

Problema da Dieta

Daniel Augusto
Julia Machado
Vitor Lima



Descrição do problema



O problema da dieta consiste em **selecionar um conjunto de alimentos** que satisfaçam um conjunto de **necessidades nutricionais diárias** a um **custo mínimo**.

Para isso, foram considerados os alimentos mais usualmente consumidos na região da cidade de Belo Horizonte – MG, como por exemplo, arroz, feijão, frango, ovo, peixe, carne vermelha, verduras, legumes e frutas; o custo destes alimentos em um mercado da cidade; e as recomendações máximas e mínimas de ingestão diária de cada nutriente (como carboidrato, proteína, vitaminas e minerais) para atletas.

Revisão da literatura

Diversos estudos aplicaram a técnica de otimização para resolver problemas relacionados ao custo mínimo de uma dieta balanceada.

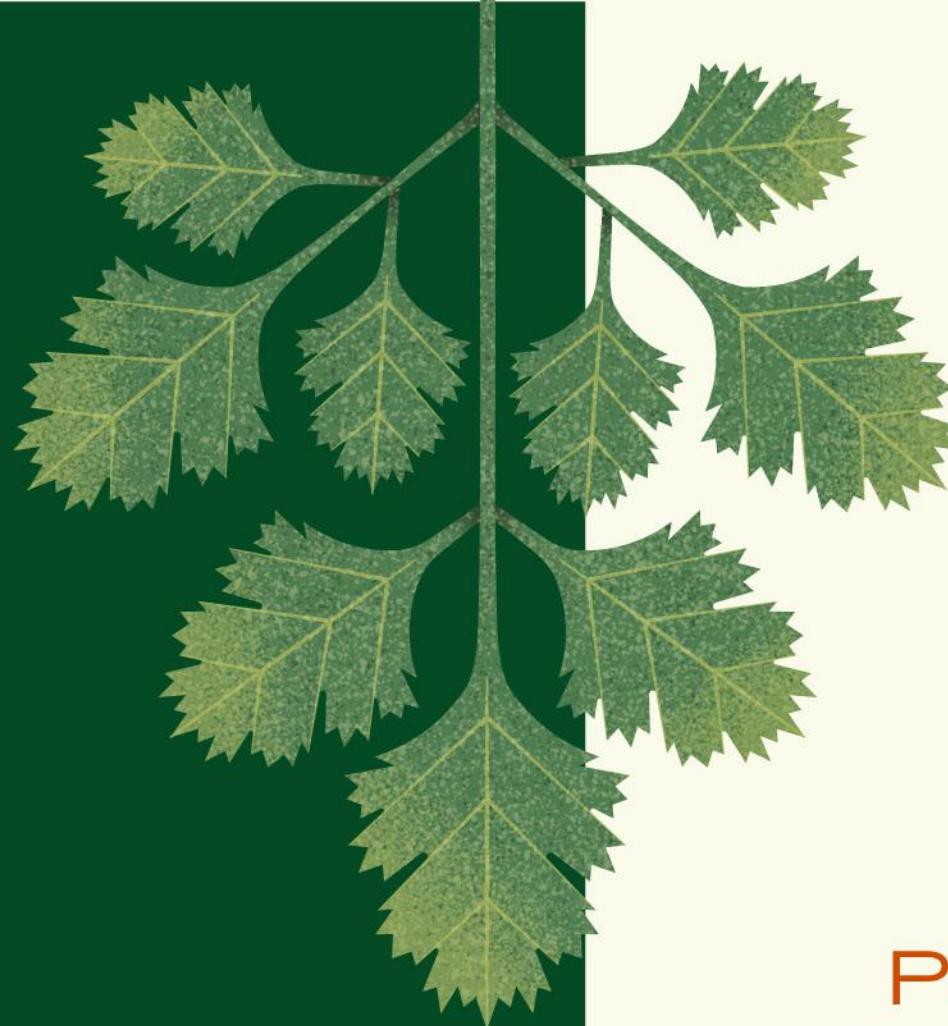
[1] Estudo de Danilo Oliveira, Ana Clara Borges e Vinicius Vieira

- Programação Linear aplicada para minimizar o custo da dieta diária de idosos em Monte Carmelo - MG.
- Utilização do software LINGO.
- Custo diário da dieta: R\$6,92, muito abaixo da média local.
- Limitação: ausência de restrições de variedade alimentar e repetição.

