

Lista 3

Professores: Sandro, Mauricio e Eliza*Name:* Dalmo da Silva Daltohttps://github.com/Dalmomr/web-project/tree/new_branch**EXERCICIO 0:**

```
1 {
2
3 TFile *input = new TFile("DYJetsToLL.root","read");
4 TTree *t = (TTree *)input->Get("Events");
5
6 c1->cd(1);
7
8 t->MakeClass("meu");
9
10 fiz essa modifica o no .h{}
11 virtual void Loop(TTree *tree1);
12 }
13
14
15 #define meu_cxx
16 #include "meu.h"
17 #include <TH2.h>
18 #include <TStyle.h>
19 #include <TCanvas.h>
20
21 void meu::Loop(TTree* tree1)
22 {
23
24     TFile *file = new TFile("exercicio_0", "RECREATE");
25
26     if (!file || file->IsZombie()) {
27         throw std::runtime_error("Erro ao abrir o arquivo para escrita");
28     }
29
30     // Verifica se a rvore um ponteiro v lido
31     if (!tree1) {
32         throw std::runtime_error("Erro: Ponteiro de rvore inv lido");
33     }
34
35     // Define as vari veis que ser o preenchidas na TTree
36     Float_t muon_mass;
37     Float_t tau_mass;
38
39     // Associa as vari veis aos ramos da TTree
40     tree1->Branch("muon_mass", &muon_mass, "muon_mass/F");
41     tree1->Branch("tau_mass", &tau_mass, "tau_mass/F");
42
43     if (!fChain) {
44         throw std::runtime_error("Erro: fChain n o est inicializado");
45     }
46
47     Long64_t nentries = fChain->GetEntriesFast();
48     Long64_t nbytes = 0, nb = 0;
49 }
```

```

50     for (Long64_t jentry = 0; jentry < nentries; jentry++) {
51         Long64_t ientry = LoadTree(jentry);
52         if (ientry < 0) break;
53
54         nb = fChain->GetEntry(jentry);
55         nbytes += nb;
56
57         // Verifica se Muon_mass e Tau_mass t m pelo menos um elemento
58         if (Muon_mass[0] && Tau_mass[0]) {
59             // Preenche as vari veis com os valores dos ramos
60             muon_mass = Muon_mass[0];
61             tau_mass = Tau_mass[0];
62
63             // Preenche os ramos da TTree com os valores das vari veis
64             tree1->Fill();
65         } else {
66             std::cerr << "Acesso inv lido aos dados de Muon_mass ou Tau_mass na
67                 entrada " << jentry << std::endl;
68         }
69     }
70
71     .L meu.C;
72
73     meu l;
74     TTree *tree2 = new TTree("myTree", "Tree Title");
75     meu l;
76     l.Loop(tree2);
77
78     TCanvas *c1= new TCanvas("c1","c1",1000,500);
79     c1->Divide(1,2);
80     c1->cd(1);
81     tree1->Draw("muon_mass");
82     c1->cd(2);
83     tree1->Draw("tau_mass");
84 }
85

```

EXERCICIO 1:

```

1  {
2
3  TF1 * f1 = new TF1("f1","[0]*sin([1]*x)/x", -500, +500);
4
5  f1->SetParameter(0, 1); // p0 = 1
6  f1->SetParameter(1, 2); // p1 = 2
7  f1->SetLineColor(kBlue);
8
9
10 TCanvas *c1 = new TCanvas("c1", "Function Plot", 800, 600);
11
12 f1->Draw();
13
14 double functionValue = f1->Eval(1);
15 printf("a. Function value for x = 1: %.4f\n", functionValue);
16
17 double functionDerivative = f1->Derivative(1);
18 printf("b. Function derivative for x = 1: %.4f\n", functionDerivative);
19
20 double integral = f1->Integral(0, 3);
21 printf("c. Integral of the function between 0 and 3: %.4f\n", integral);

