Nome: Clauder Noronha

https://claudernoronha.github.io/acoes-Petrobras/ (https://claudernoronha.github.io/acoes-Petrobras/)

Base de dados: Yahoo Finance - https://finance.yahoo.com/ (https://finance.yahoo.com/)

Maio de 2019

Bolsa de Valore - Petrobras

In [3]:

```
from IPython.display import Image
Image(url='valores.png')
```

Out[3]:

Petrobras, reconhecida mundialmente pela sua tecnologia de exploração de petróleo em águas ultraprofundas. Tem um grande poder de extrair óleo e gás. A petrobras esta presente em diferentes países, com atuação própria ou através de parcerias.

Para mais detalhes https://petrobras.com.br/ (https://petrobras.com.br/)

In [5]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
import warnings
warnings.filterwarnings('ignore')
```

```
C:\Users\claud\Anaconda3\lib\site-packages\statsmodels\tools\_testing.py:19:
FutureWarning: pandas.util.testing is deprecated. Use the functions in the p
ublic API at pandas.testing instead.
  import pandas.util.testing as tm
```

In [6]:

```
petro = pd.read_csv('PBR.csv')
petro.head()
```

Out[6]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2019-01-04	14.43	14.80	14.31	14.76	14.155388	21083900
1	2019-01-07	14.91	15.54	14.45	15.06	14.443099	37305100
2	2019-01-08	15.34	15.43	15.07	15.20	14.577363	17556900
3	2019-01-09	15.57	15.71	15.51	15.62	14.980160	19822300
4	2019-01-10	15.42	15.57	15.25	15.48	14.845895	15290900

Como nosso período de analise é grande, podemos relaxar com o número de casa decimais a considerar

In [7]:

```
petro = petro.round(2)
petro.head(2)
```

Out[7]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2019-01-04	14.43	14.80	14.31	14.76	14.16	21083900
1	2019-01-07	14.91	15.54	14.45	15.06	14.44	37305100

In [ ]:

```

```

Agora, vou determinar a forma do conjunto

In [8]:

```
petro.shape
```

Out[8]:

```
(398, 7)
```

O conjunto de dados tem 398 linhas e 7 colunas

In [ ]:

```

```

Descobrir valores nulos

In [9]:

```
petro.isnull().sum()
```

Out[9]:

```
Date         0
Open         0
High         0
Low          0
Close        0
Adj Close    0
Volume       0
dtype: int64
```

A base de dados não tem nenhum valor nulo.

In [ ]:

```

```

vamos verificar o tipo de dados de cada coluna

In [10]:

```
petro.dtypes
```

Out[10]:

```
Date          object
Open         float64
High         float64
Low          float64
Close        float64
Adj Close    float64
Volume         int64
dtype: object
```

In [ ]:

```

```

precisamos mudar a coluna Date que esta em object para datetime. Caso não mude, vai ser dificl lidar com os dados de séries temporais de maneira mais inteligene>

In [11]:

```
petro['Date'] = pd.to_datetime(petro['Date'])
petro.head(3)
```

Out[11]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2019-01-04	14.43	14.80	14.31	14.76	14.16	21083900
1	2019-01-07	14.91	15.54	14.45	15.06	14.44	37305100
2	2019-01-08	15.34	15.43	15.07	15.20	14.58	17556900

vamos conferir se teve a mudança de object para Datetime

In [12]:

```
petro.dtypes
```

Out[12]:

```
Date         datetime64[ns]
Open                float64
High                float64
Low                 float64
Close               float64
Adj Close           float64
Volume                int64
dtype: object
```

In [ ]:

```

```

Para obeter a duração total do tempo pelo qual estamos realizando essa análise.

In [13]:

```
petro['Date'].max() - petro['Date'].min()
```

Out[13]:

```
Timedelta('577 days 00:00:00')
```

Existem aproximadamente 252 negociaçãodias em um ano com uma média de 21 dias por mês ou 63 dias por trimestre. Dos 365 dias possíveis, 104 dias são finais de semana (sábado e domingo), quando as bolsas de valores estão fechadas.

In [ ]:

```

```

Queremos saber como a Petrobras saiu nos ultimos dois meses. Para isso vouusar o comando Describe.

In [14]:

```
petro.iloc[-90:].describe().astype(int)
```

Out[14]:

	Open	High	Low	Close	Adj Close	Volume
count	90	90	90	90	90	90
mean	7	7	7	7	7	27676776
std	1	1	1	1	1	10003161
min	5	5	4	5	5	11541200
25%	6	6	6	6	6	20582675
50%	7	7	7	7	7	25486350
75%	8	8	8	8	8	32595200
max	9	9	9	9	9	60850400

In [ ]:

```

```

Nos últimos 90 dias, o preçp médio de fechamneto das ações da petrobras foi de R$ 7.00 .

Por cerca de 75% do tempo, as ações foram negociada abaixo de R$8.00 *eatingiuomáximodeR* 9.00.

O volume máximo de ações neogciadas em um único dia foi de 60850400 com a quantidad ede mediana de 25486350.

In [ ]:

