

## BasketStats

**Vinicius Goulart Almeida, Rodrigo Curvello<sup>2</sup>**

<sup>1,2</sup>Instituto Federal Catarinense – Campus Rio do Sul  
vivicoalmeida@gmail.com, rodrigo.curvello@ifc.edu.br

**Abstract.** *This article aims to present the BasketStats application developed for the Object Oriented Programming, which reports statistics taken from basketball matches, emphasizing the amount of basket performed by the player.*

**Key-words:** *Stats; Object Oriented Programming; Basketball.*

**Resumo.** *Este artigo tem como objetivo apresentar sobre o aplicativo BasketStats desenvolvido para a matéria de Programação Orientada à Objetos, onde se relata estatísticas retiradas de partidas de basquete, dando ênfase na quantidade de cestas realizadas pelo jogador.*

**Palavras-chave:** *Estatísticas; Programação Orientada à Objetos; Basquete.*

### 1. Introdução

O universo dos esportes é bastante complexo, sempre envolvendo técnicas e estratégias para a melhor performance dos atletas. Com o passar dos anos, novos métodos, acessórios e objetos passaram a ser desenvolvidos e implementados nas rotinas de diversas modalidades, com o intuito de proporcionar maior conforto e para explorar o que há de melhor. Nesse sentido, a tecnologia tem sido cada vez mais aliada para as práticas esportivas. Mais do que as tecnologias usadas nas roupas, calçados e aparelhos, as novas tecnologias proporcionam a obtenção de dados essenciais e seguros para a criação e cruzamento de informações para a reprodução de estatísticas individuais ou coletivas da prática esportiva. Com isso, a tomada de decisão de um treinador para com o seu atleta, foi aprimorada. Assim, são pontuados picos de desempenho, pontos fracos e fortes dos atletas, movimentos corretos e incorretos, entre outros detalhes que fazem toda a diferença na montagem dos treinos e na atuação do treinador para com seu atleta.

### 2. Linguagem de programação e paradigma utilizado

Para o desenvolvimento do aplicativo, foi utilizado a linguagem JAVA, por meio da IDE Eclipse, com o projeto totalmente focado no desenvolvimento com o paradigma de programação Orientada à Objeto. Eclipse é uma IDE que suporta várias outras linguagens a partir de plugins como C/C++, PHP e Python, foi feito em JAVA e segue o modelo *open source* de desenvolvimento de software.

### 3. Código fonte

O código fonte foi dividido em 4 pacotes, sendo elas:

O gerador com a geração dos dados necessários para o programa.

O analisador executa o estudo dos dados para execução e finalização de execução do software.

O *Main* inicia todo o sistema e “seta” algumas informações importantes.

A persistência salva os arquivos para se obter um uso fácil do software, sem perda de dados.