[2] Trabalho de Rafaela Priscila Cruz Moreira e Prof^a Dr^a. Elizabeth Fialho Wanner

- Modelos de Programação Linear para cardápios escolares, alinhados ao PNAE.
- Software: IBM CPLEX Optimizer.
- Restrições: nutricionais, qualitativas (composição, cor, consistência) e incentivo a alimentos minimamente processados.
- Resultados: cardápios nutricionalmente adequados, viáveis economicamente e culturalmente alinhados.
- Desafio: complexidade das restrições e dependência de bases de dados atualizadas.

Modelo de otimização



Variáveis:

$X_{i,j,k}$ Número de porções do alimento i consumidas na refeição k do dia j

$Z_{i,j,k}$ Variável binária que indica se o alimento i foi consumido na refeição k do dia j

$Y_{i,j}$ Variável binária que indica se o alimento i foi consumido no dia j

Parâmetros:

C_i Custo por porção do alimento i

$N_{i,p}$ Quantidade do nutriente p presente em uma porção do alimento i

$N_{p,min}, N_{p,max}$ Quantidade mínima e máxima de nutrientes p por dia

$A_{i,g}$ Indica se o alimento i pertence ao grupo g

$G_{g,min}$ Quantidade mínima de porções do grupo alimentar g por dia

$P_{i,max}$ Quantidade máxima de porções do alimento i por dia

$D_{i,max}$ Dias máximos que o alimento i pode ser consumido na semana

Função objetivo e restrições

$$\text{Min } \sum_i \sum_j \sum_k c_{i,j,k} \cdot x_{i,j,k}$$

Minimizar o custo total da compra

$$N_{p,min} \leq \sum_i \sum_k n_{i,p} \cdot x_{i,j,k} \leq N_{p,max}, \forall j \forall p$$

Cada nutriente p deve ter uma quantidade mínima e máxima diaria

$$\sum_{i \in g} \sum_k x_{i,j,k} \geq G_{g,min}, \forall j \forall g$$

Mínimo de porções para cada grupo g por dia

$$\sum_k x_{i,j,k} \leq Y_{i,j} \cdot P_{i,max}, \forall j \forall i$$

Máximo de porções de cada alimento i por dia



Função objetivo e restrições

$$\sum_j Y_{i,j} \leq D_{i,max}, \forall i$$

Quantidade máxima de dias que um alimento pode ser consumido em uma semana

$$\sum_k Z_{i,j,k} \leq 2 \cdot Y_{i,j}, \forall i \forall j$$

Um alimento i pode ser usado em no máximo 2 das k refeições diárias

$$\sum_i \sum_k X_{i,j,k} \leq 50, \forall j$$

Límite total de 5kg de alimento por dia (50 porções)



Função objetivo e restrições

$$\sum_i X_{i,j,k} \geq 1, \forall j \forall k$$

Em cada refeição pelo menos uma porção de alimento deve ser consumida
(impedir que a refeição fique sem alimento)

$$\sum_k X_{i,j,k} \geq Y_{i,j}, \forall i \forall j$$

Amarrar X e Y de tal forma que ativamos $Y_{i,j}$ somente se usamos uma quantidade maior que 0 para o alimento i em alguma refeição k desse dia j.

$$M \cdot Z_{i,j,k} \geq X_{i,j,k}, \forall i \forall j \forall k$$

M sendo um valor suficientemente grande garante que se $x > 0$ o Z está ativado.

$$X_{i,j,k} \geq Z_{i,j,k}, \forall i \forall j \forall k$$

Garantir que $Z_{i,j,k}$ seja 0 se $X_{i,j,k}$ for 0



Domínio

$X_{i,j,k} \in \mathbb{Z}^+, \forall i, \forall j, \forall k$ As porções devem ser inteiras e não negativas

$Z_{i,j,k} \in \{0, 1\}, \forall i, \forall j, \forall k$ Se o alimento i foi consumido no dia j na refeição k

$Y_{i,j} \in \{0,1\}, \forall i, \forall j$ Se o alimento i foi consumido no dia j

$C_i \in \mathbb{R}^+, \forall i$ Custo por porção de alimento i

$N_{i,p} \in \mathbb{R}^+, \forall i, \forall j, \forall k$ Quantidade de nutriente p no alimento i

$N_{p,min}, N_{p,max} \in \mathbb{R}^+, N_{p,min} < N_{p,max}, \forall p$ Quantidade mínima e máxima de cada nutriente p