```

```

Variação geral no preço das ações

> Antes de avançarmos para uma investigação mais aprofundada, definiremos a coluna 'Data' como o índice do quadro de dados. Facilita a plotagem.

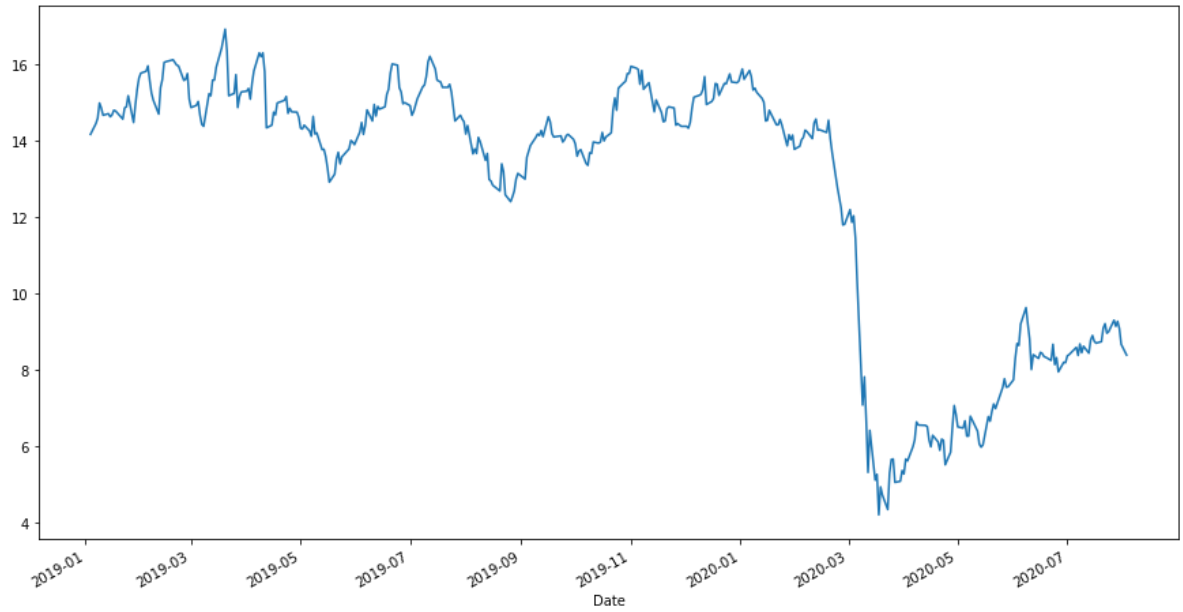
In [15]:

```
petro.index = petro['Date']
```

Agora plote o preço de fechamento (ajustado) das ações durante o período de 2 anos para obter uma idéia geral de como as ações se saíram no período especificado.

In [16]:

```
petro['Adj Close'].plot(figsize=(15,8))
title ='VAriação do preços das ações'
plt.show()
```



No gráfico acima, podemos observar, há uma queda drástica no preço das ações por volta de março/ abril de 2020.

As ações da Petrobras chegaram a desabar quase 30% nesta segunda-feira (9), após uma disputa entre Rússia e Arábia Saudita derrubar os preços do petróleo. É a maior queda já registrada.... - Veja mais em

https://economia.uol.com.br/cotacoes/noticias/redacao/2020/03/09/acoes-da-petrobras-tombo-do-petroleo-no-exterior.htm?cmpid=copiaecola (https://economia.uol.com.br/cotacoes/noticias/redacao/2020/03/09/acoes-da-petrobras-tombo-do-petroleo-no-exterior.htm?cmpid=copiaecola)

In [ ]:

```

```

Variação percentual diária (retornos diários)

> A variação percentual diária no preço das ações é calculada com base na variação percentual entre os preços de fechamento de 2 dias consecutivos. Digamos que o preço de fechamento das ações ontem fosse de R$500,00 e hoje as ações fecharam como R$550,00.

Portanto, a variação percentual é de 10%. ou seja ((550–500) / 500) * 100. Nenhum mistério aqui!

> Conseqüentemente, apresentaremos uma nova coluna ' Day_Perc_Change ' indicando os retornos diários no preço das ações. Isso pode ser feito usando a função pct_change () embutida no python.

In [17]:

```
#Vamos criar uam nova coluna com o nome day-perc-change
petro['Day_Perc_Change'] = petro['Adj Close'].pct_change()*100
petro.head()
```

Out[17]:

Date	Date	Open	High	Low	Close	Adj Close	Volume	Day_Perc_Change
2019-01-04	2019-01-04	14.43	14.80	14.31	14.76	14.16	21083900	NaN
2019-01-07	2019-01-07	14.91	15.54	14.45	15.06	14.44	37305100	1.977401
2019-01-08	2019-01-08	15.34	15.43	15.07	15.20	14.58	17556900	0.969529
2019-01-09	2019-01-09	15.57	15.71	15.51	15.62	14.98	19822300	2.743484
2019-01-10	2019-01-10	15.42	15.57	15.25	15.48	14.85	15290900	-0.867824

In [ ]:

```

```

temos valores nulo na coluna Day_Perc_Change , vamos tratar esses valores nulos.

In [18]:

