

TOOLBOX

SOLVER DE PROGRAMAÇÃO LINEAR

Desenvolvido por :
- João Victor Duarte Viana



REQUISITOS

➤ **SISTEMA OPERACIONAL :**

- **LINUX ;**
- **WINDOWS ;**
- **MAC ;**

➤ **PYTHON 3**

PRIMEIRO ETAPA – AMBIENTE VIRTUAL

- **ABRA O TERMINAL/PROMPT;**
- **VÁ PARA O DIRETÓRIO DE DESTINO;**
- **DIGITE: *PYENV SOLVERPL*;**
- **UMA OUTRA FORMA, NO WINDOWS, É DIGITANDO: *PYTHON -M VENV SOLVERPL*;**
- **LOGO APÓS CRIAR O AMBIENTE, ATIVE-O DIGITANDO: *SCRIPTS/ACTIVATE.BAT*.**

SEGUNDA ETAPA – INSTALANDO O PULP

- **MÓDULO PARA MODELAR E SOLUCIONAR PROBLEMAS DE PL;**
- **ARQUIVOS LP;**

➤ **COM O AMBIENTE VIRTUAL ATIVADO, INSTALE O MÓDULO PULP DIGITANDO: *PIP INSTALL PULP***

TERCEIRA ETAPA – MODELAR O PROBLEMA

```
In [1]: import sys  
        from pulp import *
```

```
In [2]: def createProblem(name,is_max):  
        return LpProblem(name,(LpMaximize if is_max else LpMinimize))
```

```
In [*]: name = input ("Qual o nome do seu problema linear? ")  
        is_max = True if input ("Você quer máximizar esse problema? [y]") == "y" else False  
  
        problem = createProblem(name,is_max)
```

...

```
In [7]: def createVariables():
        num_variables = int(input("Quantas variáveis irá precisar para este problema? "))
        vetor = []
        for count in range(num_variables):
            print(count + 1, "Variável: ")
            name_variable = input("Qual o seu nome? ")
            low_Bound = int(input("Qual seu limite mínimo? "))
            up_Bound = int(input("Qual seu limite máximo? "))
            vetor.append(LpVariable(name_variable, low_Bound, up_Bound))
        return vetor
```

```
In [8]: variables = createVariables()
```

...

```
In [9]: def createFuncObj(variables, problem):
        print("Crie agora a função objetivo.")
        func = 0
        for count in range(len(variables)):
            value = float(input("Qual o coeficiente da variável " + str(variables[count]) + " ? "))
            func += value*variables[count]
        problem += func, "Total de custos"
```

```
In [10]: createFuncObj(variables, problem)
```

...


```

In [11]: def createRestriction(variables,problem,number):
    print("Crie a",number,"º restrição.")
    name_restriction = input("Qual o nome desta restrição? ")
    restrict = 0
    for count in range(len(variables)):
        value = float(input("Qual o coeficiente da variável " + str(variables[count]) + " ? "))
        restrict += value*variables[count]

    correct = False

    while correct == False:
        logic = input("Qual é a inequação da " + str(count) + "º restrição (" + name_restriction + ")? (>= ou <=) ")
        value_b = float(input("Qual é o valor desta restrição? "))
        if logic == ">=":
            problem += restrict >= value_b,name_restriction
            correct = True
        elif logic == "<=":
            problem += restrict <= value_b,name_restriction
            correct = True
        else:
            print("Opção inválida! Por favor, digite novamente.")

```

```

In [14]: num_restriccion = int(input("Quantas restrições você quer para este problema?"))
    for count in range(num_restriccion):
        createRestriction(variables,problem,count+1)

```

...

```
In [15]: filepath = name + ".lp"
        problem.writeLP(filepath)
```

```
In [16]: problem.solve()
```

...

```
In [17]: print("Status : ", LpStatus[problem.status])
```

...

```
In [18]: for variable in problem.variables():
        print(variable.name,variable.varValue)
```

...

```
In [19]: print("Custo total :",value(problem.objective))
```

...

EXEMPLO

- **UMA PEQUENA FÁBRICA DE CALÇADOS PRODUZ SAPATOS E SANDÁLIAS. AS SANDÁLIAS SÃO VENDIDAS NO MERCADO POR \$ 22,00 E OS SAPATOS POR \$48,00. PARA A FABRICAÇÃO DE UMA SANDÁLIAS GASTA-SE 0,30 M² DE COURO, LEVAM-SE TRÊS HORAS NO CORTE E COSTURA E É UTILIZADA UMA PESSOA PARA O ACABAMENTO, DETALHES FINAIS E EMBALAGEM, E NA CONFEÇÃO DE UM SAPATO GASTA-SE 0,50 M² DE COURO, QUATRO HORAS NO CORTE E COSTURA E DUAS PESSOAS SÃO UTILIZADAS PARA O ACABAMENTO, DETALHES FINAIS E EMBALAGEM. A EMPRESA CONTA, DIARIAMENTE, COM 15 M² DE COURO, 120 HORAS DE TRABALHO E 15 PESSOAS. DESEJA-SE FORMULAR O MODELO DE PROGRAMAÇÃO LINEAR QUE MAXIMIZA A RECEITA DIÁRIA DA FÁBRICA DE CALÇADOS.**