$A_{i,g} \in \{0,1\}, \forall i \forall g$ Se o alimento i pertence ao grupo g

$G_{g,min} \in \mathbb{Z}^+, \forall g$ Quantidade mínima de porções para o grupo g

$P_{i,max} \in \mathbb{Z}^+, \forall i$ Quantidade máxima de porções do alimento i

$D_{i,max} \in \mathbb{Z}^+$ Dias máximos na semana que o alimento i pode ser consumido

$i,j,k,p,g \in \mathbb{Z}^+$ Alimento i, dia j, refeição k, nutriente p, grupo g são inteiros





Dados

Utilizamos por padrão uma dieta para um atleta do sexo masculino em torno dos 25 anos que treina de 3 a 5 horas diariamente

- **Calorias:** 3.000 a 5.000 kcal/dia
- **Carboidratos:** 450–900 g/dia
- **Proteínas:** 120–150 g/dia
- **Gorduras:** 70–120 g/dia
- **Ferro:** 8-18 mg/dia
- **Magnésio:** 400-420 mg/dia
- **Vitamina C:** 75-90 mg/dia
- **Zinco:** 11-16 mg/dia
- **Sódio:** 1.500-6.000 mg/dia

As recomendações de ingestão de nutrientes foram obtidas a partir dos sites Medicine LibreTexts e American Sports and Fitness Association



Os dados nutricionais dos alimentos foram obtidos a partir da Tabela Brasileira de Composição de Alimentos –TACO

Tabela 1. Composição de alimentos por 100 gramas de parte comestível: Centesimal, minerais, vitaminas e colesterol

Número do Alimento	Descrição dos alimentos	Umidade (%)	Energia		Proteína (g)	Lipídeos (g)	Colesterol (mg)	Carbohidrato (g)	Fibra Alimentar		
			(kcal)	(kJ)					Cinzas (g)	Cálcio (mg)	Magnésio (mg)
Cereais e derivados											
1	Arroz, integral, cozido	70,1	124	517	2,6	1,0	NA	25,8	2,7	0,5	5
2	Arroz, integral, cru	12,2	360	1505	7,3	1,9	NA	77,5	4,8	1,2	8
3	Arroz, tipo 1, cozido	69,1	128	537	2,5	0,2	NA	28,1	1,6	0,1	4
4	Arroz, tipo 1, cru	13,2	358	1497	7,2	0,3	NA	78,8	1,6	0,5	4
5	Arroz, tipo 2, cozido	68,7	130	544	2,6	0,4	NA	28,2	1,1	0,1	3
6	Arroz, tipo 2, cru	13,2	358	1498	7,2	0,3	NA	78,9	1,7	0,4	5
7	Aveia, flocos, crua	9,1	394	1648	13,9	8,5	NA	66,6	9,1	1,8	48
8	Biscoito, doce, maisena	3,2	443	1853	8,1	12,0	NA	75,2	2,1	1,5	54
9	Biscoito, doce, recheado com chocolate	2,2	472	1974	6,4	19,6	Tr	70,5	3,0	1,3	27
10	Biscoito, doce, recheado com morango	2,7	471	1971	5,7	19,6	Tr	71,0	1,5	1,0	36
11	Biscoito, doce, wafer, recheado de chocolate	1,2	502	2102	5,6	24,7	Tr	67,5	1,8	1,1	23
12	Biscoito, doce, wafer, recheado de morango	1,2	513	2148	4,5	26,4	1	67,4	0,8	0,6	14
13	Biscoito, salgado, cream cracker	4,1	432	1806	10,1	14,4	NA	68,7	2,5	2,7	20
14	Bolo, mistura para	1,0	419	1752	6,2	6,1	Tr	84,7	1,7	2,0	59
15	Bolo, pronto, aipim	34,1	324	1355	4,4	12,7	73	47,9	0,7	0,8	85
16	Bolo, pronto, chocolate	19,3	410	1715	6,2	18,5	77	54,7	1,4	1,3	75
17	Bolo, pronto, coco	29,3	333	1395	5,7	11,3	63	52,3	1,1	1,4	57
18	Bolo, pronto, milho	36,7	311	1303	4,8	12,4	82	45,1	0,7	1,0	83
19	Canjica, branca, crua	13,6	358	1496	7,2	1,0	NA	78,1	5,5	0,2	2
20	Canjica, com leite integral	72,5	112	471	2,4	1,2	1	23,6	1,2	0,3	43
21	Cereais, milho, flocos, com sal	9,3	370	1546	7,3	1,6	NA	80,8	5,3	1,0	2
22	Cereais, milho, flocos, sem sal	11,2	363	1520	6,9	1,2	NA	80,4	1,8	0,3	2
23	Cereais, mingau, milho, infantil	4,7	394	1650	6,4	1,1	NA	87,3	3,2	0,5	219
24	Cereais, mistura para vitamina, trigo, cevada e aveia	4,4	381	1595	8,9	2,1	NA	81,6	5,0	3,0	584
25	Cereal matinal, milho	5,5	365	1529	7,2	1,0	NA	83,8	4,1	2,5	143
26	Cereal matinal, milho, açúcar	4,3	377	1576	4,7	0,7	NA	88,8	2,1	1,5	56
27	Creme de arroz, pó	7,3	386	1615	7,0	1,2	NA	83,9	1,1	0,5	7
28	Creme de milho, pó	5,7	333	1393	4,8	1,6	NA	86,1	3,7	1,7	323
29	Curau, milho verde	81,6	78	328	2,4	1,6	5	13,9	0,5	0,5	53
30	Curau, milho verde, mistura para	3,9	402	1683	2,2	13,4	NA	79,8	2,5	0,7	31
31	Farinha, de arroz, enriquecida	12,7	363	1519	1,3	0,3	NA	85,5	0,6	0,2	4



