0.321136	
46	0.102579	0.210838	0.319757	
1354	0.102579	0.210838	0.319731	
47	0.102579	0.210838	0.319731	

64	0.102579	0.210838	0.318229
74	0.102579	0.210838	0.317556
943	0.102579	0.210838	0.317556
84	0.102579	0.210838	0.317101

	Model Code_weight
1	0.020833
28	0.008567
35	0.007718
46	0.006339
1354	0.006313
47	0.006313
64	0.004811
74	0.004138
943	0.004138
84	0.003683

[10 rows x 22 columns]

[ ]:

```
[541]: df2_weights.loc[df2_weights['Model Name'].isin(['Samsung S7','Moto G5s_
↳Plus','Galaxy A01 Core','Galaxy J5','Galaxy A5'])]
```

```
[541]:      Device  Active Device  Installs  Percentage_installs  \
35      j5lte                    4702.0             0.768960
53     a01core                   3587.0             0.586614
250     a5lte                    271.0             0.044319
779     j53g                     12.0             0.001962
1200    a53g                      4.0             0.000654
1923    j5nlte                    1.0             0.000164
2262    a5ulte                    1.0             0.000164
```

	Percentage_installs_acc	Manufacturer	Model Name	Model Code	\
35	42.648023	Samsung	Galaxy J5	j5lte	
53	54.523079	Samsung	Galaxy A01 Core	a01core	
250	93.815937	Samsung	Galaxy A5	a5lte	
779	99.147308	Samsung	Galaxy J5	j53g	
1200	99.629421	Samsung	Galaxy A5	a53g	
1923	99.895008	Samsung	Galaxy J5	j5nlte	
2262	99.950448	Samsung	Galaxy A5	a5ulte	

	RAM (TotalMem)	Form Factor	System on Chip	... Screen Densities	\
35	1383-1388MB	Phone	Qualcomm MSM8916	...	320
53	888-1905MB	Phone	Mediatek MT6739WW	...	320
250	1891-1898MB	Phone	Qualcomm MSM8916	...	320
779	1390-1395MB	Phone	Qualcomm MSM8216	...	320

1200	1891-1898MB	Phone	Qualcomm MSM8216	...	320
1923	1383-1388MB	Phone	Qualcomm MSM8916	...	320
2262	1891-1898MB	Phone	Qualcomm MSM8916	...	320

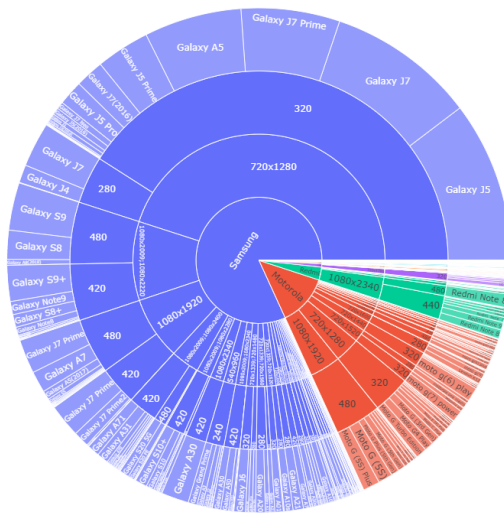
	ABIs	Android SDK Versions	OpenGL ES Versions	\
35	armeabi;armeabi-v7a	22;23	3.0	
53	armeabi;armeabi-v7a	29;30	3.2	
250	armeabi;armeabi-v7a	19;21;23	3.0	
779	armeabi;armeabi-v7a	22;23	3.0	
1200	armeabi;armeabi-v7a	19;21;23	3.0	
1923	armeabi;armeabi-v7a	22;23	3.0	
2262	armeabi;armeabi-v7a	19;21;23	3.0	

	Manufacturer_weight	Model Name_weight	Form Factor_weight	\
35	0.470654	0.020833	0.988464	
53	0.470654	0.005888	0.988464	
250	0.470654	0.012608	0.988464	
779	0.470654	0.020833	0.988464	
1200	0.470654	0.012608	0.988464	
1923	0.470654	0.020833	0.988464	
2262	0.470654	0.012608	0.988464	

	Screen Sizes_weight	Screen Densities_weight	Result
35	0.102579	0.210838	0.021207
53	0.024128	0.210838	0.001410
250	0.102579	0.210838	0.012834
779	0.102579	0.210838	0.021207
1200	0.102579	0.210838	0.012834
1923	0.102579	0.210838	0.021207
2262	0.102579	0.210838	0.012834

[7 rows x 21 columns]

```
[162]: fig = px.sunburst(df2_weights[['Screen Sizes','Manufacturer','Model_
↳Name','Screen Densities','Active Device Installs','Result','Model_
↳Name_weight']]).dropna(), path=['Manufacturer','Screen Sizes','Screen_
↳Densities','Model Name'],
values='Result',height = 800)
fig.show()
```



```
[ ]: fig = px.sunburst(df2_weights[['Screen Sizes','Manufacturer','Model_
↳Name','Screen Densities','Active Device Installs','Result','Model_
↳Name_weight']].dropna(), path=['Manufacturer','Screen Sizes','Screen_
↳Densities','Model Name'],
                        values='Model Name_weight',height = 800)
```

## 8 Fraquezas por Device ? Quais são os devices ou tipo que mais tem crash ?

Analisar crash\_device / crash\_ios / crash\_app\_version

```
[ ]: device
```