Uma Guerra de 12 Dias

Mário Leite

Este programa simula um cenário fictício de defesa aérea durante um conflito hipotético entre dois países fictícios, denominados país "A" e "B", respectivamente. O foco da simulação é o sistema de defesa do país "A", chamado "Capacete de Aço", composto por três tipos de mísseis terra-ar com nomes fictícios: "Xandir", "Samir" e "Flecha". O objetivo do programa é estimar o consumo e o custo desses mísseis durante 12 dias consecutivos de confrontos, levando em consideração ataques diários do país "B" usando diferentes tipos de armas aéreas - foguetes, drones, mísseis médios e mísseis hipersônicos - com números e custos totalmente inventados para fins didáticos e ilustrativos. Além disso, o programa calcula a reposição necessária para manter os estoques dos mísseis do país "A", custos operacionais do sistema de defesa, e apresenta gráficos demonstrativos do consumo e dos custos de reposição ao longo do período considerado.

Enfim, esta simulação visa ser uma ferramenta educacional e exploratória para entender conceitos básicos de logística e custos em um sistema de defesa imaginário, sem nenhuma relação com situações reais ou atuais. Entretanto, vale para que todos nós nos preocupemos com qualquer cenário de guerra em que vidas estão em perigo, e que qualquer tipo de guerra deve ser evitado.

As figuras **1a**, **1b** e **1c** mostram, respectivamente: uma tabela-resumo da situação do país "A", com relação aos seus estoques de mísseis na defesa contra os ataques do país “B”, um gráfico de barras mostrando o consumo de misseis do sistema de defesa e um gráfico pizza com os percentuais de custos de cada tipo de míssil desse sistema. O programa foi codificado em Python pode ter bons recursos gráficos.