Os preços dos alimentos foram retirados do site do Carrefour,
em dezembro de 2024

				
Arroz Branco Carrefour 1 Kg R\$ 6,29	Farinha de Mandioca Biju Carrefour Classic 500g R\$ 6,49	Feijão Branco Carrefour 500 g R\$ 10,29	Feijão Preto Carrefour Classic 1 Kg R\$ 7,59	Farinha de Rosca Carrefour 500 g R\$ 9,79
ADICIONAR	ADICIONAR	ADICIONAR	ADICIONAR	ADICIONAR

Os preços de algumas frutas e verduras foram obtidas do site de um hortifruti chamado Instafruta no site <https://instafruta.com.br/>



Planilha com entrada de dados para o modelo

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	id	alimento	calorias	carboidratos	proteinas	gorduras	ferro	magnesio	vitamina_c	zinco	sodio	preco	max_porcoes_dia	max_dias	grupo
2	2	arroz	360	77.5	7.3	1.9	0.9	110	0	1.4	2	1.3	20	7	4
3	7	aveia	394	66.6	13.9	8.5	4.4	119	1.4	2.6	5	2.44	10	7	4
4	40	macarrão	371	77.9	10	1.3	0.9	28	0	0.8	7	0.86	15	3	4
5	42	milho	361	87.1	0.6	0	0.1	3	0	0.1	8	1.82	10	2	4
6	21	cereais	370	80.8	7.3	1.6	0.5	20	0	0.6	272	6.9	10	3	4
7	27	cremeDeArroz	386	83.9	7	1.2	0.6	51	0	1.9	1	4.99	5	1	4
8	28	cremeDeMilho	333	86.1	4.8	1.6	4.3	30	96.3	0.8	594	1.31	5	1	4
9	19	canjica	358	78.1	7.2	1	0.3	12	0	0.4	1	2.27	5	1	4
10	86	batataBaroa	80	18.9	0.9	0.2	0.4	8	17.1	0.4	2	2.6	10	3	4
11	88	batataDoce	77	18.4	0.6	0.1	0.2	11	23.8	0.1	3	1.5	15	3	4
12	91	batatalInglesa	52	11.9	1.2	0	0.2	5	3.8	0.2	2	699	15	3	4
13	64	abobora	48	10.8	1.4	0.7	0.3	9	7.5	0.3	1	0.4	12	7	3
14	75	agriao	17	2.3	2.7	0.2	3.1	18	60.1	0.7	7	3.75	4	7	3
15	76	aipo	19	4.3	0.8	0.1	0.7	9	5.9	0.1	10	4.5	3	7	3
16	80	alface	13	2.5	0.9	0.2	2.5	9	13.5	0.2	7	2.39	6	5	3
17	82	alho	113	23.9	7	0.2	0.8	21	0	0.8	5	2.49	5	3	3
18	83	alhoPoro	32	6.9	1.4	0.1	0.6	11	14.1	0.2	2	3.59	8	3	3
19	84	almeirao	18	3.3	1.8	0.2	0.7	21	1.7	0.3	2	2.75	5	3	3
20	95	berinjela	19	4.5	0.7	0.1	0.2	9	0	0.1	1	1.12	5	3	3
21	97	beterraba	32	7.2	1.3	0.1	0.2	17	1.2	0.4	23	0.5	5	3	3
22	126	inhame	97	23.2	2.1	0.2	0.4	29	5.6	0.3	0	1.39	5	3	3
23	127	jilo	27	6.2	1.4	0.2	0.3	21	6.8	0.1	0	2.75	8	3	3
24	130	mandioca	151	36.2	1.1	0.3	0.3	44	16.5	0.2	2	0.5	12	5	3
25	133	manjericao	21	3.6	2	0.4	1	58	2.3	0.5	4	2	5	3	3
26	135	mostarda	18	3.2	2.1	0.2	1.1	16	38.6	0.3	3	1.92	5	3	3
27	139	palmito	29	5.5	2.5	0.5	0.2	25	8.7	0.4	563	4.058	8	4	3
28	142	pepino	10	2	0.9	0	0.1	9	5	0.1	0	0.95	4	4	3
29	143	pimentaoAmarelo	28	6	1.2	0.4	0.4	11	201.4	0.2	0	2.6	4	4	3
30	144	pimentaoVerde	21	4.9	1.1	0.2	0.4	8	100.2	0.1	0	1.5	4	4	3
31	145	pimentaoVermelho	23	5.5	1	0.1	0.3	11	158.2	0.2	0	4.86	4	4	3
32	147	quiabo	30	6.4	1.9	0.3	0.4	50	5.6	0.6	1	3.3	3	3	3
33	148	rahanete	14	2.7	1.4	0.1	0.4	10	9.6	0.2	11	3.12	8	3	3