```

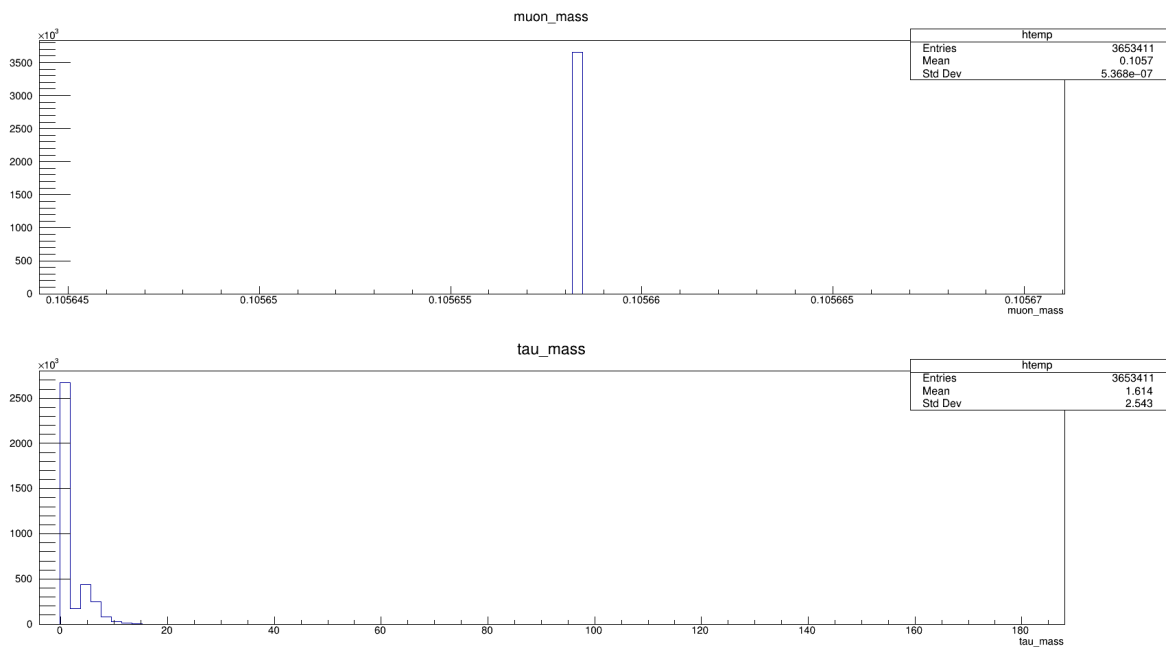


Figura 1: EXERCICIO 0

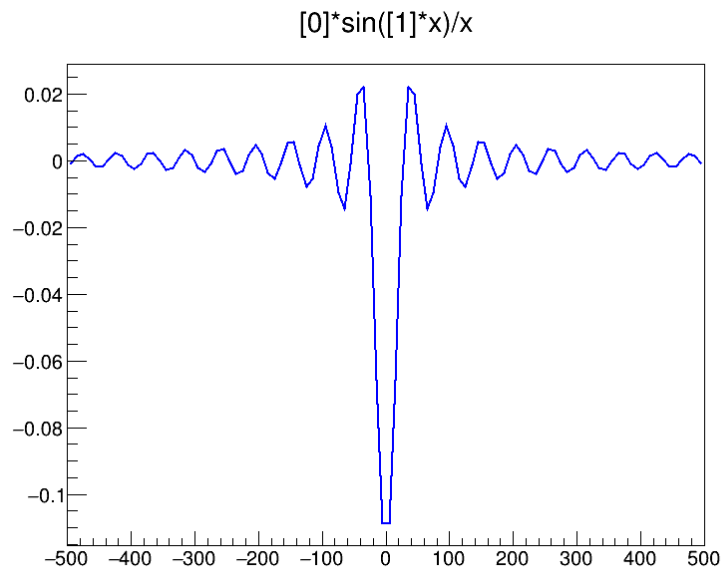


Figura 2: EXERCICIO 1

22 }

A saída dessa macro resulta em um plot mostrado acima e os seguintes resultados:

- Function value for $x = 1$: 0.9093
 - Function derivative for $x = 1$: -nan
 - Integral of the function between 0 and 3: 1.4247
- O resultado da derivada é nan porque não tem derivada naquele ponto.

EXERCICIO 2:

```

1      {
2
3      ifstream arq1;
4      ifstream arq2;
5
6      arq1.open("graphdata.txt");
7      arq2.open("graphdata_error.txt");
8
9
10     float x[10], y[10];
11     float x_1[10], ex[10], y_1[10], ey[10];
12
13     int i=0;
14
15     while(!arq1.eof() and !arq2.eof()){
16
17         arq1>>x[i]>>y[i];
18         arq2>>x_1[i]>>y_1[i]>>ex[i]>>ey[i];
19
20         cout<<x[i]<<" "<<y[i]<<endl;
21
22         cout<<x_1[i]<<" "<<y_1[i]<<" "<<ex[i]<<" "<<ey[i]<<endl;
23
24
25         i++;
26
27     }
28
29
30     TCanvas *c1= new TCanvas("c1","c1",1000,500);
31
32     c1->Divide(2,1);
33     c1->cd(1);
34     TGraph *t= new TGraph(10,x,y);
35     t->Draw();
36     t->SetTitle("Plot sem barra de erros");
37
38     c1->cd(2);
39     TGraphErrors *t1= new TGraphErrors(10,x_1,y_1,ex,ey);
40     t1->Draw();
41     t1->SetTitle("Plot com barra de erros");
42
43 }
```