```
petro.dropna(axis=0, inplace=True)
```

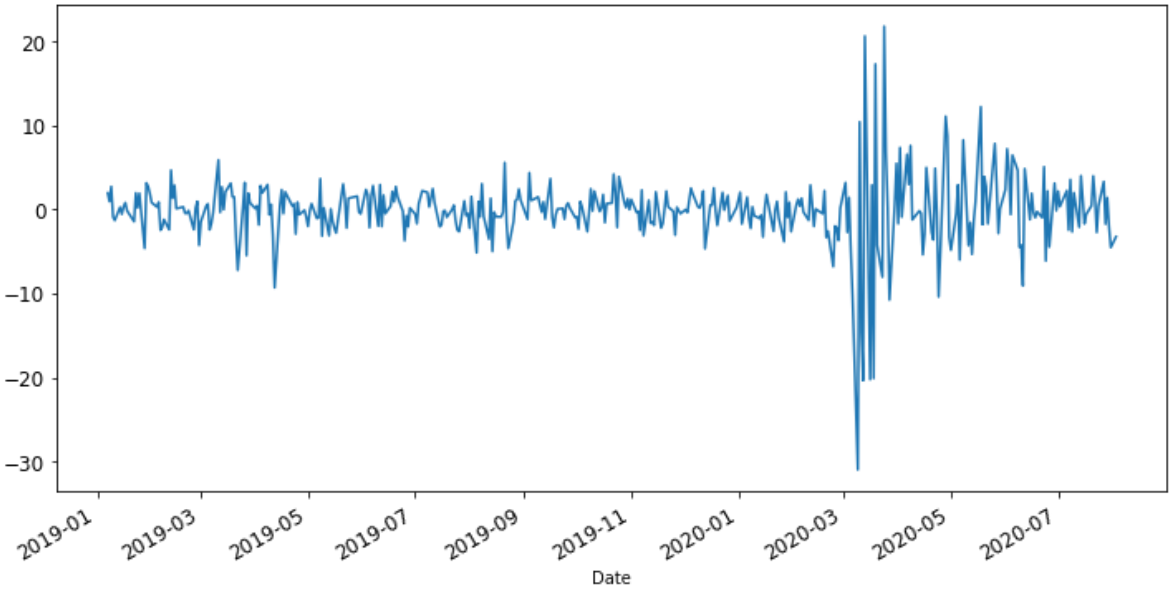
In [ ]:

```

```

Valores diarios em formato de grafico

In [19]:

```
petro['Day_Perc_Change'].plot(figsize=(12,6), fontsize = 12 )
plt.show()
```



Observamos que na maioria dos dias. os retornos estão entre -10 e 10%. Mas no mes de março e abril tivemos um pico negativo de -30%.

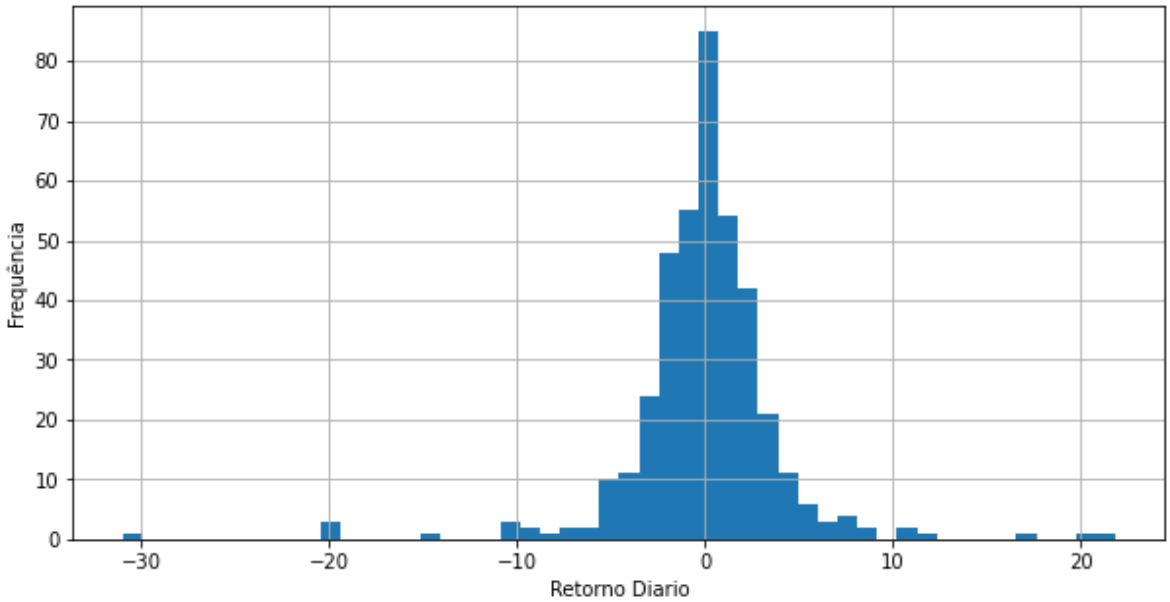
In [ ]:

```

```

Histograma de Distribuição de Retorno Diário

In [20]:

```
petro['Day_Perc_Change'].hist(bins=50, figsize=(10,5))
plt.xlabel('Retorno Diario')
plt.ylabel('Frequência')
plt.show()

petro.Day_Perc_Change.describe()
```



Out[20]:

```
count    397.000000
mean      -0.039319
std        4.207368
min      -30.994152
25%       -1.629328
50%        0.124688
75%        1.624294
max       21.839080
Name: Day_Perc_Change, dtype: float64
```

O histograma de retornos diários é centrado sobre a origem. Nos últimos, o retorno médio diário foi de 0,039 e na maioria dos dias o retorno diário foi inferior a 1%, o que implica que o estoque de PETRO4 foi menos volátil no período. Durante o período, a maior variação% na direção positiva foi de 21,83% e de -30,99% na direção negativa.

In [ ]:

```

```

Análise de tendência

Em seguida, adicionamos uma nova coluna ' Tendência ' cujos valores são baseados na alteração percentual diária calculada acima. A tendência é determinada abaixo do relacionamento -

In [21]:

