

```

1  #Curso de Extensão:
2  #Lógica de programação aplicada à criação e análise da
   forma
3  #UFBa - Faculdade de Arquitetura - 2017.1
4  #Coordenador: Arivaldo Leão de Amorim
5  #Professor: Fernando Ferraz Ribeiro
6  #Exemplo treliça a partir do(s) eixos dos banzos
7
8  #Importando Módulos
9  import rhinoscriptsyntax as rs
10
11 #entradas
12
13
14 #Saidas
15 Cordas = []
16 Diag = []
17 Bsup = []
18 Binf = []
19
20
21
22 # versor Z auxiliar
23 vrZ= rs.VectorCreate([0,0,1],[0,0,0])
24
25 # -> Tratando variáveis
26 # Se o numero de divisões não foi lido...
27 if not N1:
28     # ... considerar N1 = 10
29     N1 = 10
30
31 # se a entrada C2 contem um valor numérico
32 if type(C2) == float:
33     # criar um offset da curva C1 na distância C2
34     # na direção oposta ao versor Z auxiliar
35     C2 = rs.OffsetCurve(C1,vrZ,C2)
36
37 if OfSup:
38
39     C1 = rs.OffsetCurve(C1,vrZ,OfSup)
40
41 if OfInf :
42
43     C2 = rs.OffsetCurve(C2,vrZ,-OfInf)
44
45 # --> Métodos
46
47 # -> Método que desenha as cordas verticais
48 def EixosVert(lp1, lp2): #recebe duas listas de pontos
49     ev = []

```

```

50     # Cria lista de pontos entre as lista de pontos
51     for i in range(len(lp1)):
52         lAux = rs.AddLine(lp1[i],lp2[i])
53         ev.append(lAux)
54     return ev
55
56 # -> Fim do metodo
57
58
59 # -> metodo que desenha o eixo das diagonais
60 def EixoCordas(banSup, banInf, num):
61     # lista de pontos que dividem o banzo superior
62     # em num segmentos
63     pointsC1 = rs.DivideCurve(banSup,num)
64     # lista de pontos que dividem o banzo inferior
65     # em num segmentos
66     pointsC2 = rs.DivideCurve(banInf,num)
67
68     cordas = EixosVert(pointsC1, pointsC2 )
69
70
71
72
73     # lista dos pontos de indice par do banzo sup
74     banSupPt = pointsC1[0::2]
75     # lista dos pontos de indice impar do banzo inf
76     banInfPt = pointsC2[1::2]
77
78     # lista vazia com tamanho igula a banSupPt + banInfPt
79     diagPt = [None]*(len(banSupPt)+len(banInfPt))
80
81     # intercalando as listas
82     diagPt[::2] = banSupPt
83     diagPt[1::2] = banInfPt
84
85     # desenha polyline entre os pontos intercalados
86     eixoDiag = rs.AddPolyline(diagPt)
87     # retorna os eixos
88     return [eixoDiag] , cordas
89 # -> Fim do metodo
90
91 # --> Função Principal
92 Diag, Cordas = EixoCordas(C1, C2, N1)
93
94 Bsup = C1
95
96 Binf = C2
97
98 # --> Fim

```