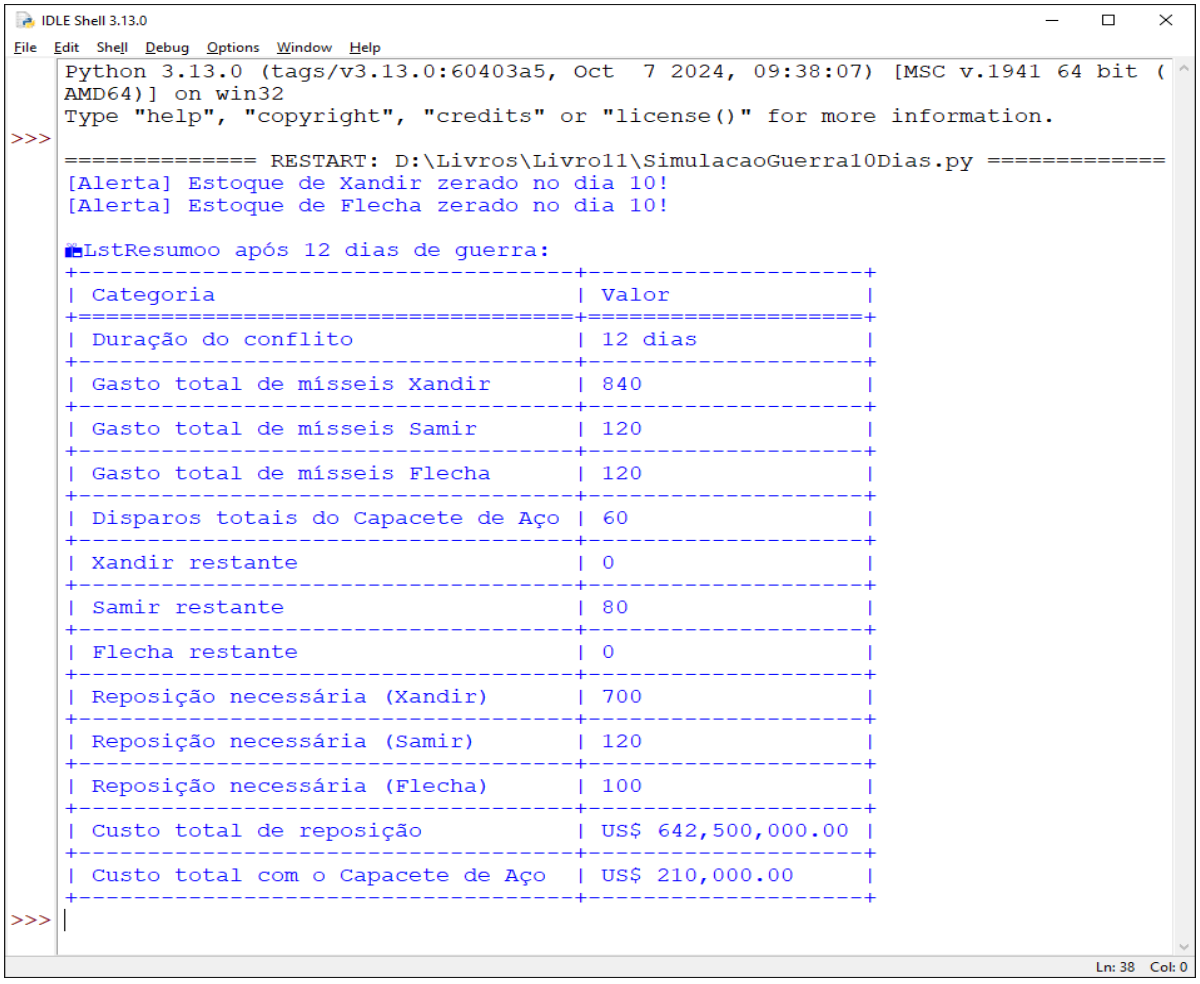

Figura 1a - Tabela-resumo dos valores envolvidos no conflito

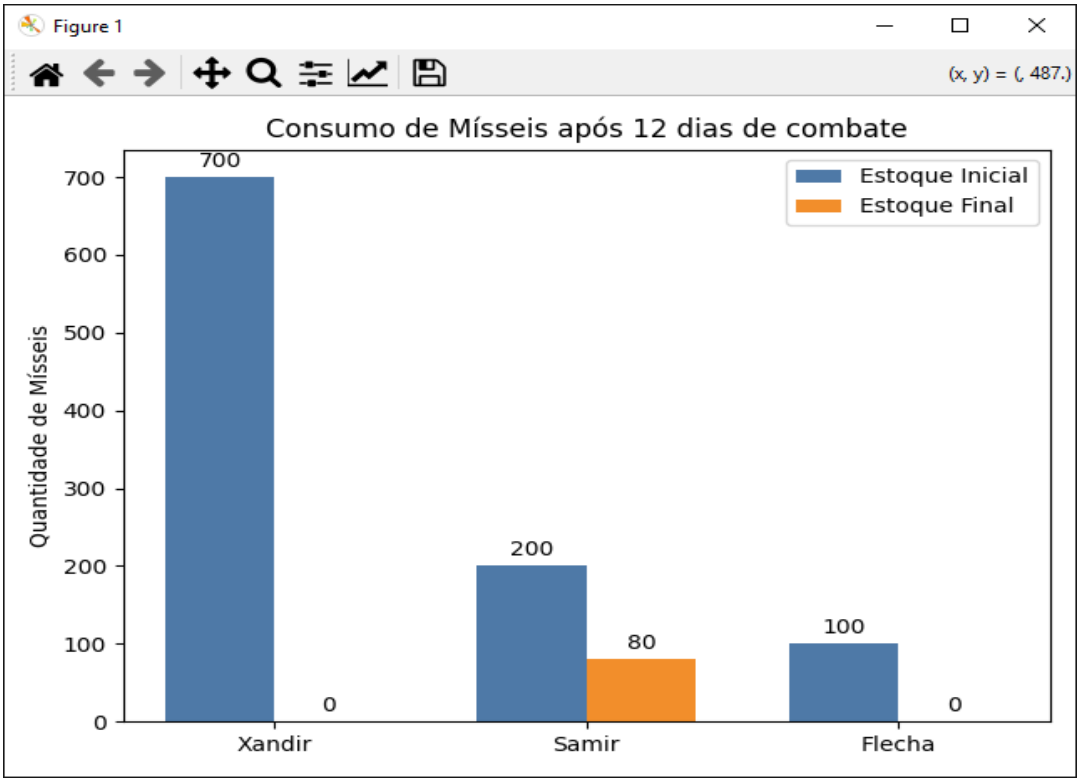

Figura 1b - Consumo de mísseis de defesa no conflito

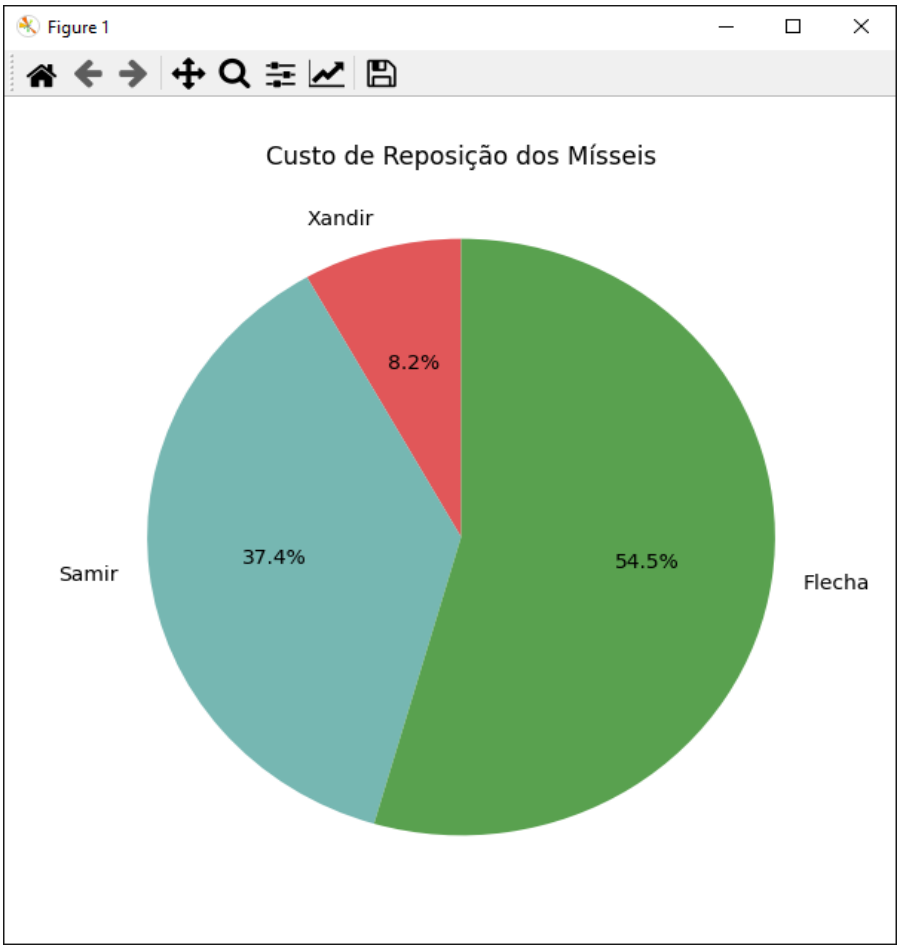

Figura 1c - Custos dos mísseis de defesa envolvidos no conflito

SimulacaoGuerra12Dias.py

```
-------------------------------------------------------------------------------
Simula um cenário hipotético, mas bem fundamentado, que retrata uma situação de
guerra limitada entre um país fictício "A" e um outro país fictício "B", com valores
hipotéticos. Considera ataques intensos de mísseis e drones do país "B" por 12 dias
consecutivos, com o país "A" se defendendo com um sistema chamado "Capacete de Aço",
composto por apenas três tipos de mísseis terra-ar: 'Xamir', 'Samir' e 'Flecha'.
-------------------------------------------------------------------------------
'''
import matplotlib.pyplot as plt
import numpy as np
from tabulate import tabulate

#Configurações iniciais
dias = 12  #simula 12 dias consecutivos de guerra

#Estoque inicial dos mísseis
estoqueXandir = 700
estoqueSamir = 200
estoqueFlecha = 100

#Custo médio de cada míssil (em dólares)
custoXandir = 75_000
custoSamir = 2_000_000
custoFlecha = 3_500_000
custoCapaceteDisparo = 3500  #custo operacional do laser (mais barato)

#Ameaças diárias esperadas
foguetesPorDia = 40
dronesPorDia = 30
misseisMediosPorDia = 10
misseisHipersonicosPorDia = 5

#Interceptações médias por alvo
xandirPorFoguete = 1
xandirPorDrone = 1
samirPormissilMedio = 1
flechaPorMissilHiperssonico = 2

#Variáveis acumuladoras
gastoXandirTotal = 0
gastoSamirTotal = 0
gastoFlechaTotal = 0
disparosCapaceteTotal = 0

#Simulação por dia
for dia in range(dias):
    #Gastos diários estimados
    gastoXandir = foguetesPorDia * xandirPorFoguete + dronesPorDia * xandirPorDrone
    gastoSamir = misseisMediosPorDia * samirPormissilMedio
    gastoFlecha = misseisHipersonicosPorDia * flechaPorMissilHiperssonico