```
#Criar uma função
def tendencia(x):
  if x > -0.5 and x <= 0.5:
    return 'Levemente sem alteração'
  elif x > 0.5 and x <= 1:
    return 'Levemente positivo'
  elif x > -1 and x <= -0.5:
    return 'Levemente Negativo'
  elif x > 1 and x <= 3:
    return 'Positivo'
  elif x > -3 and x <= -1:
    return 'Negativo'
  elif x > 3 and x <= 7:
    return 'Pricipal gannhador'
  elif x > -7 and x <= -3:
    return 'pricnipal Perdedor'
  elif x > 7:
    return 'Corrida do Touro'
  elif x <= -7:
    return 'Corrida do Urso'


petro['Tendencia']= np.zeros(petro['Day_Perc_Change'].count())
petro['Tendencia']= petro['Day_Perc_Change'].apply(lambda x:tendencia(x))
petro.head()
```

Out[21]:

Date	Date	Open	High	Low	Close	Adj Close	Volume	Day_Perc_Change	Tendencia
2019-01-07	2019-01-07	14.91	15.54	14.45	15.06	14.44	37305100	1.977401	Positivo
2019-01-08	2019-01-08	15.34	15.43	15.07	15.20	14.58	17556900	0.969529	Levemente positivo
2019-01-09	2019-01-09	15.57	15.71	15.51	15.62	14.98	19822300	2.743484	Positivo
2019-01-10	2019-01-10	15.42	15.57	15.25	15.48	14.85	15290900	-0.867824	Levemente Negativo
2019-01-11	2019-01-11	15.22	15.36	15.12	15.29	14.66	10191300	-1.279461	Negativo

In [ ]:

```
```

Desejamos ver como as ações estavam em alta nos últimos anos.

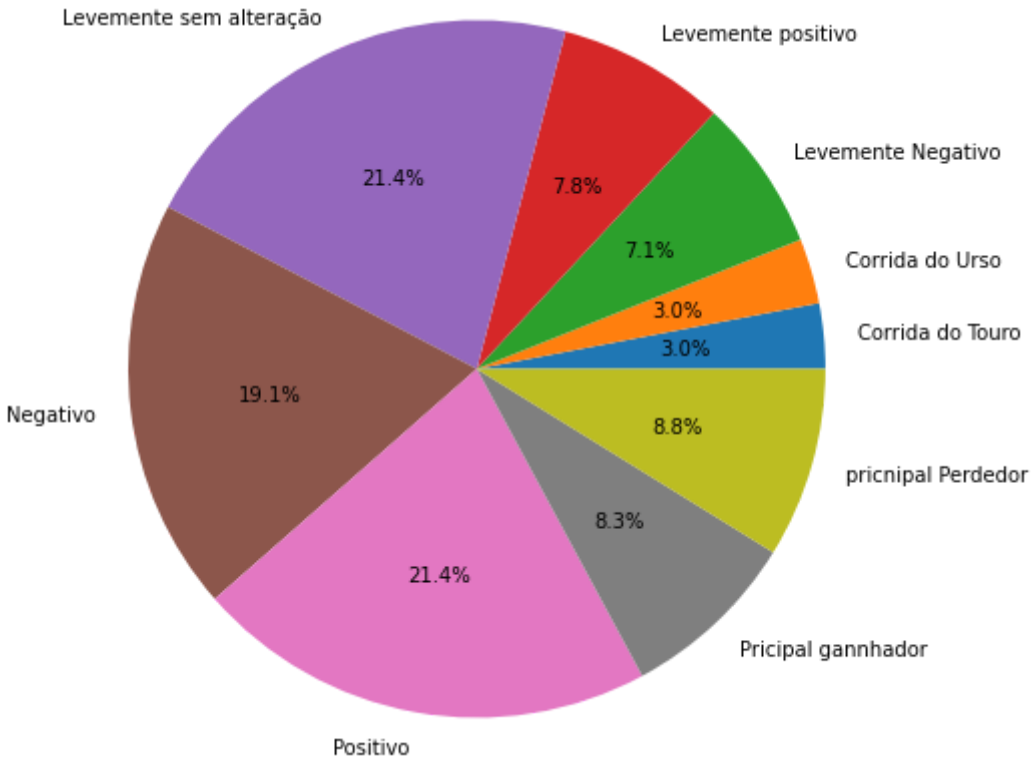
Isso pode ser visualizado como um gráfico de setores circulares, com cada setor representando a porcentagem de dias em que cada tendência ocorreu.

Traçaremos um gráfico de pizza para a coluna 'Tendência' para visualizar a frequência relativa de cada categoria de tendência.

Para isso, usaremos a função groupby () com a coluna de tendência para agregar todos os dias com a mesma tendência em um único grupo antes de plotar o gráfico de pizza.

In [22]:

