



MBA  
USP  
ESALQ

# Título do Trabalho de Conclusão de Curso

NOME DO ALUNO: WANDERLEI JOSÉ BOLZAN

NOME DO ORIENTADOR: PROF. MARCOS DOS SANTOS

# Apresentação

- ▶ Objetivo
- ▶ Contextualização do problema
- ▶ Apresentação do método Mult-Critério **MAIRCA**
- ▶ Explicação dos critérios a serem analisados: Rentabilidade, Risco, Volatilidade, Liquidez
- ▶ Abordagem referente aos recursos utilizados na linguagem R, para manipular os dados

# Objetivo

O objetivo dessa pesquisa é adquirir séries temporais de séries históricas ou atuais do site da bovespa, através dos recursos da linguagem R, como os comandos **yf()** e **o getSymbols()**.

Através dessas informações adquiridas da API bovespa, serão criados índices tais como:

- ▶ Índices de Retorno, para que seja possível validar que esses ativos deram como resultado em sua série histórica
- ▶ Obter os índices que serão usados como critérios no método MAIRCA que são: Rentabilidade, Risco, Volatilidade e Liquidez

Uma vez obtidas essas informações, será possível executar o método MAIRCA, para se obter um ranking de desempenho, e tomar uma decisão de compra.

# Contextualização do Problema

Conforme comentado na parte inicial desse TCC, temos: onde observamos no item Introdução que: Ao longo do século XX, e início de XXI ocorreram outras crises globais, das quais os principais motivos para essas crises foram por, fraude (corrupção), má gestão, guerras, como o que ocorreu nos Estados Unidos, que provocou falência de grandes bancos e contaminou o mundo todo, o calote da Argentina com relação ao não pagamento de sua dívida externa, etc.

# Apresentação do método Mult-Critério MAIRCA

De acordo com o que foi abordado nessa pesquisa, temos os passos necessários para a obtenção das matrizes que são pré-requisitos para a geração do ranking de opções, esses passos foram retirados dos artigos:

- ▶ **An integrated approach for fuzzy failure modes and effects analysis using fuzzy AHP and fuzzy MAIRCA** Soumava Boral<sup>a,b</sup>, Ian Howard<sup>b,\*</sup>, Sanjay K. Chaturvedi<sup>a</sup>, Kristoffer McKee<sup>b</sup>, V.N.A. Naikan<sup>a</sup>
- ▶ ***MultiAtributive Ideal-Real Comparative Analysis (MAIRCA)*** Capitão-Tenente Arthur Pinheiro de Araújo Costa Mestrando em Sistemas em Computação -IME

## Passos previstos para obtenção do MAIRCA

- ▶ **Passo1:** Determinar a Matriz de Decisão;
- ▶ **Passo2:** Definição de preferências para a escolha de alternativas  $PA_i$ ;
- ▶ **Passo3:** Determinar a Matriz de Decisão Teórica( $T_p$ );
- ▶ **Passo4:** Determinar a Matriz de Decisão Real( $T_r$ );
- ▶ **Passo5:** Determinar a Matriz de Decisão do Gap( $G$ );e
- ▶ **Passo6:** Ranqueamento das alternativas.

## Apresentação da Matriz de Decisão – Carteira

	Lucro	Lucro	Lucro	Lucro	Lucro	Lucro	Lucro	Lucro
Ações	Preço	Venda	Qtd/Ações	R\$ Investido	Rentabilidade	Risco	Vol.	Liquidez
<b>AZUL4.SA</b>	R\$ 18,1717	R\$ 18,1136	5000	R\$ 90.858,50	0,0032	5,3812	0,7012	0,2980
<b>BBDC3.SA</b>	R\$ 14,2905	R\$ 14,2856	15500	R\$ 221.502,75	0,0003	1,0982	0,3311	0,0048
<b>BBSE3.SA</b>	R\$ 23,4791	R\$ 23,5325	20000	R\$ 469.582,00	-0,0023	3,3719	0,2516	-0,0535
<b>BRAC11.SA</b>	R\$ 23,4288	R\$ 23,4480	30000	R\$ 702.864,00	-0,0008	2,1293	0,4244	0,0407
<b>BRAP4.SA</b>	R\$ 23,8808	R\$ 23,9103	15000	R\$ 358.212,00	-0,0012	2,9302	0,3421	0,0296
<b>BRFS3.SA</b>	R\$ 15,1487	R\$ 15,0860	10500	R\$ 159.061,35	0,0041	3,8438	0,5589	0,0624



