

# Magic Formula

License Apache 2.0

GitHub Actions success

Projeto para usar a formula criada por Joel Greenblat no livro "The little book that beats the market" para analisar as acoes da Bovespa, **que fique claro que de forma alguma é uma recomendação de compra ou de venda, apenas um projeto pra auxiliar a analise.**

## Projeto

A ideia desse projeto é usar a formula do Joel Greenblat somado com algumas ideias propostas pelo Ramiro (Clube do Valor) para gerar uma planilha com as informações das ações mais baratas do indice IBRX100, outro proposito desse projeto é servir como um projeto para o meu portfolio como programador

## Requerimentos

Para executar esse programa os seguintes requerimentos devem ser atendidos: Versão do Python

```
$ Python 3.6.5
```

As seguintes libs são utilizadas:

```
pandas==1.2.4
openpyxl==3.0.7
yahooquery
bs4==0.0.1
requests==2.25.1
numpy==1.20.2
pytest==6.2.4
pytest-cov==2.11.1
```

As libs estão todas listadas no requirements.txt e pode ser instalado usando o pip conforme abaixo:

## Configuração

O programa usa um arquivo de configuração exemplificado no arquivo src/config.example.json

```
{
  "BRX100_URL": "https://statusinvest.com.br/indices/indice-brasil-100",
  "SMALL_URL": "https://statusinvest.com.br/indices/indice-small-cap",
  "IBOV_URL": "https://statusinvest.com.br/indices/ibovespa",
  "IDIV_URL": "https://statusinvest.com.br/indices/indice-dividendos",
  "STATUS_INVEST_URL":
  "https://statusinvest.com.br/category/advancedsearchresult?
  search=%7B%22Sector%22%3A%22%22%2C%22SubSector%22%3A%22%22%2C%22Segment%22
```

```
%3A%22%22%2C%22my_range%22%3A%220%3B25%22%2C%22dy%22%3A%7B%22Item1%22%3Anu
ll%2C%22Item2%22%3Anull%7D%2C%22p_L%22%3A%7B%22Item1%22%3Anull%2C%22Item2%
22%3Anull%7D%2C%22peg_Ratio%22%3A%7B%22Item1%22%3Anull%2C%22Item2%22%3Anu
ll%7D%2C%22p_VP%22%3A%7B%22Item1%22%3Anull%2C%22Item2%22%3Anull%7D%2C%22p_A
tivo%22%3A%7B%22Item1%22%3Anull%2C%22Item2%22%3Anull%7D%2C%22margemBruta%2
2%3A%7B%22Item1%22%3Anull%2C%22Item2%22%3Anull%7D%2C%22margemEbit%22%3A%7B
%22Item1%22%3Anull%2C%22Item2%22%3Anull%7D%2C%22margemLiquida%22%3A%7B%22I
tem1%22%3Anull%2C%22Item2%22%3Anull%7D%2C%22p_Ebit%22%3A%7B%22Item1%22%3An
ull%2C%22Item2%22%3Anull%7D%2C%22eV_Ebit%22%3A%7B%22Item1%22%3Anull%2C%22I
tem2%22%3Anull%7D%2C%22dividaLiquidaEbit%22%3A%7B%22Item1%22%3Anull%2C%22I
tem2%22%3Anull%7D%2C%22dividaLiquidaPatrimonioLiquido%22%3A%7B%22Item1%22%
3Anull%2C%22Item2%22%3Anull%7D%2C%22p_SR%22%3A%7B%22Item1%22%3Anull%2C%22I
tem2%22%3Anull%7D%2C%22p_CapitalGiro%22%3A%7B%22Item1%22%3Anull%2C%22Item2
%22%3Anull%7D%2C%22p_AtivoCirculante%22%3A%7B%22Item1%22%3Anull%2C%22Item2
%22%3Anull%7D%2C%22roe%22%3A%7B%22Item1%22%3Anull%2C%22Item2%22%3Anull%7D%
2C%22roic%22%3A%7B%22Item1%22%3Anull%2C%22Item2%22%3Anull%7D%2C%22roa%22%3
A%7B%22Item1%22%3Anull%2C%22Item2%22%3Anull%7D%2C%22liquidezCorrente%22%3A
%7B%22Item1%22%3Anull%2C%22Item2%22%3Anull%7D%2C%22pl_Ativo%22%3A%7B%22Ite
m1%22%3Anull%2C%22Item2%22%3Anull%7D%2C%22passivo_Ativo%22%3A%7B%22Item1%2
2%3Anull%2C%22Item2%22%3Anull%7D%2C%22giroAtivos%22%3A%7B%22Item1%22%3Anu
ll%2C%22Item2%22%3Anull%7D%2C%22receitas_Cagr5%22%3A%7B%22Item1%22%3Anull%2
C%22Item2%22%3Anull%7D%2C%22lucros_Cagr5%22%3A%7B%22Item1%22%3Anull%2C%22I
tem2%22%3Anull%7D%2C%22liquidezMediaDiaria%22%3A%7B%22Item1%22%3Anull%2C%2
2Item2%22%3Anull%7D%2C%22vpa%22%3A%7B%22Item1%22%3Anull%2C%22Item2%22%3Anu
ll%7D%2C%22lpa%22%3A%7B%22Item1%22%3Anull%2C%22Item2%22%3Anull%7D%2C%22val
orMercado%22%3A%7B%22Item1%22%3Anull%2C%22Item2%22%3Anull%7D%7D&CategoryTy
pe=1",
    "POSTGRESQL_STRING":
"postgresql+psycopg2://postgres:example@0.0.0.0/fmsdeinvestimento",
    "MAX_NUMBER_THREADS": 30
}
```