```
data = petro.groupby('Tendencia')
pie_label = sorted([i for i in petro.loc[:, 'Tendencia'].unique()])
plt.pie(data['Tendencia'].count(), labels = pie_label,
        autopct = '%1.1f%%', radius=2)
plt.show()
```



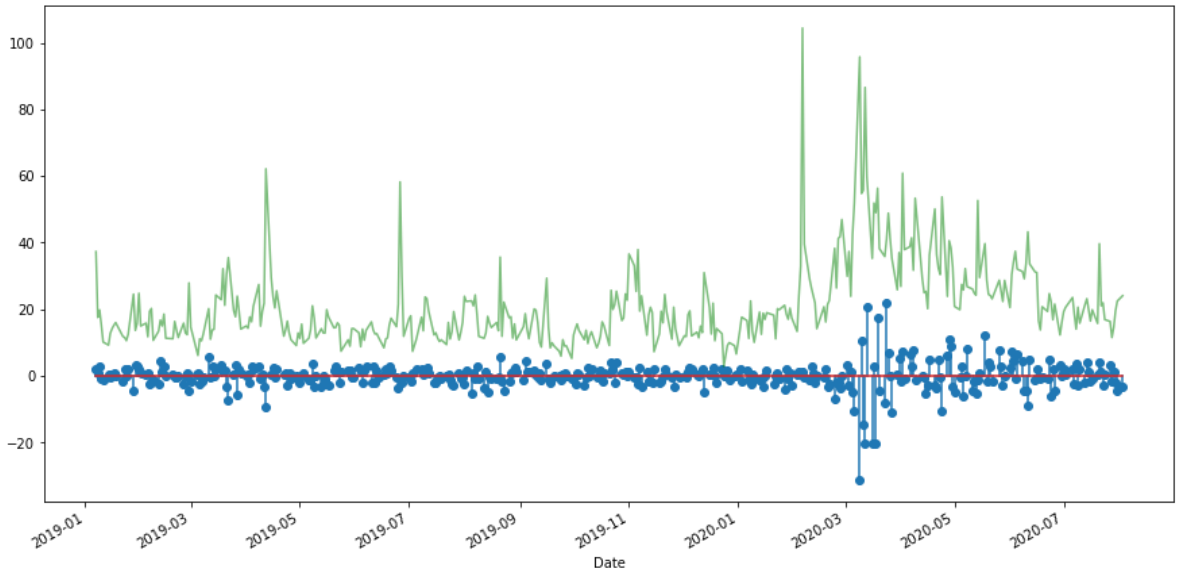
In [ ]:

```
```

Retorno diário e volume

In [23]:

```
plt.stem(petro['Date'], petro['Day_Perc_Change'])
(petro['Volume']/1000000).plot(figsize=(15, 7.5,),
                               color ='green',
                               alpha = 0.5)
plt.show()
```



(* O volume diário de negociação foi reduzido em escala para corresponder à escala de retorno diário)

Ao justapor o volume diário de negociação (em verde) com os retornos diários (em azul), observou-se que sempre que o volume de ações negociadas é alto, há um aumento ou queda comparativamente alto no preço das ações, levando a altos retornos .

Assim, em um determinado dia, se ocorrer um volume de negociação não convencional alto, é possível esperar uma grande mudança no mercado em qualquer direção. O volume de ações negociadas quando associado ao aumento ou queda do preço das ações, em geral, é um indicador da confiança dos traders e investidores em uma determinada empresa.

In [ ]:

```
```

Análise de correlação de estoques com plotagem par e plotagem conjunta

In [ ]:

```
```

NÃO queremos que os estoques sejam relacionados entre si. Matematicamente, o coeficiente de correlação de Pearson (também chamado de valor R de Pearson) entre qualquer par de ações deve ser próximo de 0.

A idéia por trás é simples - suponha que seu portfólio seja composto por ações altamente correlacionadas; se uma ação cair, as demais poderão cair também e você corre o risco de perder todo o seu investimento!

Selecionei as ações mencionadas acima para realizar a análise de correlação. Todas essas ações são de diferentes segmentos da indústria e do mercado. Você é livre para escolher as ações de seu interesse. o procedimento permanece o mesmo.

Na seção anterior, usamos o arquivo csv pré-baixado para análise. Nesta seção, teremos a ajuda do pacote do leitor de dados da web do Pandas para extrair os preços dos estoques.

In [39]:

```
#pip install pandas_datareader
```

In [32]:

```
# import package
import pandas_datareader.data as web
# set start and end dates
start = datetime.datetime(2019, 3, 15)
end = datetime.datetime(2020, 7, 14)
# extract the closing price data
dados = web.DataReader([ 'BBDC4.SA', 'BBAS3.SA','BPGIX'],
'yahoo', start = start, end = end)['Adj Close']
```

In [ ]:

```
```

Indice

BBDC4.SA ==== Bradesco

BBAS3.SA ==== BANCO DO BRASIL

BPGIX ======= CAIXA

In [33]:

```
dados.head(1000)
```

Out[33]:

Symbols	BBDC4.SA	BBAS3.SA	BPGIX
Date			
2019-03-15	35.351208	51.483345	15.937509
2019-03-18	35.536858	51.093971	16.026215
2019-03-19	34.771053	50.001816	15.996645
2019-03-20	33.827377	48.881168	15.898083
2019-03-21	33.084770	47.855476	15.976933
...	...	...	...
2020-07-08	22.430813	34.090000	14.170000
2020-07-09	21.931240	33.959999	13.950000
2020-07-10	22.250967	34.139999	14.170000
2020-07-13	21.941231	33.799999	14.170000
2020-07-14	22.310915	34.349998	14.420000

344 rows × 3 columns

```
Agora vamos procurar e tratar valores nulos
```

In [34]:

```
dados.isnull().sum()
```

Out[34]:

```
Symbols
BBDC4.SA    13
BBAS3.SA    13
BPGIX        8
dtype: int64
```

In [35]:

```
#Tratar valore nulos

dados.dropna(inplace=True, axis=0)

dados.head()
```

Out[35]:

Symbols	BBDC4.SA	BBAS3.SA	BPGIX
Date			
2019-03-15	35.351208	51.483345	15.937509
2019-03-18	35.536858	51.093971	16.026215
2019-03-19	34.771053	50.001816	15.996645
2019-03-20	33.827377	48.881168	15.898083
2019-03-21	33.084770	47.855476	15.976933

In [ ]:

