

# Pesquisa Operacional Para Computação

2022.2



Universidade Federal  
de São João del-Rei

- Nome do Aluno:  
João Antônio Santos Carvalho
- Nome do Professor:  
Álvaro Rodrigues Pereira Júnior
- Curso:  
Ciências da Computação

## 1 Introdução

O programa começa mostrando o usuário quais são as variáveis e as restrições do programa. As variáveis estão registradas em uma planilha, a mesma que o programa está retirando as informações de preço dos itens.

```
Variáveis:
x[1] = Coca
x[3] = Guaraná
x[2] = Fanta
x[4] = Pepsi
x[5] = H2O
x[6] = Brahma
x[7] = Skol
x[8] = Heineken
x[9] = Suco
x[10] = Água

Restrições:
1) Tem que ter no mínimo um refrigerante, uma cerveja, água e suco
2) Só pode um tipo de cerveja
3) Só pode um tipo de refrigerante
4) O preço da compra deve ser menor ou igual ao valor total disponibilizado pelo usuário
5) Se for festa de criança o número de refrigerante deve ser maior que o de cerveja
6) Se for festa de adulto o número de cerveja deve ser maior que o de refrigerante
7) A soma da quantidade de água e suco deve ser no máximo a metade da soma de refrigerante e cerveja
```

## 2 Entrada de Dados

A entrada de dados acontece de duas maneiras, por meio da planilha de itens e preços e por meio de perguntas ao usuário

	A	B	C	D	E	F	G	H	I	J	K	L
1	Fornecedor	Coca	Guaraná	Fanta	Pepsi	H2O	Brahma	Skol	Heineken	Suco	Água	Tempo (dias)
2	Fornecedor	R\$ 12,00	R\$ 13,00	R\$ 12,00	R\$ 10,00	R\$ 8,00	R\$ 5,00	R\$ 5,00	R\$ 8,00	R\$ 4,00	R\$ 5,00	1
3	Fornecedor	R\$ 11,00	R\$ 10,00	R\$ 11,00	R\$ 10,00	R\$ 7,00	R\$ 4,00	R\$ 4,00	R\$ 5,00	R\$ 4,00	R\$ 4,00	3
4	Fornecedor	R\$ 12,00	R\$ 9,00	R\$ 8,00	R\$ 11,00	R\$ 8,00	R\$ 5,00	R\$ 3,00	R\$ 3,00	R\$ 3,00	R\$ 2,00	5
5	Fornecedor	R\$ 13,00	R\$ 8,00	R\$ 9,00	R\$ 8,00	R\$ 9,00	R\$ 6,00	R\$ 2,00	R\$ 4,00	R\$ 5,00	R\$ 1,00	6

```
Qual fornecedor você vai querer contratar?
(1)Fornecedor 1: entregará a compra em 1 dia
(2)Fornecedor 2: entregará a compra em 3 dias
(3)Fornecedor 3: entregará a compra em 5 dias
(4)Fornecedor 4: entregará a compra em 6 dias
-> 1
Qual o tipo de evento você irá fazer?
(1) Festa de Criança
(2) Festa de Adulto
-> 1
Quanto, no máximo, você quer gastar para compras as bebidas?
-> 1
```