EXERCICIO 3:

```

1      {
2
3      TRandom *t= new TRandom;
```

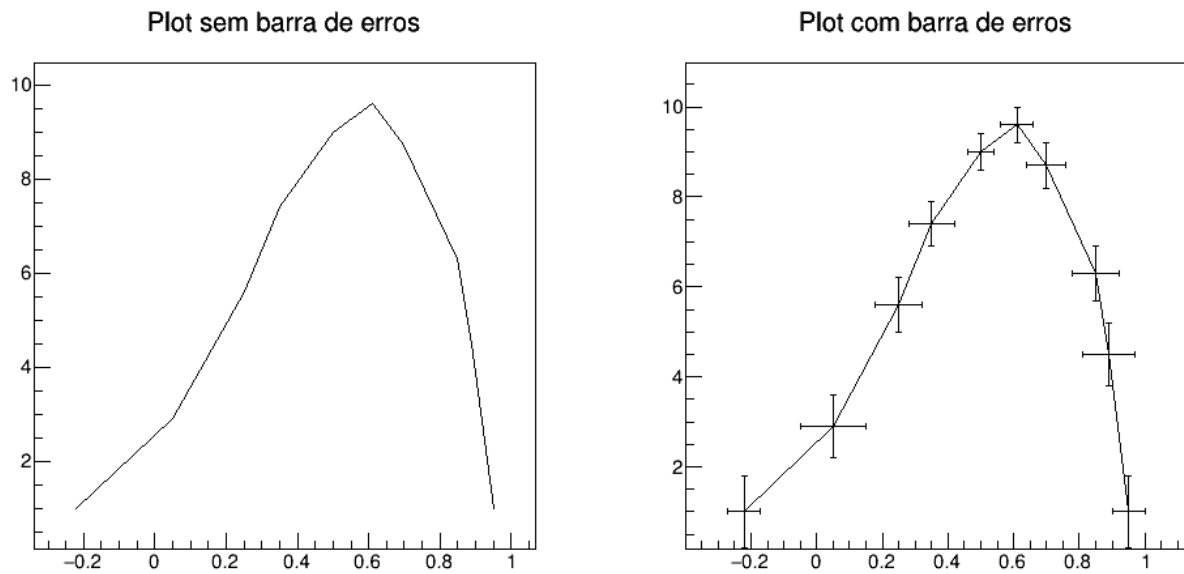


Figura 3: EXERCICIO 2

```

4 TH1F *h1= new TH1F("Fit gauss","Fit gauss",50,0,10);
5
6 for(int i=0; i<10000;i++){
7
8     float random_gauss= t->Gaus(5,2);
9
10    h1->Fill(random_gauss);
11 }
12
13 h1->Draw();
14 h1->Fit("gaus");
15 gStyle->SetOptFit(1111);
16 }

```

EXERCICIO 4:

```

1 {
2 TCanvas *c1= new TCanvas("c1","c1",1500,800);
3
4 TFile *input = new TFile("tree.root","read");
5 TTree *t = (TTree *)input->Get("tree1");
6
7 gStyle->SetOptFit(1111);
8
9 TF1 *gauss = new TF1("g1","gaus",0,10);
10
11
12 c1->Divide(3,1);
13 c1->cd(1);
14 t->Draw("ebeam>>h1");
15 h1->Fit(gauss);
16
17 float mean=gauss->GetParameter(1);
18 float sigma=gauss->GetParameter(2);
19
20 c1->cd(2);