## Passo 2 - Definição de preferências para a escolha de alternativas PAi

	Lucro	Lucro	Lucro	Lucro	Lucro	Lucro	Lucro	Lucro
Ações	Preço	Venda	Qtd/Ações	R\$ Investido	Rentabilidade	Risco	Vol.	Liquidez
AZUL4.SA	R\$ 18,1717	R\$ 18,1136	5000	R\$ 90.858,50	0,0032	5,3812	0,7012	0,2980
BBDC3.SA	R\$ 14,2905	R\$ 14,2856	15500	R\$ 221.502,75	0,0003	1,0982	0,3311	0,0048
BBSE3.SA	R\$ 23,4791	R\$ 23,5325	20000	R\$ 469.582,00	-0,0023	3,3719	0,2516	-0,0535
BRAC11.SA	R\$ 23,4288	R\$ 23,4480	30000	R\$ 702.864,00	-0,0008	2,1293	0,4244	0,0407
BRAP4.SA	R\$ 23,8808	R\$ 23,9103	15000	R\$ 358.212,00	-0,0012	2,9302	0,3421	0,0296
BRFS3.SA	R\$ 15,1487	R\$ 15,0860	10500	R\$ 159.061,35	0,0041	3,8438	0,5589	0,0624

Para se obter esse índice, que deve ser gerado através da

fórmula:

$$P_{A_i} = \frac{1}{m}; \sum_{i=1}^m P_{A_i} = 1, \quad i = 1, 2, \dots, m$$

dessa forma temos: 1/6  
portanto teremos: 0,16

## Passo3: Determinar a Matriz de Decisão Teórica(Tp)

Lucro	Lucro	Lucro	Lucro	Lucro
Valor	Rentabilidade	Risco	Volatilidade	Liquidez
0,02996	0,1717	0,0478	0,0955	0,0644

Uma vez obtidos os índices, temos condições de criar a matriz de decisão Real e para isso, temos as fórmulas a serem seguidas como constam o presente artigo:

$$t_{rij} = t_{pij} \cdot \left( \frac{x_{ij} - x_j^{\min}}{x_j^{\max} - x_j^{\min}} \right)$$

$$t_{rij} = t_{pij} \cdot \left( \frac{x_{ij} - x_j^{\max}}{x_j^{\min} - x_j^{\max}} \right)$$

$$T_r = \begin{matrix} & C_1 & C_2 & \dots & C_n \\ \begin{matrix} A_1 \\ A_2 \\ \dots \\ A_m \end{matrix} & \begin{bmatrix} t_{r11} & t_{r12} & \dots & t_{r1n} \\ t_{r21} & t_{r22} & & t_{r2n} \\ \dots & \dots & \dots & \dots \\ t_{rm1} & t_{rm2} & \dots & t_{rmn} \end{bmatrix} \end{matrix}$$

### Matriz de Decisão Real

	Lucro	Lucro	Lucro	Lucro	Lucro
Ações	Valor	Rentabilidade	Risco	Volatilidade	Liquidez
AZUL4.SA	0,012125	0,146540	0,047820	0,095560	-0,016465
BBDC3.SA	0,000000	0,070094	0,000000	0,016897	-0,016508
BBSE3.SA	0,028705	0,000000	0,025386	0,000000	-0,016547
BRAC11.SA	0,028548	0,038964	0,011512	0,036728	-0,016525
BRAP4.SA	0,029960	0,027828	0,020454	0,019235	-0,016531
BRFS3.SA	0,002681	0,171760	0,030655	0,065315	-0,016451

## Memorial de Cálculos

Pesos de cada critério	Obtenção dos Pesos
W1	0,1498
W2	0,8588
W3	0,2391
W4	0,4778
W5	0,322
W6	0,1031
PAI	0,2

$$w_j = \frac{E_j}{\sum_k E_k}$$

**Formula 1 – Quadro 1, quando o critério maximiza o lucro, Temos:**

$$= \$D\$15 * (D5 - \text{MÍNIMO}(\$D\$5:\$D\$10)) / (\text{MÁXIMO}(\$D\$5:\$D\$9) - \text{MÍNIMO}(\$D\$5:\$D\$9))$$

**Formula 2 – Quadro 1, quando o critério minimiza o lucro, Temos:**

$$= \$G\$15 * (G5 - (\text{MÁXIMO}(\$G\$5:\$G\$10))) / ((\text{MÍNIMO}(\$G\$5:\$G\$9)) - (\text{MÁXIMO}(\$G\$5:\$G\$9)))$$

- Nota 1 – Esse mesmo processo de ocorrer para cada critério, de cada linha.
- Nota 2 – Para calcular a matriz (TP), é necessário obter os pesos de cada critério, portanto, usa – se o critério que se deseja calcular e divide – se pelo índice PAI.