Esse arquivo deve ser usado como base para a criação do arquivo src/config.json

```
$ cp src/config.example.json src/config.json
```

## Diretamente

Devem ser instaladas as dependencias usando os seguintes comandos abaixo:

Windows:

```
# criando o ambiente virtual
$ python -m venv venv
# Iniciando o ambiente virtual
$ .\venv\Scripts\activate.bat
# Instalando as dependencias
$ python -m pip install -r requirements.txt
```

## Linux/Macos

```
# criando o ambiente virtual
$ python3 -m venv .venv
# Iniciando o ambiente virtual
$ source .venv/bin/activate
# Instalando as dependencias
$ python3 -m pip install -r requirements.txt
```

## Usando o docker

Usando o arquivo docker-compose.yml pode ser criada a imagem com o comando abaixo:

```
$ docker-compose build
```

## Executando

Podem ser verificados os comandos de usando o argumento -h:

```
usage: stocks_greenblat_magic_formula.py [-h] [-V] [-i INDEX] [-e EBIT]
                                         [-m MARKET_CAP] [-q QTY]
```

Parses command.

optional arguments:

```
-h, --help            show this help message and exit
-V, --version          Show program version
-i INDEX, --index INDEX
                        Bovespa index (BRX100, IBOV, SMALL, IDIV)
-e EBIT, --ebit EBIT  Minimun ebit to be considered
-m MARKET_CAP, --market_cap MARKET_CAP
                        Minimun market cap
-q QTY, --qty QTY     Quantity of stocks to be exported.
```

O programa pode ser executado usando o seguinte comando

```
$ python3 src/stocks_greenblat_magic_formula.py
```

Ou pode ser executado com o Docker usando o comando abaixo:

```
$ docker-compose up -d
```

