


In [1]:

```
1 from selenium import webdriver
2 from selenium.webdriver.common.keys import Keys
3 from selenium.webdriver.common.by import By
4 from selenium.webdriver.support.ui import WebDriverWait
5 from selenium.webdriver.support import expected_conditions as EC
6 from webdriver_manager.chrome import ChromeDriverManager
7
8 import pandas as pd
9 import time
10 import re
```

In [2]:

```
1 driver = webdriver.Chrome(executable_path='./chromedriver.exe')
2
3 driver.get("https://chrome.google.com/webstore/detail/adblock-%E2%80%94-best-ad-blocker
4
```



## Busca Genérica

Resulta um modelo de cada linha de celular. Exemplo: Motorola Edge Pro s

In [4]:

```

1 columns = ['modelo', 'marca', 'lançamento', 'memoria_ram', 'memoria_interna',
2           'tamanho_tela', 'resolucao_tela', 'resolucao_tela_direita', 'resolucao_tela
3           'resolucao_camera_principal', 'resolucao_camera_principal_direita', 'resoluca
4           'resolucao_camera_frontal', 'resolucao_camera_frontal_direita', 'resolucao_ca
5           'capacidade_bateria']
6
7 df = pd.DataFrame(columns=columns) # Criação do data frame, onde irá armazenar todos o
8
9 # Entra no site do adblock para ligarmos o bloqueador de anuncio.
10 # Quando estiver no site do adblock, precisamos fazer a instalação clicando no 'Usar n
11 # apos isso aceitar o pop-up 'Adicionar extensão' e em 30s automaticamente entraremos
12
13 driver = webdriver.Chrome(executable_path='./chromedriver.exe')
14
15 driver.get("https://chrome.google.com/webstore/detail/adblock-%E2%80%94best-ad-blocke
16
17 time.sleep(30)
18
19 driver.get("https://www.maiscelular.com.br/fichas-tecnicas/?aparelho=1&z=0")
20 cookies = driver.find_element_by_xpath('//*[ @id="frmMain"] /div[3] /div /div /a ')
21 cookies.click()
22 time.sleep(3)
23 escopo = driver.find_element_by_xpath('/html/body/form/main/div[2]/div/div[2]/div[4]/s
24 escopo = escopo.text
25 escopo = escopo.split()
26 pg_atual = escopo[5]
27 qtd_pg_total = int(escopo[7])
28 qtd_cell = int(escopo[1])
29 cont = 0
30
31 while (qtd_pg_total >= cont):
32     for seq in range(1, qtd_cell+1):
33
34         try:
35             modelo = WebDriverWait(driver, 5).until(EC.presence_of_element_located((By
36             modelo = modelo.text
37
38             nav = driver.find_element_by_xpath('//*[ @id="accessibility_content"] /div[2
39             nav.click()
40         except:
41             try:
42                 spam = driver.find_element_by_xpath('//*[ @id="dismiss-button"] /div/svg
43                 spam.click()
44                 time.sleep(3)
45             except:
46                 driver.back()
47
48
49             modelo = driver.find_element_by_xpath('//*[ @id="accessibility_content"] /di
50             modelo = modelo.text
51
52             nav = driver.find_element_by_xpath('//*[ @id="accessibility_content"] /div[2
53             nav.click()
54
55         time.sleep(3)
56
57     try:
58         marca = WebDriverWait(driver, 5).until(EC.presence_of_element_located((By.
59         marca = marca.text