## Passo 5: Determinar a Matriz de Decisão do Gap(G)

	Lucro	Lucro	Lucro	Lucro	Lucro
Ações	Valor	Rentabilidade	Risco	Volatilidade	Liquidez
AZUL4.SA	0,017835	0,025220	0,000000	0,000000	0,080865
BBDC3.SA	0,029960	0,101666	0,047820	0,078663	0,080908
BBSE3.SA	0,001255	0,171760	0,022434	0,095560	0,080947
BRAC11.SA	0,001412	0,132796	0,036308	0,058832	0,080925
BRAP4.SA	0,000000	0,143932	0,027366	0,076325	0,080931
BRFS3.SA	0,027279	0,000000	0,017165	0,030245	0,080851

Para compor esta tabela, temos que seguir a fórmula abaixo:

$$G = T_p - T_r = \begin{bmatrix} g_{11} & g_{12} & \dots & g_{1n} \\ g_{21} & g_{22} & \dots & g_{2n} \\ \dots & \dots & \dots & \dots \\ g_{m1} & g_{m2} & \dots & g_{mn} \end{bmatrix} = \begin{bmatrix} t_{p11} - t_{r11} & t_{p12} - t_{r12} & \dots & t_{p1n} - t_{r1n} \\ t_{p21} - t_{r21} & t_{p22} - t_{r22} & \dots & t_{p2n} - t_{r2n} \\ \dots & \dots & \dots & \dots \\ t_{pm1} - t_{rm1} & t_{pm2} - t_{rm2} & \dots & t_{pmn} - t_{rmn} \end{bmatrix}$$

Em outras palavras, essa fórmula é a diferença de cada critério da matriz(TP), pelo valor de cada elemento da matriz(TR)

## Passo 6: Ranqueamento das alternativas

Somatório dos Gaps (Qi)		Rank
AZUL4.SA	0,123920	5
BBDC3.SA	0,339016	2
BBSE3.SA	0,371956	1
BRAC11.SA	0,310274	4
BRAP4.SA	0,328554	3
BRFS3.SA	0,155540	4

Para compor esta tabela, temos que seguir a fórmula abaixo:

- Basta somar as linhas para cada critério da matriz (Gap)

# O Projeto

Conforme dito anteriormente, este projeto fará a integração das informações geradas pelos **script's R** acima citadas e essas informações serão usadas, processadas para que seja possível criar um ranking que auxiliará o processo de tomada de decisão de qual ação possa ser adquirida, por um investidor através de análises do seu desempenho histórico.

Esse Software, que se trata do desenvolvimento de uma API padrão **Restful** que possui como principal regra de negócio criar uma matriz baseada nos dados de uma carteira de ações, onde será feita uma análise dos indicadores gerados pelos Scripts R que são:

- ▶ Rentabilidade
- ▶ Risco
- ▶ Liquidez
- ▶ Volatilidade

# Algoritmo para seja calculado o índice - Rentabilidade

## Rentabilidade

Baseado no script abaixo, calcula-se a Rentabilidade:

a) Cria-se o histórico temporal da ação que se deseja analisar:

```
env = new.env()  
getSymbols(dfCarteira1$ticker,src="yahoo",env=env,  
from=as.Date('2022-01-01'),  
to=as.Date('2022-12-31'))
```

b) Obter as informações de retorno da ação estudada

```
azul4_Vol = env$AZUL4.SA[,6]  
azul4_ret <- Return.calculate(Ad(azul4_Vol), method = "log") %>%  
na.omit()
```

# Algoritmo para seja calculado o índice - Rentabilidade

## Rentabilidade

Baseado no script abaixo, calcula-se a Rentabilidade:

c) Criar e incluir no data frame o atributo rentabilidade criado:

```
rentabilidade = mean(azul4_ret)
```

```
azul4 = dfCarteira1 %>% select(ticker, ref_date, price_adjusted, risco) %>%  
  mutate(rentabilidade)
```

# Algoritmo para seja calculado o índice - Risco

## ► Risco

O risco é calculado, baseado no desvio padrão, onde observa – se o Script abaixo:

```
BPAC11=yf_get(tickers = 'BPAC11.SA',  
first_date = first_date,  
last_date = last_date)  
risco=sd(BPAC11$price_adjusted)
```



# Algoritmo para seja calculado o índice - Volatilidade

## Volatilidade

Inicialmente, obten – se o data frame conforme é criado abaixo:

```
env = new.env()  
getSymbols(Df$ticker,src="yahoo",  
env=env,  
from=as.Date('2022-01-01'),  
to=as.Date('2022-12-31'))
```

# Algoritmo para seja calculado o índice - Volatilidade

```
Vol = env$acao[,6]
acao_ret <- Return.calculate(Ad(Vol), method = "log") %>%
na.omit()
acao_gspec <- ugarchspec(mean.model = list(armaOrder = c(0,0)),
variance.model = list("sGARCH"),
distribution.model = "norm")
acao_gfit <- ugarchfit(data = acao_ret, spec = acao_gspec)
```

- ▶ Na linha abaixo o índice é criado através do comando mutate:

```
acao = Df %>% select(ticker, ref_date, price_adjusted, risco) %>%
mutate(volatiliade = mediaVolatilidade)
```

## Algoritmo para seja calculado o índice - Liquidez

- ▶ O índice de liquidez está diretamente relacionado ao índice de rentabilidade, portanto, uma vez obtido o data frame de uma ação específica, o mesmo será usado para se obter esse índice que estará em função do índice de rentabilidade, dessa forma temos:

```
azul4=yf_get(tickers = 'AZUL4.SA',  
             first_date = first_date,  
             last_date  = last_date)
```