Algoritmo

```
import pandas as pd
from gurobipy import Model, GRB, quicksum

# Carregar os dados da planilha
df = pd.read_excel("alimentosCompleto.xlsx")

# Valor suficientemente grande para M
M = 50

# Parâmetros do problema
alimentos = df['alimento'].tolist() #alimentos da planilha
dias = 7 # número de dias
refeicoes = 4 # número de refeições por dia
grupos = list(range(1, 8)) #quantidade de grupos

# Nutrientes e limites
nutrientes = ["calorias", "carboidratos", "proteinas", "gorduras", "ferro", "magnesio", "vitamina_c", "zinco", "sodio"]
limites_nutrientes = {
    "calorias": (3000, 5000),
    "carboidratos": (450, 900),
    "proteinas": (120, 150),
    "gorduras": (70, 120),
    "ferro": (8, 18),
    "magnesio": (400, 420),
    "vitamina_c": (75, 90),
    "zinco": (11, 16),
    "sodio": (1500, 6000)
}
```



Algoritmo



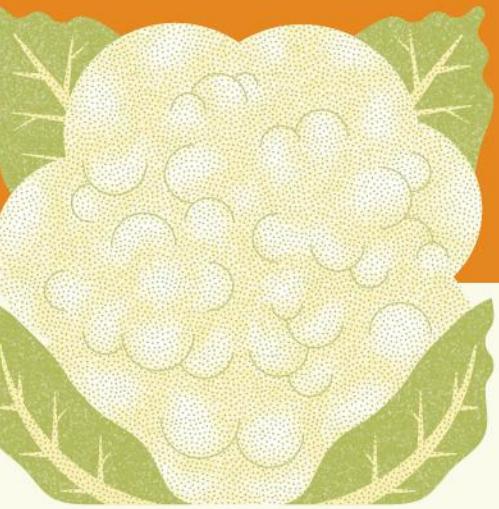
```
# Dados das colunas da planilha
calorias = df['calorias'].tolist()
carboidratos = df['carboidratos'].tolist()
proteinas = df['proteinas'].tolist()
gorduras = df['gorduras'].tolist()
ferro = df['ferro'].tolist()
magnesio = df['magnesio'].tolist()
vitamina_c = df['vitamina_c'].tolist()
zinco = df['zinco'].tolist()
sodio = df['sodio'].tolist()
preco = df['preco'].tolist()
max_porcoes = df['max_porcoes_dia'].tolist()
max_dias = df['max_dias'].tolist()
grupo = df['grupo'].tolist()

# Modelo
model = Model("problemaDaDieta")

# Variáveis de decisão
X = model.addVars(len(alimentos), range(dias), range(refeicoes), vtype=GRB.INTEGER, name="X")
Z = model.addVars(len(alimentos), range(dias), range(refeicoes), vtype=GRB.BINARY, name="Z")
Y = model.addVars(len(alimentos), range(dias), vtype=GRB.BINARY, name="Y")

# Função objetivo: minimizar o custo total
model.setObjective(quicksum(preco[i] * X[i,j,k] for i in range(len(alimentos)) for j in range(dias) for k in range(refeicoes)), GRB.MINIMIZE)
```