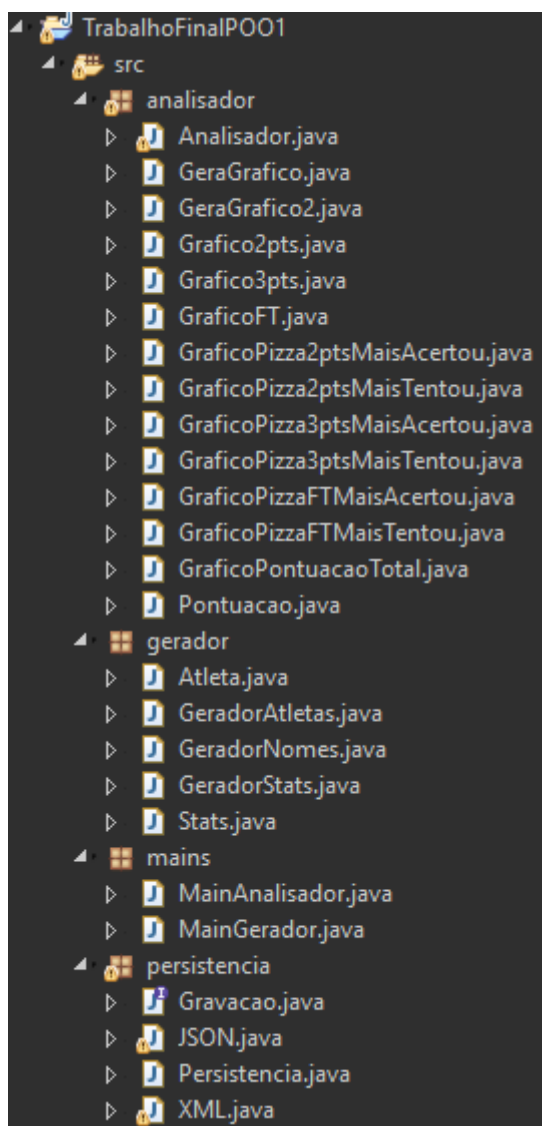


Figura 1. Distribuição dos pacotes e classes

#### 3.1. Gerador

O pacote gerador tem como ênfase a geração de todos os dados necessários para que o programa execute perfeitamente. Dentro do pacote se encontra 5 classes, sendo 3 delas geradores e 2 com as informações necessária para gerar os dados corretamente.

Na classe GeradorStats é gerado aleatoriamente os dados realizados pelos atletas em partidas, com ênfase em bolas de 3 pontos, 2 pontos e lances livres.

Na classe GeradorNomes é obtido nomes aleatórios recebidos pelo site curvello.com e salvo em *json* para a utilização na classe GeradorAtletas que recebe as informações obtidas pelas classes anteriores.

### 3.2. Analisador

Após o desenvolvimento da parte de geração de dados, o foco foi para a utilização dos dados gerados. No analisador é coletado todas as informações sobre os atletas e com isso gera os gráficos dos atletas cadastrados.

Método para gerar o gráfico com o percentual de quem acertou mais chutes de 3 pontos:

```

12 public static void gerarHtml(List<Atleta> vetor) throws IOException {
13
14     String htmlGera = "<html>\r\n" +
15         "    <html>" +
16         "<head>" +
17         "<scripttype=\"text/javascript\"src=\"https://www.gstatic.com/charts/loader.js\"></script>" +
18         "<script type=\"text/javascript\">" +
19         "    google.charts.load(\"current\", {packages:[\"corechart\"]});"+
20         "    google.charts.setOnLoadCallback(drawChart);"+
21         "    function drawChart() { "+
22         "        var data = new google.visualization.DataTable();\r\n" +
23         "        data.addColumn('string', 'Nome');\r\n" +
24         "        data.addColumn('number', 'Arremessos Acertados 3 pontos');\r\n" +
25         "        data.addColumn('number', 'Arremessos Tentados 3 pontos');\r\n" +
26         "\r\n" +
27         "        data.addRows([\r\n" + dadosGrafico(vetor) + "    ]);\r\n" +
28         "\r\n" +
29         "        var options = { "+
30         "            'title: 'Mais Acertou Chutes de 3 pts (em Quantidade)',"+
31         "            'is3D: true,'+
32         "        }";
33         "    }";
34
35         "    var chart = new
36         google.visualization.PieChart(document.getElementById('piechart_3d'))";
37         "    chart.draw(data, options);"+
38         "    }"+
39         "</script>" +
40         "</head>" +
41         "<body>" +
42         "    <div id=\"piechart_3d\" style=\"width: 900px; height: 500px;\"></div>" +
43         "</body>" +
44         "</html>"
45     ;
46
47     FileWriter arq = new
48     FileWriter("C:\\Users\\Vinicius\\Desktop\\graficoPizzaMaisChutes3pts.html");
49     PrintWriter gravarArq = new PrintWriter(arq);
50
51     gravarArq.printf(htmlGera);
52     arq.close();
53 }
54
55 private static String dadosGrafico(List<Atleta> vetor) {
56     String dados = "";
57     for (Atleta atleta:vetor) {
58         for (int i = 0;i < atleta.getLista().size() ;i++) {