Esse comando ja esta contido nos arquivos run.sh(Linux e Macos) e run.cmd(Windows), pode ser observado abaixo o output do programa abaixo:

```
2021-06-03 20:53:52,358 - main - (Thread-89) - DEBUG - Ticker: RAPT4.SA validated and its Valid.
2021-06-03 20:53:52,369 - main - (Thread-89) - DEBUG - Inserting ticker: RAPT4.SA on dataframe
2021-06-03 20:53:52,545 - main - (Thread-84) - DEBUG - Ticker: HYPE3.SA validated and its Valid.
2021-06-03 20:53:52,550 - main - (Thread-84) - DEBUG - Inserting ticker: HYPE3.SA on dataframe
2021-06-03 20:53:52,584 - main - (Thread-82) - DEBUG - Ticker: CPFE3.SA validated and its Valid.
2021-06-03 20:53:52,590 - main - (Thread-82) - DEBUG - Inserting ticker: CPFE3.SA on dataframe
2021-06-03 20:53:52,622 - main - (Thread-83) - DEBUG - Ticker: HAPV3.SA validated and its Valid.
2021-06-03 20:53:52,626 - main - (Thread-83) - DEBUG - Inserting ticker: HAPV3.SA on dataframe
2021-06-03 20:53:52,637 - main - (MainThread) - INFO - Processing ticker: SANB11 on thread <Thread(Thread-92, initial)>
2021-06-03 20:53:52,637 - main - (Thread-92) - DEBUG - Validating ticker: SANB11.SA
2021-06-03 20:53:52,638 - main - (MainThread) - INFO - Processing ticker: PRI03 on thread <Thread(Thread-93, initial)>
2021-06-03 20:53:52,639 - main - (Thread-93) - DEBUG - Validating ticker: PRI03.SA
2021-06-03 20:53:52,640 - main - (MainThread) - INFO - Processing ticker: PSSA3 on thread <Thread(Thread-94, initial)>
2021-06-03 20:53:52,641 - main - (Thread-94) - DEBUG - Validating ticker: PSSA3.SA
2021-06-03 20:53:52,642 - main - (MainThread) - INFO - Processing ticker: RADL3 on thread <Thread(Thread-95, initial)>
2021-06-03 20:53:52,644 - main - (Thread-95) - DEBUG - Validating ticker: RADL3.SA
2021-06-03 20:53:52,644 - main - (MainThread) - INFO - Processing ticker: IGTA3 on thread <Thread(Thread-96, initial)>
2021-06-03 20:53:52,649 - main - (Thread-96) - DEBUG - Validating ticker: IGTA3.SA
2021-06-03 20:53:52,650 - main - (MainThread) - INFO - Processing ticker: RENT3 on thread <Thread(Thread-97, initial)>
2021-06-03 20:53:52,651 - main - (Thread-97) - DEBUG - Validating ticker: RENT3.SA
2021-06-03 20:53:52,652 - main - (MainThread) - INFO - Processing ticker: EMBR3 on thread <Thread(Thread-98, initial)>
2021-06-03 20:53:52,654 - main - (Thread-98) - DEBUG - Validating ticker: EMBR3.SA
2021-06-03 20:53:52,656 - main - (MainThread) - INFO - Processing ticker: PETR3 on thread <Thread(Thread-99, initial)>
2021-06-03 20:53:52,657 - main - (Thread-99) - DEBUG - Validating ticker: PETR3.SA
2021-06-03 20:53:52,657 - main - (MainThread) - INFO - Processing ticker: ALPA4 on thread <Thread(Thread-100, initial)>
2021-06-03 20:53:52,660 - main - (Thread-100) - DEBUG - Validating ticker: ALPA4.SA
2021-06-03 20:53:52,661 - main - (MainThread) - INFO - Processing ticker: LINX3 on thread <Thread(Thread-101, initial)>
2021-06-03 20:53:52,663 - main - (Thread-101) - DEBUG - Validating ticker: LINX3.SA
2021-06-03 20:53:54,452 - main - (Thread-93) - DEBUG - Ticker: PRI03.SA validated and its Valid.
2021-06-03 20:53:54,456 - main - (Thread-93) - DEBUG - Inserting ticker: PRI03.SA on dataframe
2021-06-03 20:53:54,467 - main - (Thread-100) - DEBUG - Ticker: ALPA4.SA validated and its Valid.
2021-06-03 20:53:54,470 - main - (Thread-100) - DEBUG - Inserting ticker: ALPA4.SA on dataframe
2021-06-03 20:53:54,533 - main - (Thread-99) - DEBUG - Ticker: PETR3.SA validated and its Valid.
2021-06-03 20:53:54,535 - main - (Thread-99) - DEBUG - Inserting ticker: PETR3.SA on dataframe
2021-06-03 20:53:54,713 - main - (Thread-95) - DEBUG - Ticker: RADL3.SA validated and its Valid.
2021-06-03 20:53:54,717 - main - (Thread-95) - DEBUG - Inserting ticker: RADL3.SA on dataframe
2021-06-03 20:53:54,776 - main - (Thread-97) - DEBUG - Ticker: RENT3.SA validated and its Valid.
2021-06-03 20:53:54,779 - main - (Thread-97) - DEBUG - Inserting ticker: RENT3.SA on dataframe
2021-06-03 20:53:55,695 - main - (Thread-96) - DEBUG - Ticker: IGTA3.SA validated and its Valid.
2021-06-03 20:53:55,700 - main - (Thread-96) - DEBUG - Inserting ticker: IGTA3.SA on dataframe
2021-06-03 20:53:55,818 - main - (MainThread) - INFO - Processing ticker: KLBW11 on thread <Thread(Thread-102, initial)>
2021-06-03 20:53:55,819 - main - (Thread-102) - DEBUG - Validating ticker: KLBW11.SA
2021-06-03 20:53:55,820 - main - (MainThread) - INFO - Sorting dataframe
2021-06-03 20:53:55,828 - main - (MainThread) - INFO - Exporting data into excel /Users/marinellirubens/Programs/magic_formula/xlsx_files/stocks_magic_formula_20210603.xlsx
2021-06-03 20:53:57,878 - main - (Thread-102) - DEBUG - Ticker: KLBW11.SA validated and its Valid.
2021-06-03 20:53:57,884 - main - (Thread-102) - DEBUG - Inserting ticker: KLBW11.SA on dataframe
(venv) marinellirubens@MacBook-Air-de-Rubens magic_formula %
```