```

```

Vamos visualisar com o grafico Seaborn de dorma pareada

In [36]:

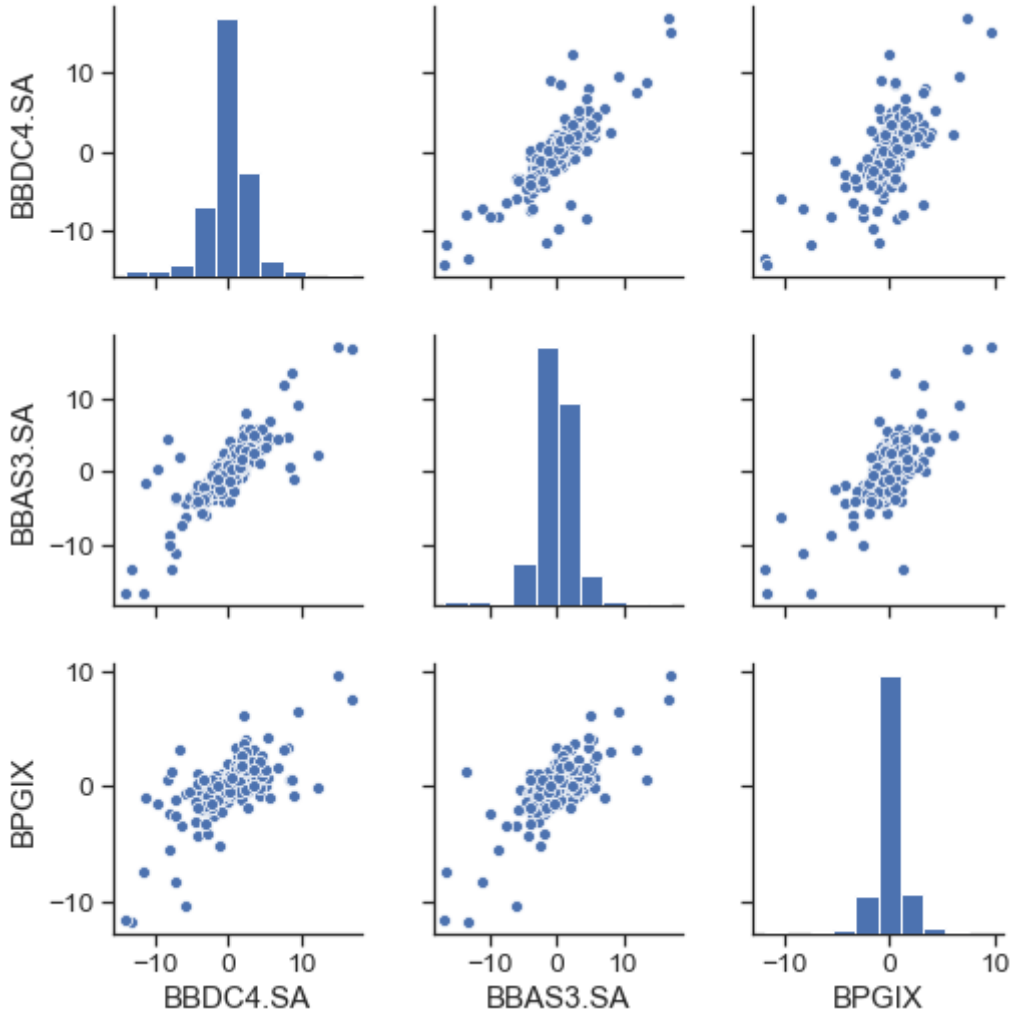
```
# armazena retornos diários de todos os estoques acima em um novo
analisar = dados.pct_change()*100
analisar.dropna(inplace = True, how='any', axis=0)
analisar.describe()
```

Out[36]:

Symbols	BBDC4.SA	BBAS3.SA	BPGIX
count	322.000000	322.000000	322.000000
mean	-0.085510	-0.064260	-0.011335
std	3.388965	3.494267	1.967131
min	-14.274416	-16.689509	-11.826347
25%	-1.491222	-1.578592	-0.519118
50%	-0.126496	-0.098257	0.066699
75%	1.428616	1.445266	0.610420
max	16.866913	17.126078	9.674332

In [37]:

```
sns.set(style = 'ticks', font_scale= 1.25)
sns.pairplot(analisar)