```

```
60     except:
61         marca = "brand"
62
63     try:
64         lancamento = WebDriverWait(driver, 5).until(EC.presence_of_element_located
65         lancamento = lancamento.text
66     except:
67         lancamento = "Não Informado"
68
69     try:
70         vendido = WebDriverWait(driver, 5).until(EC.presence_of_element_located((B
71         vendido = vendido.text
72
73     except:
74         vendido = "Não Informado"
75
76     try:
77         memoria_ram = WebDriverWait(driver, 5).until(EC.presence_of_element_locate
78         memoria_ram = memoria_ram.text
79     except:
80         memoria_ram = "Valor não informado"
81         memoria_ram = "Valor não informado"
82
83     try:
84         memoria_interna = WebDriverWait(driver, 5).until(EC.presence_of_element_lo
85         memoria_interna = memoria_interna.text
86     except:
87         memoria_interna = "Valor não informado"
88         memoria_interna = "Valor não informado"
89
90     try:
91         tamanho_tela = WebDriverWait(driver, 5).until(EC.presence_of_element_locat
92         tamanho_tela = tamanho_tela.text
93     except:
94         tamanho_tela = "Valor não informado"
95         tamanho_tela = "Valor não informado"
96
97     try:
98         resolucao_tela = WebDriverWait(driver, 5).until(EC.presence_of_element_loc
99         resolucao_tela = resolucao_tela.text
100         resolucao_tela_direita = resolucao_tela.split(" ")[0].replace("x", " ").sp
101         resolucao_tela_esquerda = resolucao_tela.split(" ")[0].replace("x", " ").s
102     except:
103         resolucao_tela = "Valor não informado"
104         resolucao_tela = "Valor não informado"
105         resolucao_tela_direita = "Valor não informado"
106         resolucao_tela_esquerda = "Valor não informado"
107
108     try:
109         resolucao_camera_principal = WebDriverWait(driver, 10).until(EC.presence_o
110         resolucao_camera_principal = resolucao_camera_principal.text
111         resolucao_camera_principal_direita = resolucao_camera_principal.split(" ")
112         resolucao_camera_principal_esquerda = resolucao_camera_principal.split("
113     except:
114         resolucao_camera_principal = "Valor não informado"
115         resolucao_camera_principal = "Valor não informado"
116         resolucao_camera_principal_direita = "Valor não informado"
117         resolucao_camera_principal_esquerda = "Valor não informado"
118
119
120
```

```
121     try:
122         resolucao_camera_frontal = WebDriverWait(driver, 5).until(EC.presence_of_e
123         resolucao_camera_frontal = resolucao_camera_frontal.text
124         resolucao_camera_frontal_direita = resolucao_camera_frontal.split(" ")[0].
125         resolucao_camera_frontal_esquerda = resolucao_camera_frontal.split(" ")[0]
126     except:
127         resolucao_camera_frontal = "Valor não informado"
128         resolucao_camera_frontal = "Valor não informado"
129         resolucao_camera_frontal_direita = 0
130         resolucao_camera_frontal_esquerda = 0
131
132     try:
133         capacidade_bateria = WebDriverWait(driver, 5).until(EC.presence_of_element
134         capacidade_bateria = capacidade_bateria.text
135     except:
136         capacidade_bateria = "Valor não informado"
137         capacidade_bateria = "Valor não informado"
138
139
140     driver.back()
141     time.sleep(3)
142
143     df = df.append({'modelo':modelo,'marca':marca, 'lançamento':lançamento, 'memor
144                    'memoria_interna':memoria_interna,'tamanho_tela':tamanho_tela,
145                    'resolucao_tela_direita':resolucao_tela_direita, 'resolucao_te
146                    'resolucao_camera_principal':resolucao_camera_principal,'resol
147                    'resolucao_camera_principal_esquerda':resolucao_camera_princip
148                    'resolucao_camera_frontal':resolucao_camera_frontal,'resolucao
149                    'resolucao_camera_frontal_esquerda':resolucao_camera_frontal_e
150                    'capacidade_bateria':capacidade_bateria}, ignore_index=True)
151
152
153     cont+=1
154     if pg_atual != qtd_pg_total:
155         proxima = driver.find_element_by_xpath('/html/body/form/main/div[2]/div/div[2]
156         proxima.click()
157         escopo = driver.find_element_by_xpath('/html/body/form/main/div[2]/div/div[2]/
158         escopo = escopo.text
159         escopo = escopo.split()
160         qtd_cell = int(escopo[1])
161         pg_atual = int(escopo[5])
162     else:
163         break
164 else:
165     print('finish')
166
167 df.to_excel('df.xlsx') # Exportar em .xlsx
```

...

In [5]:

```
1 df
```

Out[5]:

	modelo	marca	lançamento	memoria_ram	memoria_interna	tamanho_tela	resolucao_tela	resolucao
0	Apple iPhone 13	Apple	setembro, 2021	6GB LPDDR4X	128GB (109GB disponível)\nNVMe (SSD)	6.1" polegadas	1170x2532 pixels (19.5:9)	
1	Apple iPhone 13 mini	Apple	setembro, 2021	6GB LPDDR4X	128GB (109GB disponível)\nNVMe (SSD)	5.4" polegadas	1080x2340 pixels (19.5:9)	
2	Apple iPhone 13 Pro	Apple	setembro, 2021	8GB LPDDR4X	128GB (109GB disponível)\nNVMe (SSD)	6.1" polegadas	1170x2532 pixels (19.5:9)	
3	Apple iPhone 13 Pro Max	Apple	setembro, 2021	8GB LPDDR4X	128GB (109GB disponível)\nNVMe (SSD)	6.7" polegadas	1284x2778 pixels (19.5:9)	
4	Motorola Moto G50	Motorola	agosto, 2021	4GB LPDDR4X	128GB (109GB disponível)\nNVMe (SSD)	6.5" polegadas	720x1600 pixels (19.5:9)	