SOLUÇÃO

- **VARIÁVEIS DE DECISÃO :**
 - **X_1 = QUANTIDADE DIÁRIA A SER PRODUZIDA DE SANDÁLIAS;**
 - **X_2 = QUANTIDADE DIÁRIA A SER PRODUZIDA DE SAPATOS;**
- **FUNÇÃO OBJETIVO :**
 - **$\text{MAX. } Z = 22X_1 + 48X_2$**
- **RESTRIÇÕES :**
 - **RESTRIÇÃO DE MATERIAL : $0,30X_1 + 0,5X_2 \leq 15$**
 - **RESTRIÇÃO DE HORAS DE TRABALHO: $3X_1 + 4X_2 \leq 120$**
 - **RESTRIÇÃO DE MÃO-DE-OBRA : $X_1 + 2X_2 \leq 15$**

```
In [1]: import sys
        from pulp import *
```

```
In [2]: def createProblem(name,is_max):
        return LpProblem(name,(LpMaximize if is_max else LpMinimize))
```

```
In [3]: name = input ("Qual o nome do seu problema linear? ")
        is_max = True if input ("Você quer maximizar esse problema? [y]") == "y" else False

        problem = createProblem(name,is_max)
```

Qual o nome do seu problema linear? Fabrica de Calçados
Você quer maximizar esse problema? [y]y

```
In [4]: def createVariables():  
    num_variables = int(input("Quantas variáveis irá precisar para este problema? "))  
    vetor = []  
    for count in range(num_variables):  
        print(count + 1, "Variável: ")  
        name_variable = input("Qual o seu nome? ")  
        low_Bound = int(input("Qual seu limite mínimo? "))  
        up_Bound = int(input("Qual seu limite máximo? "))  
        vetor.append(LpVariable(name_variable, low_Bound, up_Bound))  
    return vetor
```

```
In [5]: variables = createVariables()
```

```
Quantas variáveis irá precisar para este problema? 2  
1 Variável:  
Qual o seu nome? x1  
Qual seu limite mínimo? 0  
Qual seu limite máximo? 100  
2 Variável:  
Qual o seu nome? x2  
Qual seu limite mínimo? 0  
Qual seu limite máximo? 100
```

```
In [6]: def createFuncObj(variables,problem):  
        print ("Crie agora a função objetivo.")  
        func = 0  
        for count in range(len(variables)):  
            value = float( input("Qual o coeficiente da variável " + str(variables[count]) + " ? "))  
            func += value*variables[count]  
        problem += func,"Total de custos"
```

```
In [7]: createFuncObj(variables,problem)
```

Crie agora a função objetivo.
Qual o coeficiente da variável x1 ? 22
Qual o coeficiente da variável x2 ? 48

```
In [9]: num_restriccion = int(input("Quantas restrições você quer para este problema?"))
        for count in range(num_restriccion):
            createRestriction(variables,problem,count+1)
```

```
Quantas restrições você quer para este problema?3
Crie a 1 º restrição.
Qual o nome desta restrição? Material
Qual o coeficiente da variável x1 ? 0.3
Qual o coeficiente da variável x2 ? 0.5
Qual é a inequação da 1º restrição (Material)? (>= ou <=) <=
Qual é o valor desta restrição? 15
Crie a 2 º restrição.
Qual o nome desta restrição? Horas de Trabalho
Qual o coeficiente da variável x1 ? 3
Qual o coeficiente da variável x2 ? 4
Qual é a inequação da 1º restrição (Horas de Trabalho)? (>= ou <=) <=
Qual é o valor desta restrição? 120
Crie a 3 º restrição.
Qual o nome desta restrição? Mão-de-obra
Qual o coeficiente da variável x1 ? 1
Qual o coeficiente da variável x2 ? 2
Qual é a inequação da 1º restrição (Mão-de-obra)? (>= ou <=) <=
Qual é o valor desta restrição? 15
```



```
In [10]: filepath = name + ".lp"
problem.writeLP(filepath)
```

```
In [11]: problem.solve()
```

```
Out[11]: 1
```

```
In [12]: print("Status : ", LpStatus[problem.status])

Status : Optimal
```

```
In [13]: for variable in problem.variables():
          print(variable.name,variable.varValue)

x1 0.0
x2 7.5
```

```
In [14]: print("Custo total :",value(problem.objective))

Custo total : 360.0
```

```
1  \* FabricaDeCalçados *\
2  Maximize
3  Total_de_custos: 22 x1 + 48 x2
4  Subject To
5  HorasDeTrabalho: 3 x1 + 4 x2 <= 120
6  MaoDeObra: x1 + 2 x2 <= 15
7  Material: 0.3 x1 + 0.5 x2 <= 15
8  Bounds
9  x1 <= 100
10 x2 <= 100
11 End|
12
```

RECURSOS

- **PULP – DOCUMENTAÇÃO:**
[HTTPS://PYTHONHOSTED.ORG/PULP/MAIN/INSTALLING_PULP_AT_HOME.HTML](https://pythonhosted.org/pulp/main/installing_pulp_at_home.html)
- **GITHUB :** [HTTPS://GITHUB.COM/JOAODEMARIA/TOOLBOXSOLVER](https://github.com/joaodemaria/toolboxsolver)