Algoritmo



```
#Restrições

#Restrições de quantidades de nutrientes
for nutriente in nutrientes:
    min_val, max_val = limites_nutrientes[nutriente]
    for j in range(dias):
        model.addConstr(
            quicksum(df.loc[i, nutriente] * X[i, j, k] for i in range(len(alimentos)) for k in range(refeicoes)) >= min_val,
            f"{nutriente}_min_dia_{j}"
        )
        model.addConstr(
            quicksum(df.loc[i, nutriente] * X[i, j, k] for i in range(len(alimentos)) for k in range(refeicoes)) <= max_val,
            f"{nutriente}_max_dia_{j}"
        )

#Restrições porções minimas de grupos alimentares por dia
#1-laticinios_acucar  2-frutas   3-horticolas   4-cereais_derivados_tuberculos  5-carne_peixe_ovos  6-leguminosas 7-oleos_gorduras
min_porcoes_grupo = {
    1: 0,
    2: 3,
    3: 3,
    4: 2,
    5: 5,
    6: 0,
    7: 0
}

for j in range(dias):
    for g in grupos:
        model.addConstr(
            quicksum(X[i,j,k] for i in range(len(alimentos)) if grupo[i] == g for k in range(refeicoes)) >= min_porcoes_grupo[g],
            f"min_porcoes_grupo_{g}_dia_{j}"
        )
```

Algoritmo

```
# Restrições de máximo de porções por alimento i por dia
for i in range(len(alimentos)):
    for j in range(dias):
        model.addConstr(
            quicksum(X[i,j,k] for k in range(refeicoes)) <= Y[i,j] * max_porcoes[i],
            f"max_porcoes_alimento_{i}_dia_{j}"
        )

# Restrição de quantidade máxima de dias que um alimento pode ser consumido em uma semana
for i in range(len(alimentos)):
    model.addConstr(
        quicksum(Y[i,j] for j in range(dias)) <= max_dias[i],
        f"max_dias_alimento_{i}"
    )

# Restrição de que um alimento i pode ser usado em no máximo 2 refeições diárias
for i in range(len(alimentos)):
    for j in range(dias):
        model.addConstr(
            quicksum(Z[i,j,k] for k in range(refeicoes)) <= 2 * Y[i,j],
            f"max_refeicoes_alimento_{i}_dia_{j}"
        )

# Restrição de limite total de 50 porções de alimento por dia
for j in range(dias):
    model.addConstr(
        quicksum(X[i,j,k] for i in range(len(alimentos)) for k in range(refeicoes)) <= 50,
        f"limite_total_porcoes_dia_{j}"
    )

# Garantir que em cada refeição pelo menos uma porção de alimento foi consumida
for j in range(dias):
    for k in range(refeicoes):
        model.addConstr(
            quicksum(X[i, j, k] for i in range(len(alimentos))) >= 1,
            f"min_1_porção_consumida_dia_{j}_refeicao_{k}"
        )
```