```

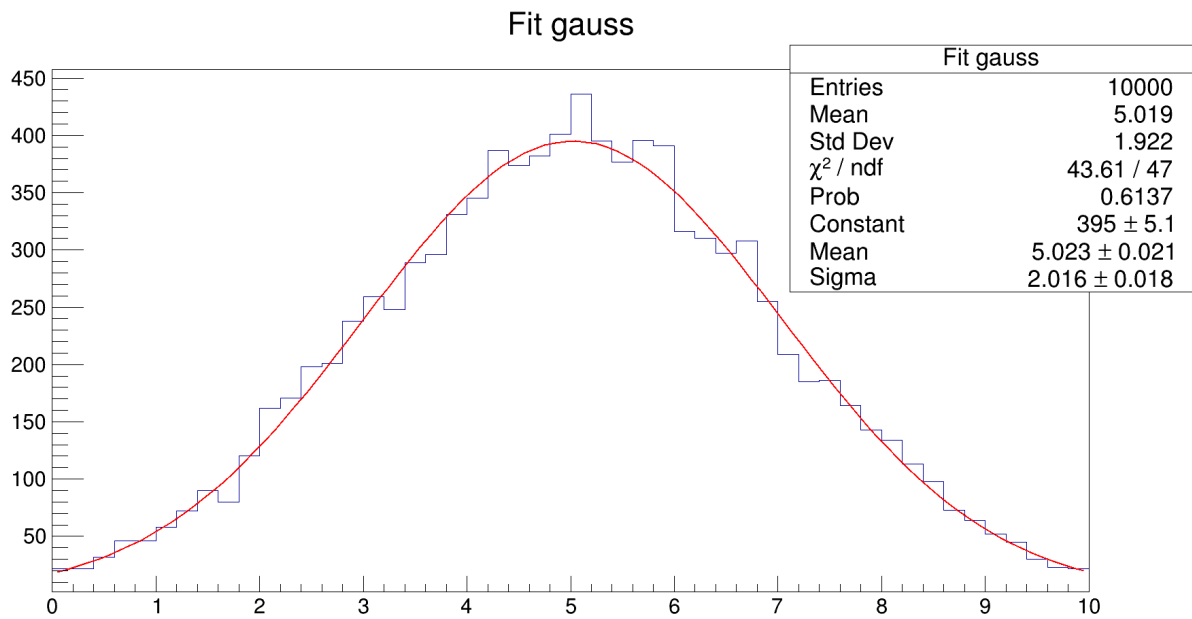


Figura 4: EXERCICIO 3

```

21
22 char cut[20];
23
24 sprintf(cut,"ebeam>%f",0.2 + mean);
25
26 t->Draw("px+py+pz>>h2",cut,"");
27
28
29 h2->Fit(gauss);
30 h2->SetTitle("px+py+pz (ebeam>0.2 + Mean_{ebeam})");
31
32 c1->cd(3);
33
34
35 t->Draw("px+py+pz>>h3","", "");
36
37 h3->Fit(gauss);
38 h3->SetTitle("px+py+pz");
39
40 c1->Print("exercicio4.png","png");
41 }

```

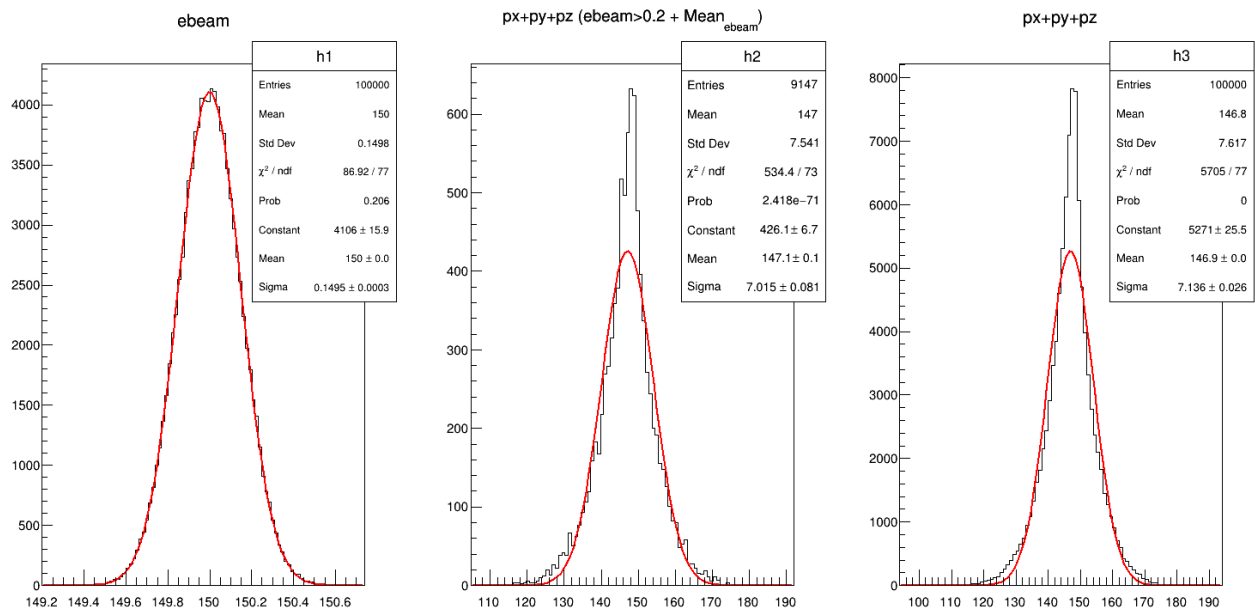


Figura 5: EXERCICIO 4