    disparosCapacete = 5  #média de disparos por dia

    #Acumulando gastos
    gastoXandirTotal += gastoXandir
    gastoSamirTotal += gastoSamir
    gastoFlechaTotal += gastoFlecha
    disparosCapaceteTotal += disparosCapacete
```

```
    #Reduzindo estoque e verificando esgotamento
    if(estoqueXandir > 0 and estoqueXandir >= gastoXandir):
        estoqueXandir -= gastoXandir
        if(estoqueXandir == 0):
            print(f"[Alerta] Estoque de Xandir zerado no dia {dia+1}!")

    if(estoqueSamir > 0 and estoqueSamir >= gastoSamir):
        estoqueSamir -= gastoSamir
        if(estoqueSamir == 0):
            print(f"[Alerta] Estoque de Samir zerado no dia {dia+1}!")
    if(estoqueFlecha > 0 and estoqueFlecha >= gastoFlecha):
        estoqueFlecha -= gastoFlecha
        if(estoqueFlecha == 0):
            print(f"[Alerta] Estoque de Flecha zerado no dia {dia+1}!")

#Calcular reposição necessária
reposicaoXandir = max(0, 700 - estoqueXandir)
reposicaoSamir = max(0, 200 - estoqueSamir)
reposicaoFlecha = max(0, 100 - estoqueFlecha)

#Custo total da reposição
custoReposicaoXandir = reposicaoXandir * custoXandir
custoReposicaoSamir = reposicaoSamir * custoSamir
custoReposicaoFlecha = reposicaoFlecha * custoFlecha
custoReposicaoTotal=custoReposicaoXandir+custoReposicaoSamir+ custoReposicaoFlecha

#Custo do uso do Iron Beam
custoCapaceteTotal = disparosCapaceteTotal * custoCapaceteDisparo

#Tabela-lista final de resultados
LstResumo = [
    ["Duração do conflito", f"{dias} dias"],
    ["Gasto total de mísseis Xandir", gastoXandirTotal],
    ["Gasto total de mísseis Samir", gastoSamirTotal],
    ["Gasto total de mísseis Flecha", gastoFlechaTotal],
    ["Disparos totais do Capacete de Aço", disparosCapaceteTotal],
    ["Xandir restante", estoqueXandir],
    ["Samir restante", estoqueSamir],
    ["Flecha restante", estoqueFlecha],
    ["Reposição necessária (Xandir)", reposicaoXandir],
    ["Reposição necessária (Samir)", reposicaoSamir],