# Busca Completa

Resulta todos os modelo de cada linha do celular. Exemplo: Motorola Edge Pro s (128GB / 6GB), Motorola Edge Pro s (128GB / 8GB) e entre outros

In [6]:

```
1 columns = ['modelo', 'marca', 'lançamento', 'vendido', 'memoria_ram', 'memoria_interna',
2           'tamanho_tela', 'resolucao_tela', 'resolucao_tela_direita', 'resolucao_tela
3           'resolucao_camera_principal', 'resolucao_camera_principal_direita', 'resoluca
4           'resolucao_camera_frontal', 'resolucao_camera_frontal_direita', 'resolucao_ca
5           'capacidade_bateria']
6
7 df = pd.DataFrame(columns=columns)
8
9
10 driver = webdriver.Chrome(executable_path='./chromedriver.exe')
11
12 driver.get("https://chrome.google.com/webstore/detail/adblock-%E2%80%94best-ad-blocke
13
14 time.sleep(30)
15
16 driver.get("https://www.maiscelular.com.br/fichas-tecnicas/?aparelho=1&z=0")
17
18 cookies = driver.find_element_by_xpath('//*[@id="frmMain"]/div[3]/div/div/a')
19 cookies.click()
20 time.sleep(3)
21 escopo = driver.find_element_by_xpath('/html/body/form/main/div[2]/div/div[2]/div[4]/s
22 escopo = escopo.text
23 escopo = escopo.split()
24 pg_atual = escopo[5]
25 qtd_pg_total = int(escopo[7])
26 qtd_cell = int(escopo[1])
27 cont = 0
28
29 while (qtd_pg_total >= cont):
30     for seq in range(1, qtd_cell+1):
31
32         try:
33             nav = WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.X
34             nav.click()
35
36             modelos = WebDriverWait(driver, 10).until(EC.presence_of_element_located((
37             modelos.click()
38         except:
39             try:
40                 spam = WebDriverWait(driver, 10).until(EC.presence_of_element_located(
41                 spam.click()
42                 time.sleep(3)
43             except:
44                 driver.back()
45
46             nav = WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.X
47             nav.click()
48
49             modelos = WebDriverWait(driver, 10).until(EC.presence_of_element_located((
50             modelos.click()
51
52         time.sleep(10)
53
54     for mo in range(3, 15):
55         try:
56             modelos = WebDriverWait(driver, 10).until(EC.presence_of_element_locat
57             modelo = modelos.text
58             modelos.click()
```

```
60     try:
61         marca = WebDriverWait(driver, 10).until(EC.presence_of_element_loc
62         marca = marca.text
63     except:
64         marca = "brand"
65
66     try:
67         lancamento = WebDriverWait(driver, 10).until(EC.presence_of_elemen
68         lancamento = lancamento.text
69     except:
70         lancamento = "Não Informado"
71
72     try:
73         vendido = WebDriverWait(driver, 10).until(EC.presence_of_element_l
74         vendido = vendido.text
75
76     except:
77         vendido = "Não Informado"
78
79     try:
80         memoria_ram = WebDriverWait(driver, 10).until(EC.presence_of_eleme
81         memoria_ram = memoria_ram.text
82     except:
83         memoria_ram = "Valor não informado"
84         memoria_ram = "Valor não informado"
85
86     try:
87         memoria_interna = WebDriverWait(driver, 10).until(EC.presence_of_e
88         memoria_interna = memoria_interna.text
89     except:
90         memoria_interna = "Valor não informado"
91         memoria_interna = "Valor não informado"
92
93     try:
94         tamanho_tela = WebDriverWait(driver, 10).until(EC.presence_of_elem
95         tamanho_tela = tamanho_tela.text
96     except:
97         tamanho_tela = "Valor não informado"
98         tamanho_tela = "Valor não informado"
99
100    try:
101        resolucao_tela = WebDriverWait(driver, 10).until(EC.presence_of_el
102        resolucao_tela = resolucao_tela.text
103        resolucao_tela_direita = resolucao_tela.split(" ")[0].replace("x",
104        resolucao_tela_esquerda = resolucao_tela.split(" ")[0].replace("x"
105    except:
106        resolucao_tela = "Valor não informado"
107        resolucao_tela = "Valor não informado"
108        resolucao_tela_direita = "Valor não informado"
109        resolucao_tela_esquerda = "Valor não informado"
110
111    try:
112        resolucao_camera_principal = WebDriverWait(driver, 10).until(EC.pr
113        resolucao_camera_principal = resolucao_camera_principal.text
114        resolucao_camera_principal_direita = resolucao_camera_principal.sp
115        resolucao_camera_principal_esquerda = resolucao_camera_principal.
116    except:
117        resolucao_camera_principal = "Valor não informado"
118        resolucao_camera_principal = "Valor não informado"
119        resolucao_camera_principal_direita = "Valor não informado"
120        resolucao_camera_principal_esquerda = "Valor não informado"
```

```

121
122
123
124         try:
125             resolucao_camera_frontal = WebDriverWait(driver, 10).until(EC.pres
126             resolucao_camera_frontal = resolucao_camera_frontal.text
127             resolucao_camera_frontal_direita = resolucao_camera_frontal.split(
128             resolucao_camera_frontal_esquerda = resolucao_camera_frontal.split
129         except:
130             resolucao_camera_frontal = "Valor não informado"
131             resolucao_camera_frontal = "Valor não informado"
132             resolucao_camera_frontal_direita = 0
133             resolucao_camera_frontal_esquerda = 0
134
135         try:
136             capacidade_bateria = WebDriverWait(driver, 10).until(EC.presence_o
137             capacidade_bateria = capacidade_bateria.text
138         except:
139             capacidade_bateria = "Valor não informado"
140             capacidade_bateria = "Valor não informado"
141
142         driver.back()
143
144         df = df.append({'modelo':modelo,'marca':marca, 'lançamento':lançamento
145                        'memoria_interna':memoria_interna,'tamanho_tela':taman
146                        'resolucao_tela_direita':resolucao_tela_direita, 'reso
147                        'resolucao_camera_principal':resolucao_camera_principa
148                        'resolucao_camera_principal_esquerda':resolucao_camera
149                        'resolucao_camera_frontal':resolucao_camera_frontal,'r
150                        'resolucao_camera_frontal_esquerda':resolucao_camera_f
151                        'capacidade_bateria':capacidade_bateria}, ignore_index
152     except:
153         break
154
155     driver.back()
156     driver.back()
157
158     cont+=1
159
160     if pg_atual != qtd_pg_total:
161         proxima = WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.X
162         proxima.click()
163         escopo = WebDriverWait(driver, 10).until(EC.presence_of_element_located((By.XP
164         escopo = escopo.text
165         escopo = escopo.split()
166         qtd_cell = int(escopo[1])
167         pg_atual = int(escopo[5])
168     else:
169         break
170 else:
171     print('finish')
172
173 df.to_excel('xiomi-27082021.xlsx') # Exportar em .xlsx