### 3 Funcionamento

O código irá coletar esses dados e fazer as contas, dependendo das escolhas e entradas do usuário. As restrições variam de acordo com o fornecedor que o usuário escolher

```
# Adicionar restrições de acordo com o fornecedor
match fornecedor:
    case 1:
        model.addConstraint(12 * x[1] + 13 * x[2] + 12 * x[3] + 10 * x[4] + 8 *
                             x[5] + 5 * x[6] + 5 * x[7] + 8 * x[8] + 4 * x[9] + 5 * x[10] <= valormax, 'restricao_0')
        model.addConstraint(12 * x[1] + 13 * x[2] + 12 * x[3] + 10 * x[4] + 8 *
                             x[5] + 5 * x[6] + 5 * x[7] + 8 * x[8] + 4 * x[9] + 5 * x[10] >= valormax / 2, 'restricao_1')
        model.addConstraint(x[1] + x[2] + x[3] + x[4] +
                             x[5] >= 1, 'restricao_2')
        model.addConstraint(x[6] + x[7] + x[8] >= 1, 'restricao_3')
        model.addConstraint(x[9] >= 1, 'restricao_4')
        model.addConstraint(x[10] >= 1, 'restricao_5')
        model.addConstraint(x[9] + x[10] <= (x[1] + x[2] +
                                             x[3] + x[4] + x[5] + x[6] + x[7] + x[8]) / 2, 'restricao_6')
        if (festa == 1):
            # criança
            model.addConstraint(x[1] + x[2] + x[3] + x[4] +
                                x[5] >= x[6] + x[7] + x[8], 'restricao_7')
        elif (festa == 2):
            # adulto
            model.addConstraint(x[1] + x[2] + x[3] + x[4] +
                                x[5] <= x[6] + x[7] + x[8], 'restricao_7')
        model.addConstraint(x[1] <= 20, 'restricao_a1')
        model.addConstraint(x[2] <= 20, 'restricao_a2')
        model.addConstraint(x[3] <= 20, 'restricao_a3')
        model.addConstraint(x[4] <= 20, 'restricao_a4')
        model.addConstraint(x[5] <= 20, 'restricao_a5')
        model.addConstraint(x[9] <= 20, 'restricao_a9')
        model.addConstraint(x[6] <= 20, 'restricao_a6')
        model.addConstraint(x[7] <= 20, 'restricao_a7')
        model.addConstraint(x[8] <= 20, 'restricao_a8')
        model.addConstraint(x[10] <= 20, 'restricao_a10')
        # Função Objetiva
        model.setObjective(12 * x[1] + 13 * x[2] + 12 * x[3] + 10 * x[4] + 8 *
                           x[5] + 5 * x[6] + 5 * x[7] + 8 * x[8] + 4 * x[9] + 5 * x[10])
    case _:
        pass
```

Depois de fazer as contas, irá mostrar o resultado. A saída representa o valor ótimo, ou seja, a melhor quantidade de cada item de acordo com a escolha do usuário.

```
At line 2 NAME          MODEL
At line 3 ROWS
At line 23 COLUMNS
At line 92 RHS
At line 111 BOUNDS
At line 112 ENDATA
Problem MODEL has 18 rows, 10 columns and 58 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Presolve determined that the problem was infeasible with tolerance of 1e-08
Analysis indicates model infeasible or unbounded
0 Obj -0 Primal inf 4.038461 (5) Dual inf 81.999999 (10)
0 Obj -0 Primal inf 4.038461 (5) Dual inf 1.6307692e+11 (10)
6 Obj 17 Primal inf 2.230769 (2)
6 Obj 17 Primal inf 2.230769 (2)
6 Obj 17 Primal inf 2.230769 (2)
6 Obj 17 Primal inf 2.230769 (2)
6 Obj 17 Primal inf 2.230769 (2)
6 Obj 17 Primal inf 2.230769 (2)
Primal infeasible - objective value 17
PrimalInfeasible objective 17 - 6 iterations time 0.012

Result - Linear relaxation infeasible

Enumerated nodes:      0
Total iterations:      0
Time (CPU seconds):    0.03
Time (Wallclock Seconds): 0.03

Option for printingOptions changed from normal to all
Total time (CPU seconds): 0.03 (Wallclock seconds): 0.03

x= {1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0, 5: 1.0, 6: 1.0, 7: 0.0, 8: 0.0, 9: 1.0, 10: 0.0}
```

## 4 Próximo passo

O próximo passo para aprimorar o programa, será a colocação de variáveis dinâmicas, ou seja, coletar os dados de uma planilha que o usuário pode editar.