## Output

Como o objetivo desse programa é listar as ações por ordem de qual esta mais barata, um excel é exportado na pasta `xlsx_files` com o seguinte padrão de nome:

```
stocks_magic_formula_{yyyymmdd}.xlsx
```

Exemplo de arquivo:

symbol	magic_index	earning_yield	roic_index_number	roic	buy_recommendation	sell_recommendation	current_price	regular_market_time
VALE3	0	3,09	0	24,34	10	0	114,8	2021-06-02 17:13:06
MRFG3	3	4,08	2	21,94	10	5	18,96	2021-06-02 17:16:13
WEGE3	9	4,87	5	19,68	3	10	33,45	2021-06-02 17:13:06
BRDT3	10	4,68	7	17,39	0	0	27,35	2021-06-02 17:13:06
QUAL3	13	4,52	11	15,54	21	0	28,52	2021-06-02 17:13:05
EGIE3	13	5,21	6	19,43	2	9	40,44	2021-06-02 17:08:11
ABEV3	18	5,7	8	17,16	19	0	19,37	2021-06-02 17:13:06
JBSS3	21	5,49	13	15,26	9	2	30,37	2021-06-02 17:16:13
CRFB3	29	5,72	18	14,18	21	1	22,83	2021-06-02 17:08:11
CSNA3	32	8,05	4	21,59	0	25	46,59	2021-06-02 17:08:11
B3SA3	33	5,98	19	13,99	0	0	17,01	2021-06-02 17:13:06
EQTL3	34	5,77	22	13,32	24	0	24,84	2021-06-02 17:16:13
PRI03	34	5,95	21	13,51	1	3	20,28	2021-06-02 17:09:00
BEEF3	34	7,06	17	14,26	33	0	10,19	2021-06-02 17:08:11
RAPT4	34	5,18	28	11,72	12	8	15	2021-06-02 17:16:13
GOAU4	35	5,51	26	11,77	10	0	14,99	2021-06-02 17:13:06