```

...



In [12]:

```
1 df
```

Out[12]:

	modelo	marca	lançamento	vendido	memoria_ram	memoria_interna	tamanho_tela	resoluçã
0	Xiaomi Poco X3 GT (128GB)	brand	julho, 2021	(Global, Internacional)	8GB LPDDR4X	128GB (106GB disponível)\nUFS 3.1	6.6" polegadas	1080 pixel
1	Xiaomi Poco X3 GT (256GB)	brand	julho, 2021	(Global, Internacional)	8GB LPDDR4X	256GB (230GB disponível)\nUFS 3.1	6.6" polegadas	1080 pixel
2	Xiaomi Poco M3 Pro (128GB)	brand	maio, 2021	(Global, Internacional)	6GB LPDDR4X	128GB (106GB disponível)\nUFS 2.2	6.5" polegadas	1080 pixel
3	Xiaomi Poco M3 Pro (64GB)	brand	maio, 2021	(Global, Internacional)	4GB LPDDR4X	64GB (49GB disponível)\nUFS 2.2	6.5" polegadas	1080 pixel
4	Xiaomi Mi 11 Ultra (512GB)	brand	abril, 2021	(Global, Internacional)	12GB LPDDR5	256GB (230GB disponível)\nUFS 3.1	6.81" polegadas	1440 pixel