plt.show()
```



Observe que a análise de correlação é realizada na alteração percentual diária (retornos diários) do preço das ações e não no preço das ações.

Se você observar com cuidado, as plotagens na área triangular inferior são as mesmas que as plotagens na área triangular superior, com apenas os eixos trocados.

Portanto, analisar qualquer conjunto de parcelas seria suficiente.

Recua:

Embora os gráficos de pares forneçam uma visualização muito boa de todas as combinações possíveis entre o monte de ações, não fornecem informações detalhadas, como o valor R de Pearson ou o valor p de hipótese nula para quantificar a correlação. É aí que a trama conjunta entra em cena!

Enquanto a plotagem Pair fornece uma visão visual de todas as correlações possíveis, a plotagem conjunta Seaborn fornece informações detalhadas como o valor R de Pearson (coeficiente de correlação de Pearson) para cada par de ações. O valor R de Pearson varia de -1 a 1. O valor negativo indica uma relação linear

negativa entre as variáveis, enquanto o valor positivo indica uma relação positiva.

In [ ]:

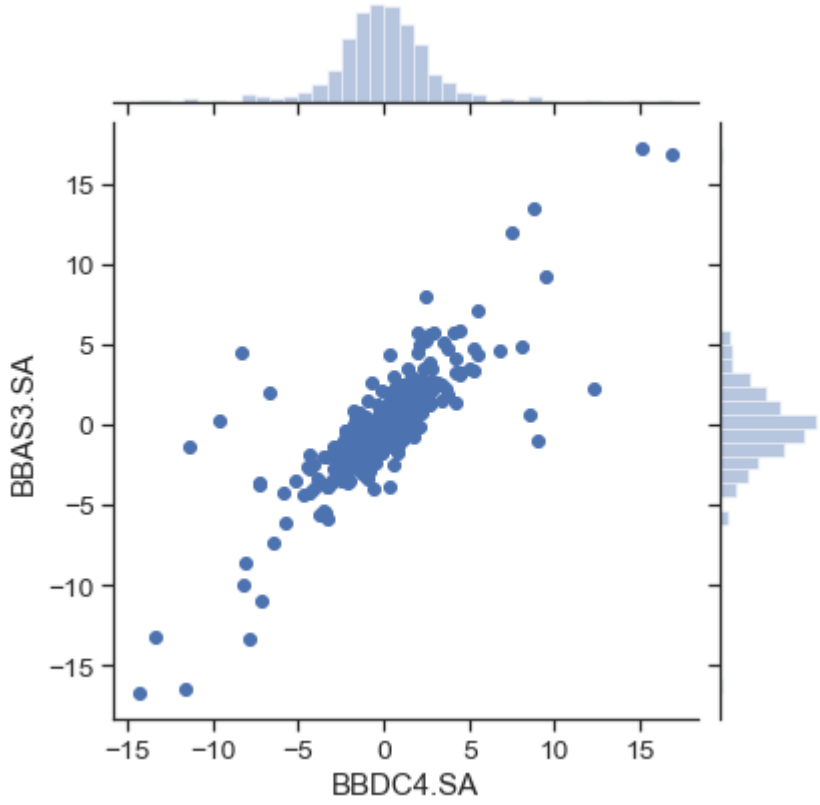
```
```

Comparação entre o BANCO DO BRASIL X BRADESCO

In [38]:

```
sns.jointplot('BBDC4.SA', 'BBAS3.SA', analisar, kind = 'scatter').annotate(stats.pearsonr)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-38-d52fadce3380> in <module>
----> 1 sns.jointplot('BBDC4.SA', 'BBAS3.SA', analisar, kind = 'scatter').an
notate(stats.pearsonr)

NameError: name 'stats' is not defined
```



Os resultados acima das plotagens conjuntas nos equipam com os números numéricos para verificar os insights que obtivemos observando visualmente a plotagem Pair anteriormente.

In [ ]:

```
```

Análise de volatilidade

A volatilidade é uma medida estática que mede o risco de um ativo, de acordo com a intensidade e frequência

de sua oscilação de preço em um determinado período de tempo.

Por meio dela, é possível entender o histórico de um ativo, qual a probabilidade de ele subir ou cair, de acordo com o período de tempo preestabelecido, e qual será a estimativa de oscilação do seu preço no futuro.

Se o preço de um ativo for muito volátil, por exemplo, é sinal de que sua cotação, em relação às flutuações do mercado, oscila muito, tornando sua compra arriscada, mas, por outro lado, proporciona maior possibilidade de lucro no curtíssimo prazo.

Tipos de volatilidade Depois de entender o que é volatilidade é necessário saber em quantos tipos elas se dividem e qual o papel de cada uma delas. Existem três tipos de volatilidade, que podem ser calculadas de maneira diferentes:

Volatilidade histórica A volatilidade histórica é o desvio padrão anualizado, que é calculado de acordo com as variações de preço de um determinado ativo em um período de tempo preestabelecido. Ela serve como balizador de qual será a previsão da volatilidade desse ativo no futuro.

Volatilidade implícita A volatilidade implícita é aquela que calcula a estimativa da volatilidade de um preço no futuro e pode ser obtida com base na volatilidade histórica e nos preços de ativos negociados na Bolsa, como os derivativos.

CLEAR (https://blog.clear.com.br/o-que-e-volatilidade/?campaignid=1676751011&adgroupid=77260160043&adid=389650790072&gclid=EAIaIQobChMI9MnHnJWC6wIV

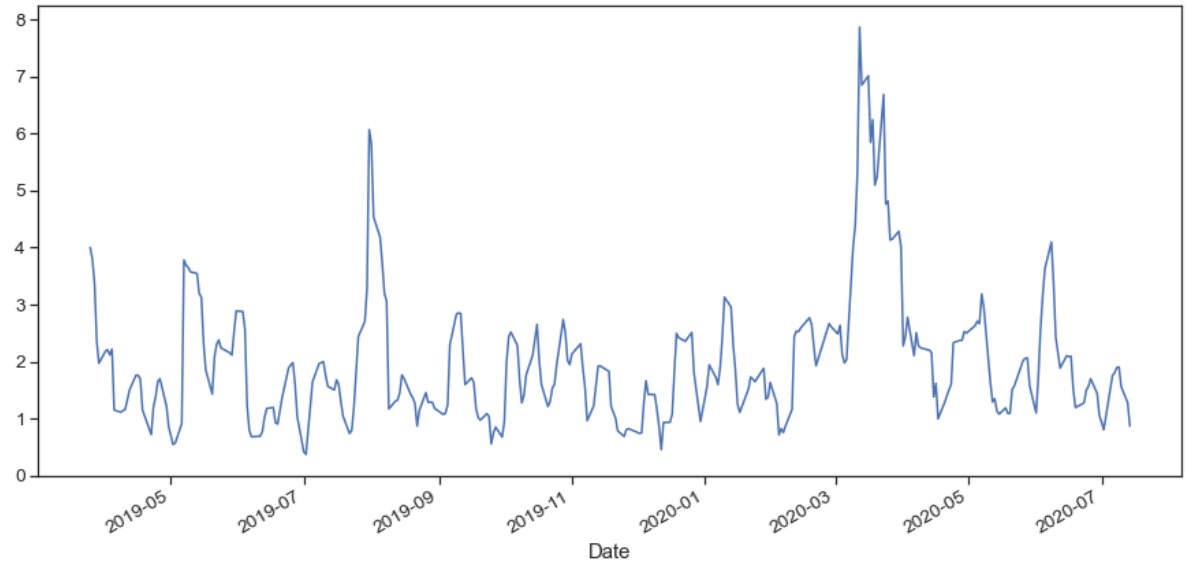
A volatilidade é um dos pilares mais importantes nos mercados financeiros. Diz-se que uma ação tem alta volatilidade se seu valor puder mudar drasticamente em um curto espaço de tempo.

Por outro lado, menor volatilidade significa que o valor das ações tende a ser relativamente estável ao longo de um período de tempo.

Esses movimentos são devidos a vários fatores, incluindo demanda e oferta, sentimentos, ações corporativas, ganância e medo, etc. Matematicamente, a volatilidade é medida usando uma medida estatística chamada 'desvio padrão', que mede a saída de um ativo em relação ao seu valor médio.

In [41]:

