

# Dicionário de Dados e Funções do Projeto Radar BEST+

## Visão Geral

Este documento detalha os principais componentes do script `radar_best.ipynb`. O objetivo do script é analisar o mercado de ações brasileiro para identificar empresas que se alinham à filosofia de investimento "BEST" de Luiz Barsi, enriquecida com o cálculo do Preço-Teto.

## I. Parâmetros Globais

Estes são os valores de configuração definidos no início do script. Eles controlam o comportamento da análise e podem ser ajustados pelo usuário.

Parâmetro	Tipo	Descrição	Exemplo de Valor
<code>SETORES_BARSI</code>	Lista de Strings	Define os setores de atuação que serão considerados na análise, de acordo com a estratégia BEST. Os nomes devem ser exatamente como os retornados pela API do <code>yfinance</code> .	<code>['Financial Services', 'Utilities', ...]</code>
<code>SCORE_MINIMO_PARA_EXIBIR</code>	Inteiro	Filtro de resultado. Apenas as empresas que atingirem uma pontuação igual ou maior que este valor serão exibidas no ranking final.	3
<code>PAUSA_ENTRE_REQUISICOES</code>	Float (decimal)	O tempo em segundos que o script espera entre a análise de um ticker e outro. Essencial para evitar bloqueios por excesso de requisições às APIs.	0.5

## II. Funções Principais

Estas são as três funções que formam o coração do nosso sistema de análise.

### 1. `obter_todos_tickers_da_b3()`

- **Descrição:** Responsável pela **descoberta de ativos**. Esta função busca uma lista completa e atualizada de todos os códigos (tickers) de ações negociadas na B3.
- **Parâmetros (Entrada):** Nenhum.
- **Retorno (Saída):** Uma **lista** de **strings**, onde cada string é um ticker formatado para a biblioteca **yfinance** (ex: `['PETR4.SA', 'VALE3.SA', ...]`).
- **Fonte de Dados:** Bravo API ([brapi.dev](https://brapi.dev)).

### 2. `calcular_best_score(info_dict)`

- **Descrição:** Calcula a pontuação base (de 0 a 4) de uma empresa. Esta função aplica 4 filtros quantitativos baseados em dados fundamentalistas atuais.
- **Parâmetros (Entrada):** `info_dict` (um **dicionário** com os dados da empresa, obtido da função `.info` do **yfinance**).
- **Retorno (Saída):** Um **inteiro** representando o score da empresa (de 0 a 4).
- **Fonte de Dados:** Yahoo Finanças (via **yfinance**).

### 3. `calcular_preco_teto(ticker_obj)`

- **Descrição:** Calcula o **Preço-Teto** de uma ação, a principal métrica de Barsi para definir o valor máximo de compra.
- **Parâmetros (Entrada):** `ticker_obj` (o objeto **Ticker** criado pela biblioteca **yfinance**, que permite acesso a dados históricos).
- **Retorno (Saída):** Um **float** (decimal) representando o Preço-Teto calculado. Retorna `0.0` se não houver um histórico de dividendos confiável (mínimo de 3 anos de pagamentos nos últimos 5 anos).
- **Fonte de Dados:** Yahoo Finanças (via **yfinance**, função `.dividends`).

### III. Estrutura de Dados Final (DataFrame do Ranking)

Este dicionário descreve cada coluna do arquivo CSV e da tabela final que o projeto gera.

Coluna	Tipo de Dado	Descrição	Exemplo
<b>Ticker</b>	<code>String</code>	O código de negociação da ação na B3.	<code>TAE11.SA</code>
<b>Empresa</b>	<code>String</code>	O nome resumido da empresa.	<code>TAESA UNT N2</code>
<b>Setor</b>	<code>String</code>	O setor de atuação da empresa.	<code>Utilities</code>
<b>BEST+ Score</b>	<code>String</code>	A pontuação final (de 0 a 5), somando o score base e o ponto do Preço-Teto.	<code>5/5</code>
<b>Preço Atual</b>	<code>Float</code>	A cotação da ação no momento da análise.	<code>35.80</code>
<b>Preço-Teto</b>	<code>Float</code>	O preço máximo de compra calculado pela média de 5 anos de dividendos.	<code>38.50</code>
<b>Abaixo do Teto?</b>	<code>String</code>	Informa se o <code>Preço Atual</code> é menor que o <code>Preço-Teto</code> .	<code>Sim</code>
<b>DY (%)</b>	<code>Float</code>	O Dividend Yield dos últimos 12 meses.	<code>10.50</code>
<b>P/L</b>	<code>Float</code>	O índice Preço/Lucro da empresa.	<code>8.20</code>
<b>Dívida/PL</b>	<code>Float</code>	A relação entre a Dívida e o Patrimônio Líquido da empresa.	<code>1.37</code>

### IV. Fluxo de Execução

O script principal orquestra as funções na seguinte ordem:

1. **Iniciação:** Define os parâmetros globais.
2. **Descoberta:** Chama `obter_todos_tickers_da_b3()` para obter a lista de todos os ativos.
3. **Iteração e Análise:** Inicia um loop por cada ticker da lista.
4. **Filtragem:** Dentro do loop, a primeira ação é verificar se o setor do ticker está na lista `SETORES_BRSI`. Se não estiver, o ativo é descartado.

5. **Pontuação:** Se o setor for válido, o script chama `calcular_best_score()` e `calcular_preco_teto()` para obter todos os dados e calcular o score final de 5 pontos.
6. **Armazenamento:** Se o score final for igual ou maior que o `SCORE_MINIMO_PARA_EXIBIR`, os dados completos da empresa são adicionados a uma lista de resultados.
7. **Finalização:** Após o loop, a lista de resultados é convertida em um DataFrame do Pandas, formatada e exibida na tela, além de ser salva em um arquivo CSV.