- ▶ Após criado o Data frame, temos:

```
azul4 = azul4 %>% select(ticker, ref_date, price_adjusted, risco, volatiliade,  
rentabilidade) %>%  
mutate(liquidez = mean(price_adjusted * rentabilidade/100))
```

# A Estrutura do Projeto – BackEnd Java/Spring boot

Esse projeto foi concebido, através do uso das tecnologias, como: IDE **Eclipse**, ele possui uma estrutura de pastas e pacotes que seu nome faz referência à sua funcionalidade. E nessa estrutura, temos os pacotes:

**entities** – Corresponde às entidades desse sistemas

Nesse projeto, existem 7 entidades, portanto, o banco de dados **tcc**, terá 7 tabelas de armazenamento de dados que serão:

- ▶ **tb\_ativos** – Tabela gerada pela classe do tipo entidade: **Ativos.java**
- ▶ **tb\_carteira** - Tabela gerada pela classe do tipo entidade: **Carteira.java**
- ▶ **tb\_segmentos** - Tabela gerada pela classe do tipo entidade: **Segmentos.java**
- ▶ **tb\_empresas** - Tabela gerada pela classe do tipo entidade: **Empresas.java**
- ▶ **tb\_user** - Tabela gerada pela classe do tipo entidade: **User.java**
- ▶ **tb\_acoes** - Tabela gerada pela classe do tipo entidade: **Acoes.java**

## A Estrutura do Projeto – Conclusão

Através dessas informações, contidas nessas tabelas, será possível, criar e manipular as matrizes usadas para se obter a análise **multicritério do MAIRCA**. Observando esse banco de dados, analisando o estado de sua execução é possível observar no **browse** do **PgAdmin** o preview dos dados incluídos na tabela **tb\_ativos** com os dados inseridos.

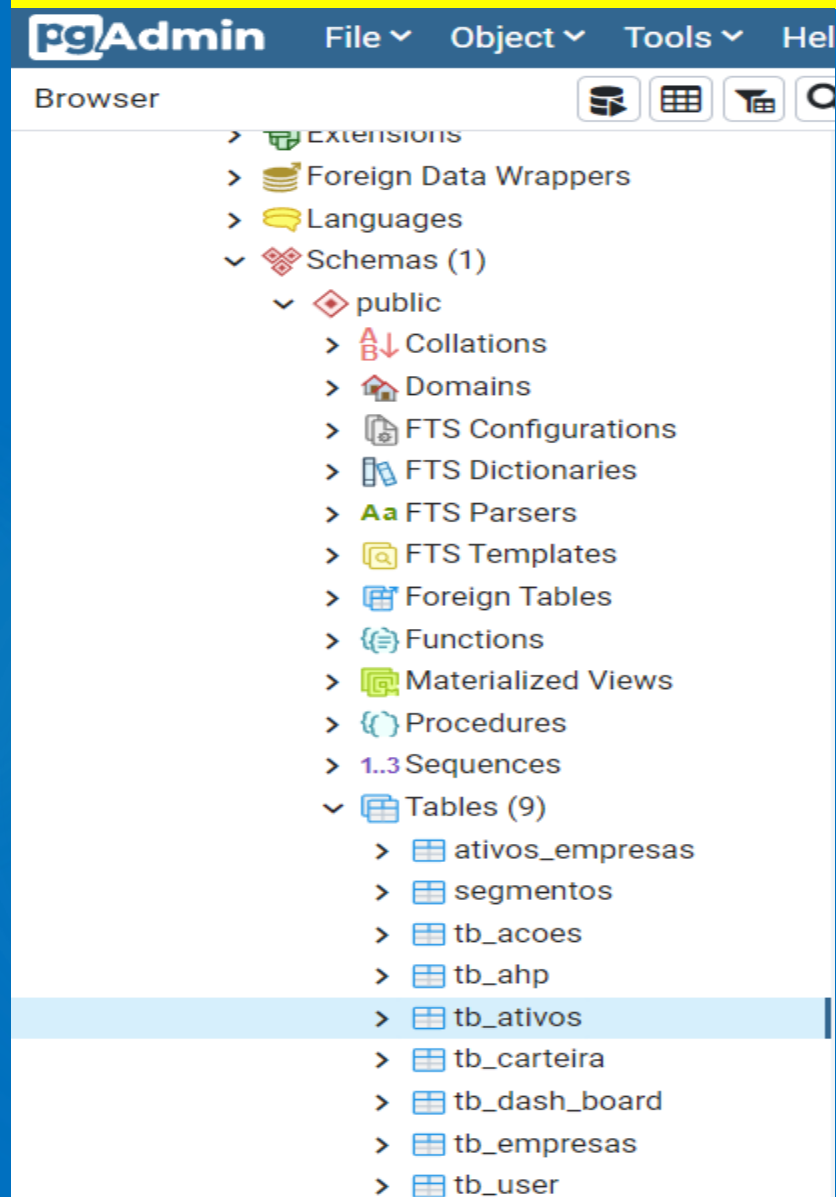
# Características técnicas da aplicação do projeto