```

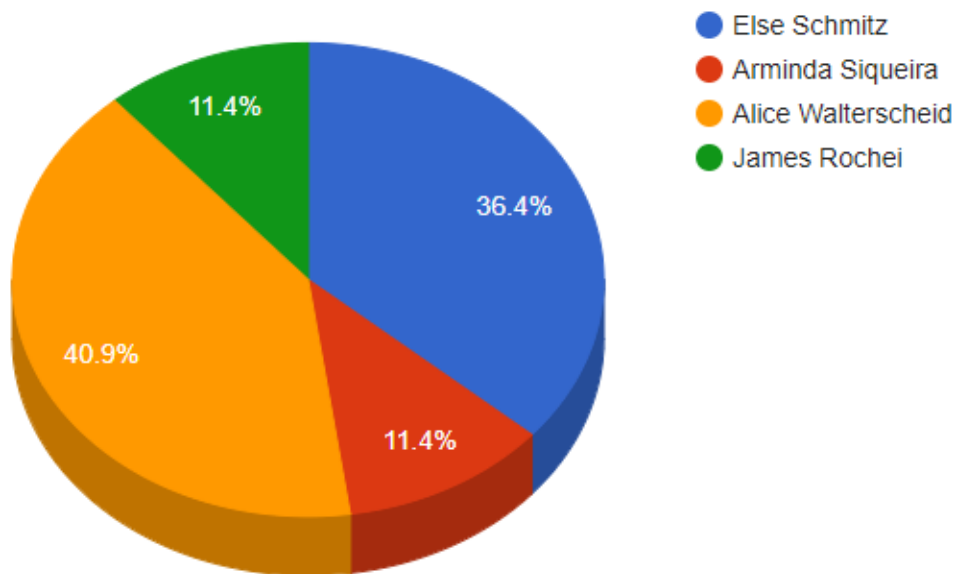
```

56         dados += "[" + atleta.getNome() + ", " +
String.valueOf(atleta.getLista().get(i).getPts3_certas()) + ", " +
String.valueOf(atleta.getLista().get(i).getPts3_tentadas()) + "],\r\n";
57
58     }
59 }
60 return dados;
61 }

```

Para gerar os gráficos foi utilizado o mesmo padrão mostrado acima, modificando apenas as informações contidas, abaixo segue resultado do gráfico gerado:

**Mais Acertou Chutes de 3 pts (em Quantidade)**



**Figura 2. Exemplo de gráfico gerado.**

### 3.3. Persistência

Na persistência o foco se deu ao modelo JSON por ser de fácil uso e atender aos requisitos do projeto, além do JSON também foi utilizado o XML. Abaixo um exemplo de alguns atletas persistidos com JSON.

```

[{"nome":"ElseSchmitz","lista":[{"Pts2_certas":3,"Pts2_tentadas":18,"Pts3_certas":16,"
Pts3_tentadas":24,"FT_certas":2,"FT_tentadas":23}]},
{"nome":"ArmindaSiqueira","lista":[{"Pts2_certas":3,"Pts2_tentadas":5,"Pts3_certas":5
,"Pts3_tentadas":12,"FT_certas":10,"FT_tentadas":24}]},
{"nome":"AliceWalterscheid","lista":[{"Pts2_certas":21,"Pts2_tentadas":21,"Pts3_certa
s":18,"Pts3_tentadas":18,"FT_certas":9,"FT_tentadas":12}]},
{"nome":"JamesRochei","lista":[{"Pts2_certas":2,"Pts2_tentadas":10,"Pts3_certas":5,"P
ts3_tentadas":8,"FT_certas":0,"FT_tentadas":23}]}]

```

#### **4. Conclusão**

Concluiu-se com o projeto BasketStats que os esportes não estão ficando para trás no quesito tecnologia e estão cada vez mais entrando no campo, com aplicativos, dispositivos e ferramentas que auxiliam no desenvolvimento do esporte, como o BasketStats que gera dados para serem analisados pelos treinadores para que os mesmos possam retirar o melhor desempenho de seus atletas.