Algoritmo

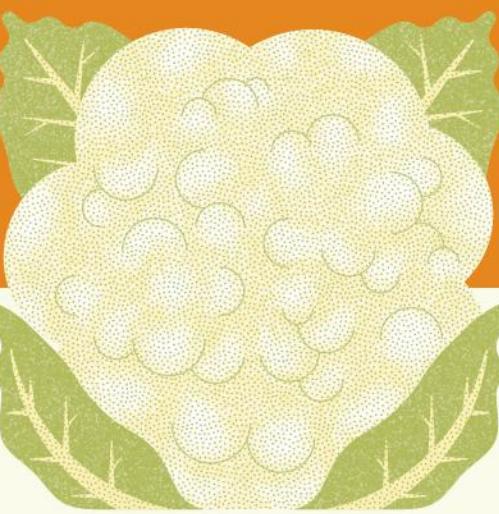
```
# Amarra Zi,j,k e Xi,j,k. Garantir que se X > 0 , o Z dele está ativado.
for i in range(len(alimentos)):
    for j in range(dias):
        for k in range(refeicoes):
            model.addConstr(
                M * Z[i, j, k] >= X[i, j, k] ,
                f"amarra_Z_X_{i}_dia_{j}_ref_{k}"
            )

# Garantir que Zi,j,k seja 0 se Xi,j,k é 0.
for i in range(len(alimentos)):
    for j in range(dias):
        for k in range(refeicoes):
            model.addConstr(
                X[i, j, k] >= Z[i, j, k],
                f"desativar_Z_se_X_zero_{i}_dia_{j}_ref_{k}"
            )

# Amarra Yi,j e Xi,j,k. Garantir que ativamos o Y somente se o alimento i tiver uma quantidade X maior que 0.
for i in range(len(alimentos)):
    for j in range(dias):
        model.addConstr(
            quicksum(X[i, j, k] for k in range(refeicoes)) >= Y[i, j],
            f"amarra_Y_X_{i}_dia_{j}"
        )

#Configurações de parada do modelo
model.setParam("MIPGap", 0.04)
model.setParam("TimeLimit", 3600)

# Resolver o modelo
model.optimize()
```



Algoritmo

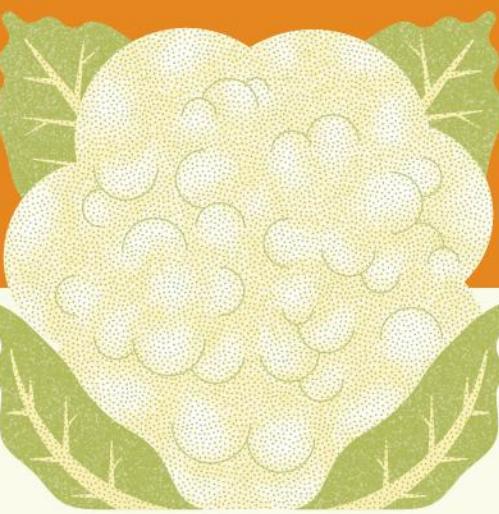
```
# Escrever os resultados em um arquivo (limpando o conteúdo antes de escrever novamente)
with open("resultado.txt", "w") as arquivo:

    if model.status == GRB.OPTIMAL:
        arquivo.write("Solução ótima encontrada devido ao atingimento do MIPGap ou conclusão satisfatória!\n")
    elif model.status == GRB.TIME_LIMIT:
        arquivo.write("Limite de tempo atingido. Apresentando a melhor solução encontrada até o momento:\n")
    else:
        arquivo.write(f"Status desconhecido ou solução não ótima: {model.status}\n")

    # Verificar o status da solução
    if model.status == GRB.OPTIMAL or model.status == GRB.TIME_LIMIT:
        arquivo.write(f"Tempo total de execução: {model.Runtime:.2f} segundos\n")
        arquivo.write(f"MIPGap atual: {model.MIPGap * 100:.2f}%\n")
        arquivo.write(f"Valor da função objetivo (custo total): {model.objVal}\n")

    # Variáveis de decisão organizadas por dia e refeição
    arquivo.write("\nPorções consumidas por refeição (organizado por dia e refeição):\n")
    for j in range(dias):
        arquivo.write(f"\nDia {j +1}:\n")
        for nutriente in nutrientes:
            total_nutriente = sum(df.loc[i, nutriente] * X[i, j, k].x for i in range(len(alimentos)) for k in range(refeicoes))
            arquivo.write(f" Total de {nutriente}: {total_nutriente:.2f}\n")
        for k in range(refeicoes):
            arquivo.write(f"\nDia {j+1}, Refeição {k+1}:\n")
            for i in range(len(alimentos)):
                if X[i, j, k].x > 0: # Mostrar apenas variáveis com valores positivos
                    arquivo.write(f" Alimento {alimentos[i]}: {X[i, j, k].x} porções\n")

    # Dias em que os alimentos foram consumidos
    arquivo.write("\nDias em que os alimentos foram consumidos (Y[i,j]):\n")
    for j in range(dias):
        arquivo.write(f"\nDia {j+1}:\n")
        for i in range(len(alimentos)):
            if Y[i, j].x > 0:
                arquivo.write(f" Alimento {alimentos[i]}\n")
```