O projeto, foi concebido fazendo uso das seguintes tecnologias:

- ▶ Linguagem Java – Versão 17
- ▶ IDE Eclipse
- ▶ **FrameWork Spring boot/Data**
- ▶ Gerenciador de dependências **Maven**
- ▶ Banco de Dados **postgreSql** e **browse de dados: Pgadmin**
- ▶ Linguagem de acesso e manipulação de dados – **Padrão SQL**
- ▶ Interface para acesso a dados – **JpaRepository**
- ▶ **Hibernate**, usado para criação automática de Queries de consulta e manipulação de dados.
- ▶ O **Postman** para testar a API
- ▶ Padrão do projeto, baseado no **protocolo HTTP**



# Demonstrando a Estrutura do Banco de Dados TCC criado



A criação do banco de dados Tcc, onde observamos existência de 9 tabelas, foi realizada no momento em que o software foi executado pela primeira vez. Como isso é feito:

1. No momento da execução do Software, o compilador identifica todas as classes que são do tipo – Entidade, ou seja, é identificada a instrução `@Entity`. Esse comando, diz ao JPA(Java Persistence API), que todas as classes do projeto que tiver esse compando serão entidades, ou seja, os seus atributos serão transformados em fields de uma tabela, e essa será inserida no banco de dados.
2. Após criada as tabelas, o software irá procurar pelos arquivos: `bovespa.csv` e `carteira.csv`, caso ele encontre, essas remessas de dados serão importadas nas bases de dados.
3. Após importadas as remessas, as mesmas serão excluídas.

# Exibição dos dados importados da tabela: tb\_ativos

	id [PK] bigint	abertura double precision	ativo character varying (255)	compra double precision	data date	fechamento double precision	id_empresa bigint	id_segmento bigint	maxima double precision
1	1	559999	AZUL4.SA	0	2022-01...	24.4	0	0	24.969999
2	2	23.91	AZUL4.SA	0	2022-01...	22.360001	0	0	24.299999
3	3	22.6	AZUL4.SA	0	2022-01...	23.049999	0	0	23.049999
4	4	22.73	AZUL4.SA	0	2022-01...	23.9	0	0	24.08
5	5	23.43	AZUL4.SA	0	2022-01...	23.629999	0	0	23.860001
6	6	23.5	AZUL4.SA	0	2022-01...	24.5	0	0	24.66
7	7	24.299999	AZUL4.SA	0	2022-01...	25.610001	0	0	25.65
8	8	25.24	AZUL4.SA	0	2022-01...	26.049999	0	0	26.559999
9	9	25.940001	AZUL4.SA	0	2022-01...	26.33	0	0	26.33
10	10	26.15	AZUL4.SA	0	2022-01...	25.82	0	0	26.549999
11	11	25.610001	AZUL4.SA	0	2022-01...	25.610001	0	0	26.040001
12	12	25.91	AZUL4.SA	0	2022-01...	25.27	0	0	26.65
13	13	25.4	AZUL4.SA	0	2022-01...	27.17	0	0	27.4
14	14	27.01	AZUL4.SA	0	2022-01...	27.610001	0	0	28.08
15	15	27.5	AZUL4.SA	0	2022-01...	26.43	0	0	27.75
16	16	26.209999	AZUL4.SA	0	2022-01...	26.92	0	0	27.190001
17	17	27.48	AZUL4.SA	0	2022-01...	27.389999	0	0	28.24
18	18	27.700001	AZUL4.SA	0	2022-01...	27.50	0	0	28.400001