    ["Reposição necessária (Flecha)", reposicaoFlecha],
    ["Custo total de reposição", f"US$ {custoReposicaoTotal:,.2f}"],
    ["Custo total com o Capacete de Aço", f"US$ {custoCapaceteTotal:,.2f}"]
]

print("\n LstResumoo após 12 dias de guerra:")
print(tabulate(LstResumo, headers=["Categoria", "Valor"], tablefmt="grid"))

#Gráfico de consumo de mísseis
labels = ['Xandir', 'Samir', 'Flecha']
inicial = [700, 200, 100]
final = [estoqueXandir, estoqueSamir, estoqueFlecha]

x = np.arange(len(labels))
width = 0.35

fig, ax = plt.subplots()
rects1 = ax.bar(x-width/2, inicial, width, label='Estoque Inicial', color='#4e79a7')
rects2 = ax.bar(x+width/2, final, width, label='Estoque Final', color='#f28e2b')
```

```
ax.set_ylabel('Quantidade de Mísseis')
ax.set_title('Consumo de Mísseis após 12 dias de combate')
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.legend()

#-------------------------------------------------------------------------------------------
def CriarTitulos(rects):
    for rect in rects:
        height = rect.get_height()
        ax.annotate(f'{int(height)}',
            xy=(rect.get_x() + rect.get_width() / 2, height),
                xytext=(0, 3),
                   textcoords="offset points",
                        ha='center', va='bottom')
#-------------------------------------------------------------------------------------------
CriarTitulos(rects1)
CriarTitulos(rects2)
plt.tight_layout()
plt.show()

#Gráfico de custo de reposição
custosReposicoes = [custoReposicaoXandir, custoReposicaoSamir, custoReposicaoFlecha]
custosCustosDefesas = ['Xandir', 'Samir', 'Flecha']

plt.figure(figsize=(6, 6))
plt.pie(custosReposicoes,       labels=custosCustosDefesas,       autopct='%1.1f%%',
startangle=90,
        colors=['#e15759', '#76b7b2', '#59a14f'])
plt.title("Custo de Reposição dos Mísseis", pad=30)
plt.axis('equal')
plt.subplots_adjust(top=0.85)
plt.show()
#Fim do programa "SimulacaoGuerra12Dias" ----------------------------------------
```