Resultados e conclusões



Dia 1

Refeição	Alimento	Porções
1	mandioca	3.0
2	canjica	3.0
2	ovo	6.0
3	arroz	1.0
3	ovo	1.0
4	palmito	1.0
4	melancia	3.0

Dia 2

Refeição	Alimento	Porções
1	beterraba	1.0
1	melao	2.0
2	mandioca	2.0
2	ovo	9.0
3	milho	3.0
3	ovo	1.0
4	arroz	1.0
4	mandioca	1.0
4	mexerica	1.0

Dia 3

Refeição	Alimento	Porções
1	macarrão	4.0
1	abobora	2.0
1	ovo	2.0
2	melao	2.0
3	macarrão	1.0
3	palmito	1.0
3	ovo	3.0
4	limao	1.0
4	amendoim	1.0

Dia 4

Refeição	Alimento	Porções
1	pera	2.0
2	arroz	1.0
2	batatalnglesa	4.0
2	limao	1.0
2	ovo	6.0
3	cereais	2.0
3	ovo	1.0
4	arroz	1.0
4	abobora	4.0



Resultados e conclusões



Dia 5

Refeição	Alimento	Porções
1	mandioca	3.0
1	feijao	1.0
1	ovo	1.0
2	ovo	6.0
3	melancia	3.0
4	macarrão	4.0
4	palmito	1.0

Dia 6

Refeição	Alimento	Porções
1	mandioca	1.0
1	palmito	1.0
1	ovo	6.0
2	arroz	1.0
2	abobora	1.0
2	limao	1.0
2	ovo	1.0
3	macarrão	4.0
3	melancia	1.0
4	melancia	1.0

Dia 7

Refeição	Alimento	Porções
1	mandioca	3.0
2	beterraba	1.0
3	milho	3.0
3	mexerica	1.0
4	arroz	1.0
4	melao	2.0
4	ovo	10.0



Resultados e conclusões



Nutrientes resultantes para cada dia da semana

	Dia 1	Dia 2	Dia 3	Dia 4	Dia 5	Dia 6	Dia 7
Calorias	3037	3504	3406	3020	3163	3192	3504
Carboidratos	454.4	490.5	464.4	450.7	467.8	473.1	490.5
Proteínas	130.5	149	146.6	134.8	146.4	148.1	149
Gorduras	72.8	98	110	76.6	73.6	75.2	98
Ferro	14	17.8	14.7	15.7	17.1	16.4	17.8
Magnésio	410	402	419	415	418	407	402
Vitamina c	76.5	89.9	79.3	89	76.5	83.1	89.9
Zinco	12.3	15	13.5	14.8	13.6	14.3	15
Sódio	1596	1538	1729	1583	1621	1619	1538

Esse caso de teste apresentou um custo semanal de 191.673 reais.

Tempo total de execução 3600 segundos e MIPGap de 6.54%.



Referências

- Oliveira, D., Borges, A. C., & Vieira, V. (2020). O uso da programação linear para otimização do custo de refeições diárias para idosos em Monte Carmelo - MG. 2020. Disponível em:<https://ojs.brazilianjournals.com.br/ojs/index.php/BRJD/article/view/11443/9551>
- Moreira, R. P. C., & Wanner, E. F. (n.d.). Modelos de programação linear para a geração de cardápios escolares alinhados às diretrizes do PNAE. Disponível em:<https://sig.cefetmg.br/sigaa/verArquivo?idArquivo=4307091&key=b92444f4c785d39194b3e65a05bc61ca>
- Tabela Brasileira de Composição de Alimentos (TACO). Tabela TACO - Composição de Alimentos. Disponível em: <http://www.nepa.unicamp.br>.
- Carrefour. (n.d.). Disponível em:<https://mercado.carrefour.com.br>.
- Instafruta. (n.d.). Disponível em: <https://instafruta.com.br/>.
- Instituto Português do Desporto e Juventude. (2016). Manual de curso de treinadores de desporto: grau II. Lisboa: Instituto Português do Desporto e Juventude, 2016. Disponível em:https://www.kufunda.net/publicdocs/GrauII_08_Nutricao.pdf
- Negretti, M. (2022). Nutrição, exercício físico e desempenho: recomendações nutricionais para uma prática saudável Disponível em:<https://repositorio.unesp.br/server/api/core/bitstreams/6faaee93-9ea2-46c0-94a4-20ca76873876/content>