# Testando o Projeto

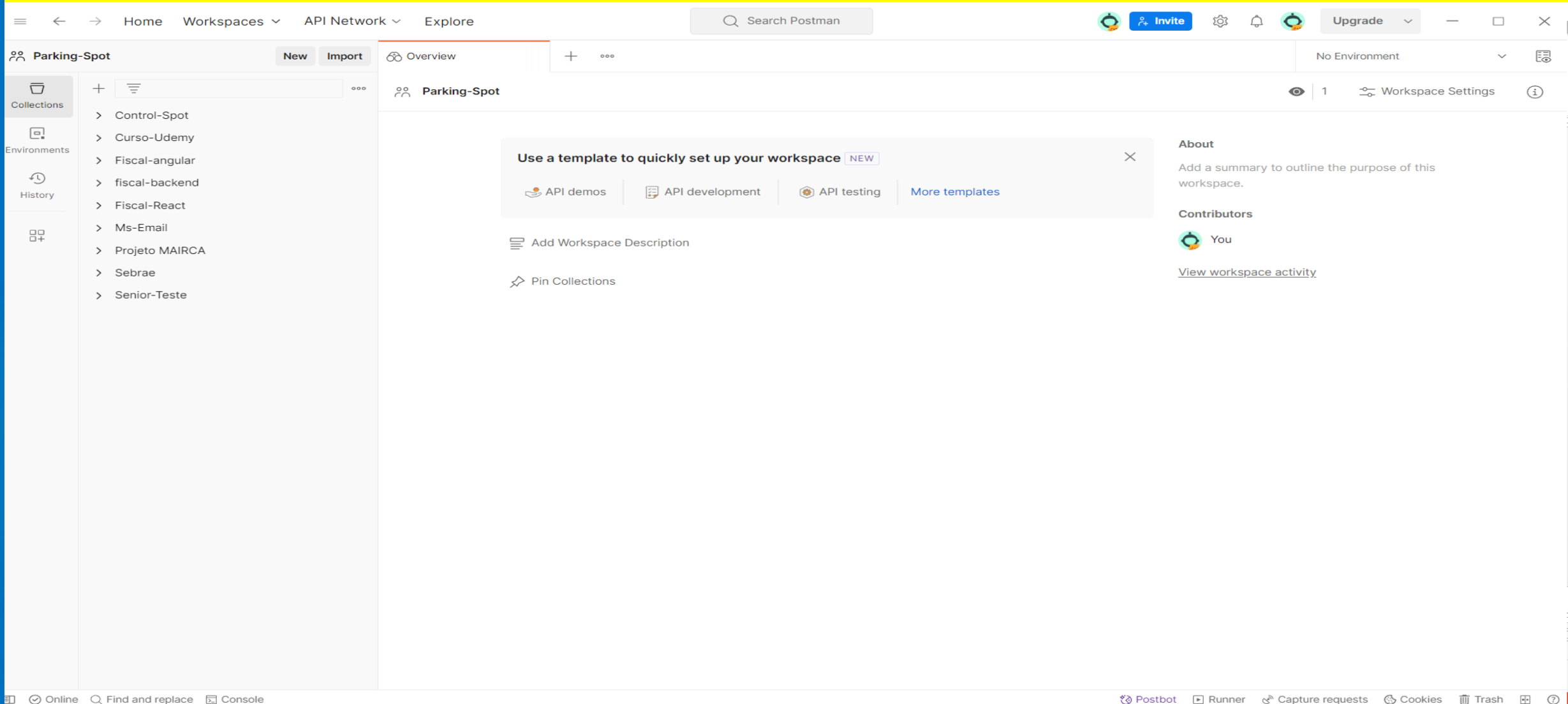
Para testar este software, é necessário antes executar o backend, pois esse software não está em ambiente de produção, está em ambiente de desenvolvimento. Portanto para executá-lo, deve – se antes abrir a **IDE** que no caso desse projeto, trata – se da **IDE Eclipse**.

Uma vez aberto o projeto, este deve ser compilado, após a compilação para que seja possível executar o projeto para testá-lo, devemos ter a possibilidade de usar o **postman**. Existem outros aplicativos, que possuem essa mesma finalidade.

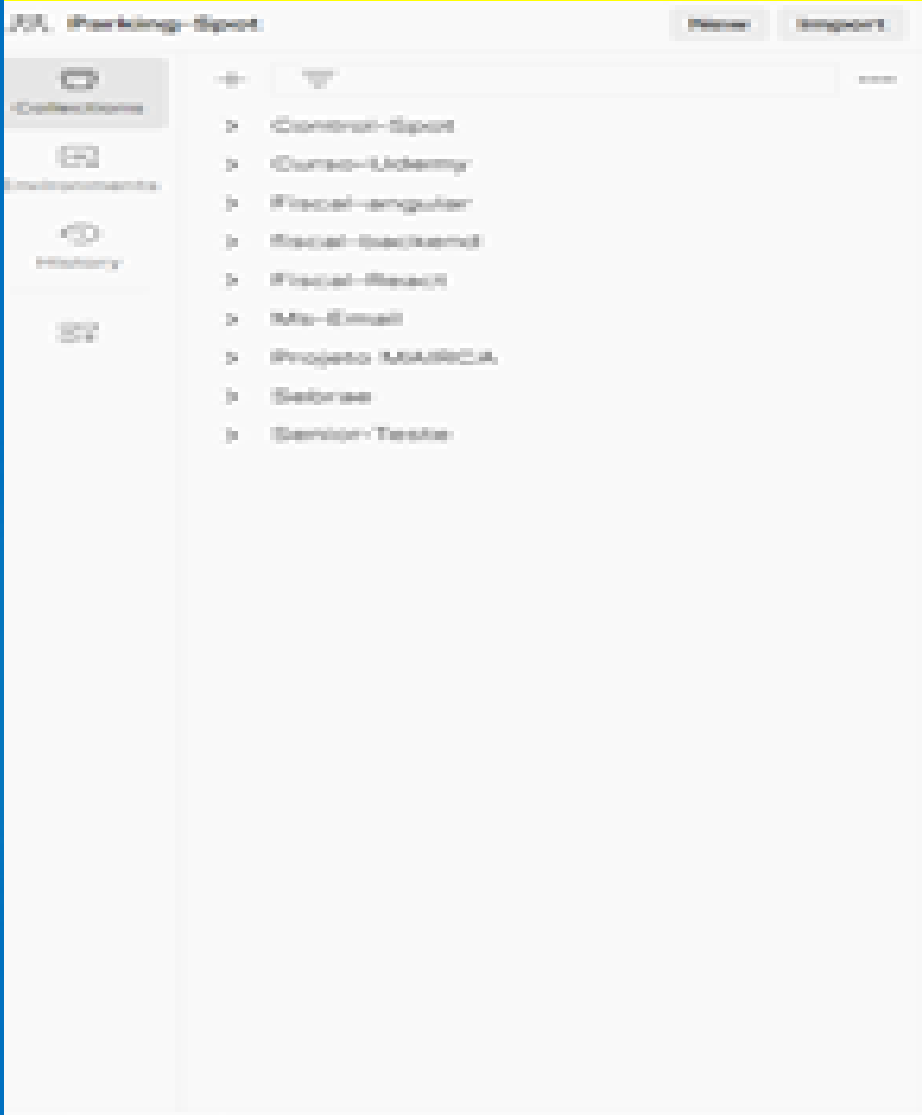
Uma vez o projeto estando compilado, ele também passa a estar em execução, portanto, é possível afirmar que há um servidor rodando. Feito isso basta no aplicativo **postman** e executar os seus **endpoints**

