



Instituto Federal de Ciência, Tecnologia e Educação do Sudeste de Minas Gerais  
Campus Rio Pomba

Lançamento de um projétil

Gabriel Barbosa de Castro Vital

Jonathas de Oliveira Fernandes Costa

Ariel Granato Bento

Claudio Henrique Oliveira de Andrade

Felipe Ferreira Costa de Oliveira

Rio Pomba, 09 de junho de 2019

# Sumário

1-Objetivo

2- Códigos e Funções utilizadas

3-Obstáculos

4-Conclusão

## 1- Objetivo

Visto que tínhamos um trabalho proposto, o grupo se reuniu e viu que fazer em programação na linguagem JAVA, um programa composto de algoritmos que retratam a física, com um canhão que atira um projétil em um alvo, sendo que é preciso informar sua velocidade inicial, ângulo, como parâmetro de entrada, e com uma distância aleatória (gerada no próprio programa) assim a saída é ver se o usuário acertou ou não o alvo, e sua trajetória.

## 2- Códigos e Funções utilizadas

```
private double Tempo(double v0y){
48     double resultado = 0;
49     if(v0y != 0){
50         resultado = (2*v0y)/9.8;// t = 2*v0y/g
51         return resultado;
52     }else{
53         System.out.println("ERRO - É necessario calcular o v0y antes");
54     }
55     return -1;
56 }
```

Essa função se refere ao tempo, de como calcula-lo,  $t = (2 \cdot v_{0y})/g$ .

```
private double velocidadeInicialX(double v0, double angulo){
36     angulo = this.converteParaRad(angulo);
37     double vEmX = v0 * Math.cos(angulo);// v0x = v0 * cos(angulo)
38
39     return vEmX;//O RETORNO É EM FUNÇÃO DO METODO velocidadeMedia()
40
41 }
```

```
private double velocidadeInicialY(double v0,double angulo){
43     angulo = this.converteParaRad(angulo);
44     double velocidadeEmY = v0 * Math.sin(angulo);// v0y = v0 * seno(angulo)
45     return velocidadeEmY;
46 }
```

Decompõe a velocidade inicial nas componentes, vertical e horizontal, como em física.

```

85 public double velocidadeMedia(double x, double angulo){
86     double distX = 0;
87     double vix = 0;
88     double viy = 0;
89     double tempo = 0;
90     double i = 0.1;
91     double cont = 0;
92     double distanciaCalculada = 0;
93
94     while(distanciaCalculada < x){//ESTA FUNÇÃO ACHA A VELOCIDADE POR MEIO DE TENT.
95
96         vix = this.velocidadeInicialX(i, angulo);
97         viy = this.velocidadeInicialY(i, angulo);
98         tempo = this.Tempo(viy);
99         distanciaCalculada = this.distanciaX(tempo,vix);
100         if(distanciaCalculada < x){
101             i = i +0.01;
102         }
103
104         this.setT(tempo);
105         this.setVO(i);
106         this.setVOx(vix);
107         this.setVoy(viy);
108         this.setDx(distanciaCalculada);
109         this.setVO(i);
110         return i;// i É a VELOCIDADE MEDIA COM UMA LEVE MARGEM DE ERRO
111     }

```

Essa função calcula a velocidade necessária para atingir a distância esperada por meio de tentativa, começando de 0 até achar a velocidade correta relacionada ao ângulo teta e a distância que são gerados randomicamente pelo programa

```

67 public double posicaoEmX(double intervaloDeTempo){
68     if(this.t != 0 && this.v0x != 0 ){
69         double posicaoX = 0 + this.v0x*intervaloDeTempo;// x = x0 + v0x*t
70         return posicaoX;
71     }
72     System.out.println("ERRO - É necessario calcular v0y");
73     return -1;
74 }

75 public double posicaoEmY(double intervaloDeTempo){
76     if(this.t != 0 && this.v0y != 0 ){
77         double posicaoY = (0 + this.v0y*intervaloDeTempo) - ((9.8 * (intervaloDeTempo*intervaloDeTempo))/2)
78         // y = y0 + v0y*t - (g*t^2)/2
79         return posicaoY;
80     }

```

Utiliza das formulas em física para achar a posição do alvo em x e em y para cada instante de tempo, utilizando as duas funções para o mesmo instante é possível gerar um ponto referente a posição do projétil para o instante e com vários pontos um gráfico.

```

58 public double distanciaX(double t, double v0x){
59     if(t != 0 && v0x != 0){
60         double dX = (0 + v0x) * t; // X = X0 + V0x * Tempo
61         return dX;
62     }else{
63         System.out.println("ERRO - É necessario calcular o v0x e o tempo antes");
64     }
65     return -1;
66 }

```

Calcula distância de acordo com a formula,  $X = X_0 + V_{0x} * T$ .

```

public int numeroAleatorio(int minimo, int maximo){//os parametros é o :

    Random aleatorio = new Random();
    int numeroR = aleatorio.nextInt((maximo-minimo)+1) + minimo;
    //System.out.println("O numero aleatorio gerado eh: " + numeroR);

    if(n == numeroR){
        while(n == numeroR){
            numeroR = aleatorio.nextInt((maximo-minimo)+1) + minimo; //
            //System.out.println("O numero aleatorio gerado eh: " + num
        }
    }

    this.setN(numeroR);
    return numeroR;
}

```

Gera um número aleatório, de acordo com intervalo passado(mínimo e máximo).

### 3-Obstáculos

O grande obstáculo do trabalho foi mostrar a parte gráfica, visto que em JAVA não conseguimos concluir, então foi preciso fazer que o programa elaborasse um arquivo .csv(arquivo de entrada do Excel), e aí sim pelo Excel mostraria a trajetória com toda parte gráfica.

#### 4- Conclusão

Apesar de não ter feito todo o processo dentro da própria linguagem, o trabalho agregou conhecimento tanto de física como de trabalhar a linguagem utilizando formulas da física aplicando em um jogo.