```
BBDC4 = dados['BBDC4.SA'].rolling(7).std() * np.sqrt(7)
BBDC4.plot(figsize = (15,7))

plt.show()
```



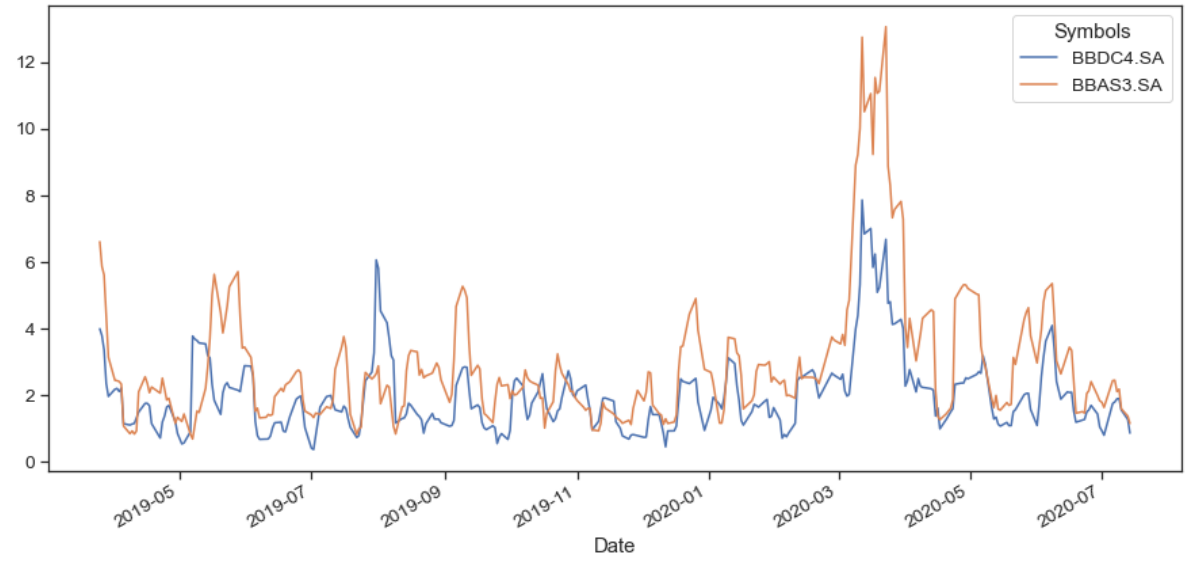
In [ ]:

```

```

A seguir, veremos a análise comparativa da volatilidade das ações BBDC4 com as ações SunPharma e o índice BPGIX. Assim como acima, calculamos a média móvel de 7 dias e o desvio padrão, tudo em uma única linha de código. Os pandas realmente facilitam nossa vida!

In [42]:

```
volatil = dados[['BBDC4.SA','BBAS3.SA']].rolling(7).std() * np.sqrt(7)
volatil.plot(figsize=(15,7))

plt.show()
```




```
Você pode observar que as ações da BBAS3 apresentam maior volatilidade em comparação com
as ações da BBDC4.
```

```
Muitos traders e investidores buscam investimentos de maior volatilidade para obter
lucros mais altos. Se uma ação não se move, não apenas apresenta baixa volatilidade, mas
também possui baixo potencial de ganho. Por outro lado, uma ação ou outro título com um
nível de volatilidade muito alto pode ter um tremendo potencial de lucro, mas o risco é
igualmente alto.
```

In [ ]:

```

```

Clauder Noronha

Junho de 2019

https://claudernoronha.github.io/acoes-Petrobras/ (https://claudernoronha.github.io/acoes-Petrobras/)

In [ ]:

```

```