Os **endpoints**, são exatamente as rotas que foram criadas no **backend** do projeto, no pacote **resources**, para cada entidade existente no projeto.

# O Aplicativo Postman



# O Aplicativo Postman



- Essa imagem é um recorte da imagem anterior referente ao lado esquerdo da tela, que por sua vez contém todos os **end-pointes** necessários para que o software possa ser testado.
- Nesse caso, iremos acessar a aba **Projeto MAIRCA**, basta então clicar em cima dele, e serão disponibilizados todos os **endpoints** criados para o devido teste.

# O Aplicativo Postman – Testando a rota: <http://localhost:8080/ativos>

The screenshot displays the Postman application interface. On the left sidebar, a list of API endpoints is shown, with `http://localhost:8080/ativos` selected. The main panel shows the details of a GET request to this endpoint. The response status is 200 OK, and the time taken is 885 ms. The response body is displayed in JSON format, showing an array of objects representing active records.

**Query Params**

Key	Value	Description
Key	Value	Description

**Body** Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "ativo": "AZUL4.SA",
4   "id_Segmento": 0,
5   "id_Empresa": 0,
6   "data": "2022-01-04",
7   "abertura": 24.559999,
8   "maxima": 24.969999,
9   "minima": 24.040001,
10  "fechamento": 24.4,
11  "volume": 6910400.0,
12  "preco_ajustado": 24.4,
13  "maxima_Anterior": -8.19000819000992E-4,
14  "minima_Anterior": 23.75,
15  "compra": 0.0,
16  "venda": 0.0,
17  "resultados": 0.0
18 }
19
20
```

Nesse caso temos a rota: <http://localhost:8080/ativos> e como não foi observada a existência de um parâmetro, como por exemplo uma ação, então o software entendeu que tinha que executar o método `findall()`, portanto todos os registros da tabela `tb_ativos`, foram exibidos, que é algo próximo a 12000 registros.

Ao pressionar o botão **Send** que se localiza no canto superior direito da tela, dará início execução, no centro abaixo, pode – se observar uma estrutura, que está no padrão JSON.



# O Aplicativo Postman – Testando a rota:

<http://localhost:8080/ativos/nome?ativo=PETR3.SA>

The screenshot shows the Postman interface with a GET request to `http://localhost:8080/ativos/nome?ativo=PETR3.SA`. The request is successful, returning a 200 OK status with a response time of 178 ms and a size of 76.43 KB. The response body is displayed in JSON format, showing a list of two objects representing active assets.

Key	Value	Description
<input checked="" type="checkbox"/> ativo	PETR3.SA	
Key	Value	Description

```
1 [
2   {
3     "id": 8966,
4     "ativo": "PETR3.SA",
5     "id_Segmento": 0,
6     "id_Empresa": 0,
7     "data": "2022-01-04",
8     "abertura": 31.57,
9     "maxima": 32.189999,
10    "minima": 31.280001,
11    "fechamento": 31.92,
12    "volume": 1.73371E7,
13    "preco_ajustado": 17.625252,
14    "maxima_Anterior": 0.0126904240156058,
15    "minima_Anterior": 30.76,
16    "compra": 0.0,
17    "venda": 0.0,
18    "resultados": 0.0
19  },
20  {
21    "id": 8967,
```

Nesse caso temos a rota: <http://localhost:8080/ativos/nome?ativo=PETR3.SA> que terá a finalidade de fazer uma busca por nome de ativo, no caso dessa consulta, está sendo pesquisado o índice PETR3.SA. Portanto essa consulta retornará todas as ocorrências dessa ação informada na rota presente no banco de dados.

# O Aplicativo Postman – Testando a rota de Carteiras de Investimentos:

<http://localhost:8080/carteiras>

The screenshot shows the Postman interface with a GET request to `http://localhost:8080/carteiras`. The response is a JSON array of two investment portfolios, displayed in the 'Body' tab with the 'Pretty' format selected.

**Query Params**

Key	Value	Description
Key	Value	Description

**Body** | Cookies | Headers (8) | Test Results

Status: 200 OK | Time: 351 ms | Size: 1.82 KB | Save as Example

Pretty | Raw | Preview | Visualize | JSON

```
[
  {
    "id": 1,
    "nome_ativo": "AZUL4.SA",
    "valor_cotacao": 18.171759972,
    "qtd_acoes": 5000,
    "preco_venda": 18.1136250440394,
    "total_investido": 90568.1252201968,
    "rentabilidade": 0.054854390037112,
    "risco": 5.38125523601318,
    "liquidez": 0.701276851061257,
    "volatilidade": 0.00996800809164868
  },
  {
    "id": 2,
    "nome_ativo": "BBDC3.SA",
    "valor_cotacao": 14.290567116,
    "qtd_acoes": 15500,
    "preco_venda": 14.2856776456017,
    "total_investido": 220428.6025060268
  }
]
```


Nesse caso temos a rota: <http://localhost:8080/carteiras> que terá a finalidade de fazer uma busca de todos os itens pertencentes à carteira, portanto nesse caso, também será usado o método `findall()`.


# A Estrutura do Projeto – FrontEnd – Angular e Angular Material


Esse projeto foi concebido, através do uso das tecnologia para desenvolvimento de FrontEnd , como: **VsCode**, ele possui uma estrutura de pastas e pacotes que seu nome faz referência à sua funcionalidade. E nessa estrutura, temos os principais pacotes que fazem referencias às entidades:


- ▶ Src
  - ▶ App
    - ▶ Acoes
    - ▶ Ativos
    - ▶ Carteiras
    - ▶ Components
    - ▶ Containers
    - ▶ Dashboard
    - ▶ Empresas
    - ▶ Segmentos
    - ▶ services


# As interfaces do Projeto – Rota /ativos


 Dashboard


 Perfil/Usuários

 Ações Pesquisadas

 Segmentos/Eco

 Empresas

 Carteiras

 Índice MAIRCA


Dashboard


Série Temporal Atual


Relatório de Movimentação das Ações

Ativo/Ação.	Data	R\$ Open	R\$ Max.	R\$ Min.	R\$ Fechamento.	R\$ Volume.
AZUL4.SA	2022-01-04	24.559999	24.969999	24.040001	24.4	6910400
AZUL4.SA	2022-01-05	23.91	24.299999	22.360001	22.360001	9798600
AZUL4.SA	2022-01-06	22.6	23.049999	21.48	23.049999	14608500
AZUL4.SA	2022-01-07	22.73	24.08	22.309999	23.9	8012100
AZUL4.SA	2022-01-10	23.43	23.860001	23.02	23.629999	6738200
AZUL4.SA	2022-01-11	23.5	24.66	23.34	24.5	8032000
AZUL4.SA	2022-01-12	24.299999	25.65	24.25	25.610001	10824800
AZUL4.SA	2022-01-13	25.24	26.559999	24.969999	26.049999	9988900
AZUL4.SA	2022-01-14	25.940001	26.33	25.43	26.33	5495500

Usuário...







# As interfaces do Projeto – Rota /carteiras

← → ↻ ⓘ localhost:4200/#/carteiras

Caixa de entrada (23)... TV Globo Ao Vivo On... Cursos online - apren... WhatsApp Backend BFF NodeJS... Bem Vindo ao Projet... Telefônica | Vivo - No... Telefônica | Vivo - NO... Projetos · Pannel · GitL... CPFL Energia - Serviç...

Dashboard

Perfil/Usuários

Ações Pesquisadas

Segmentos/Eco

Empresas

**Carteiras**

Índice MAIRCA

Carteiras

Usuário...

**Módulo de Manutenção de Carteiras**

Carteiras de Investimentos

Ativo/Ação	R\$ Cotação	Qtd/Ações	R\$/Venda	R\$ Investido	Rentabilidade	Risco	Liquidez	Volatilidade
AZUL4.SA	18.17	5000	18.11	\$90,568.13	0.05	5.38	0.7	0.009968
BBDC3.SA	14.29	15500	14.29	\$221,428.00	0.07	1.1	0.33	0.009997
BBSE3.SA	23.48	20000	23.53	\$470,651.17	0.04	3.37	0.25	0.010023
BPAC11.SA	23.43	30000	23.45	\$703,441.12	0.04	2.13	0.42	0.010008
BRAP4.SA	23.88	15000	23.91	\$358,655.84	0.04	2.93	0.34	0.010012
BRFS3.SA	15.15	10500	15.09	\$158,403.26	0.07	3.84	0.56	0.009959