Modelagem Computacional: Simulação 02 - Redes Tróficas

Alunos: Álvaro Cardoso Vicente de Souza 133536

Gabriel Angelo Cabral Neves 136124

Jhonatan Hiroo Eguchi 133691

Docente: Prof. Dr. Marcos Gonçalves Quiles

Universidade Federal de São Paulo - UNIFESP Instituto de Ciência e Tecnologia - Campus São José dos Campos

São José dos Campos - Brasil

Setembro de 2020

Descrição do modelo

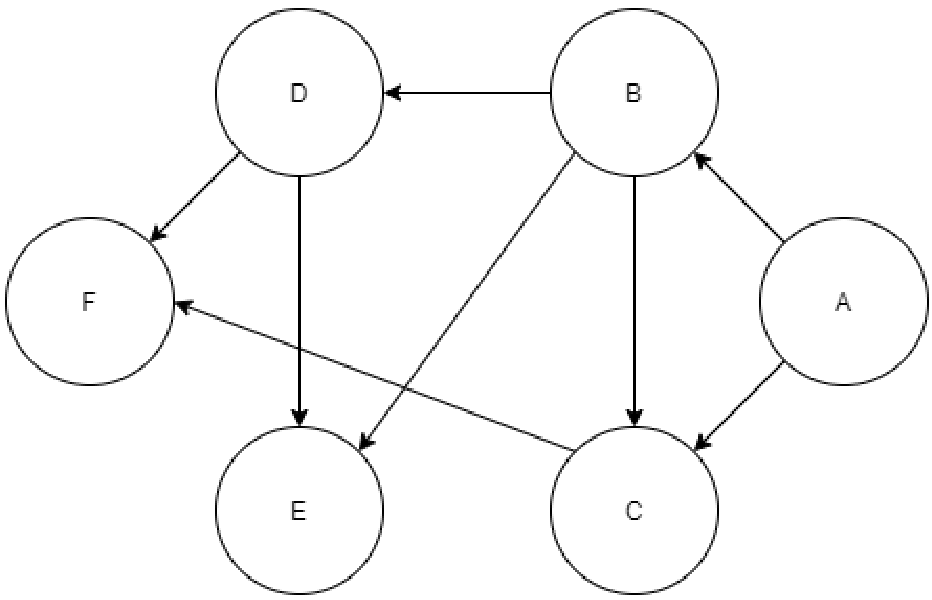
O projeto propõe a simulação de uma rede trófica, que simula um ecossistema com uma interligação entre os membros, formando um modelo de cadeia alimentar descrita pela relação Presa - Predador, entre toda a fauna e flora inserida no ecossistema. Para isso, foi usada uma implementação de um algoritmo, em Python 3, que leva em conta as relações entre os integrantes da cadeia alimentar, o nível de cada integrante, assim como, suas posições como presa ou predador, em relações distintas com cada integrante.

Utilizando o método de Euler podemos trabalhar, numericamente, com as equações diferenciais que o modelo matemático de Lotka-Volterra nos proporciona, que descrevem as oscilações na população de cada membro da rede trófica simulada, através, das relações de cada membro com o resto da cadeia alimentar.

Rede Trófica Desenvolvida

A rede trófica desenvolvida para esta simulação conta com seis integrantes e pode ser representada da seguinte maneira:

- Onça - Carnívoro **(A)**;
- Lobo guará - Onívoro **(B)**;
- Veado Campeiro - Herbívoro **(C)**;
- Caititu - Herbívoro **(D)**;
- Morango Silvestre - Planta **(E)**;
- Grama - Planta **(F)**.



Modelo Matemático

Após a definição da rede trófica com as relações entre os membros, podemos utilizar o modelo matemático de Lotka-Volterra para obtermos as equações diferenciais que descrevem a variação de cada uma das populações que compõem a rede trófica.

O modelo de Lotka-Volterra leva em conta a quantidade de predadores, quantidade de presas e os parâmetros de relação entre cada predador e presa gerando equações diferenciais que seguem o seguinte modelo:

$$\frac{dP}{dt} = P(\alpha V - \beta)$$

$$\frac{dV}{dt} = V(\lambda - \varphi P)$$

- $P(t)$ - População de predadores;
- α - Parâmetro de crescimento da população de predadores;
- β - Parâmetro de decrescimento da população de predadores;
- $V(t)$ - População de presas;
- λ - Parâmetro de crescimento da população de presas;
- φ - Parâmetro de decrescimento da população de presas.

Aplicando esse raciocínio a rede trófica desenvolvida para ser utilizada como modelo nas simulações, obtemos as seguintes equações para a variação cada população:

A: Onça

$$\frac{dA}{dt} = A(-p1 + p2B + p3C)$$

B: Lobo guará

$$\frac{dB}{dt} = B(-p4 - p5A + p6C + p7D + p8E)$$

C: Veado Campeiro

$$\frac{dC}{dt} = C(-p9 - p10A - p11B + p12F)$$

D: Caititu

$$\frac{dD}{dt} = D(-p13 - p14B + p15E + p16F)$$

E: Morango Silvestre

$$\frac{dE}{dt} = E(p17 - [\frac{p17 \cdot E}{k1}] - p18B - p19D)$$

F: Grama

$$\frac{dF}{dt} = F(p20 - [\frac{p20 \cdot F}{k2}] - p21C - p22D)$$

Onde A à F, representam o número de integrantes de cada população, $p1$ à $p22$ representam os parâmetro de crescimento e $k1$ e $k2$ são os termos logísticos responsáveis por limitar o crescimento da vegetação (grama e morangos silvestres).

Parâmetros Iniciais

O processo de determinação dos parâmetros iniciais necessários para que as populações inseridas no ecossistema cheguem a um equilíbrio não é algo trivial, podemos obtê-los através da escolha de parâmetros iniciais semelhantes para todas as populações e gerando várias simulações.

A cada simulação gerada realiza-se pequenas alterações nos parâmetros de acordo com os resultados obtidos na simulação anterior, como por exemplo, diminuir a taxa em que a presa morre, a população de um predador, entre outros fatores, para ao fim obtermos um sistema equilibrado. Começou-se pela base da cadeia, e a medida que conseguia o equilíbrio, subia-se mais um nível, até o predador primário.

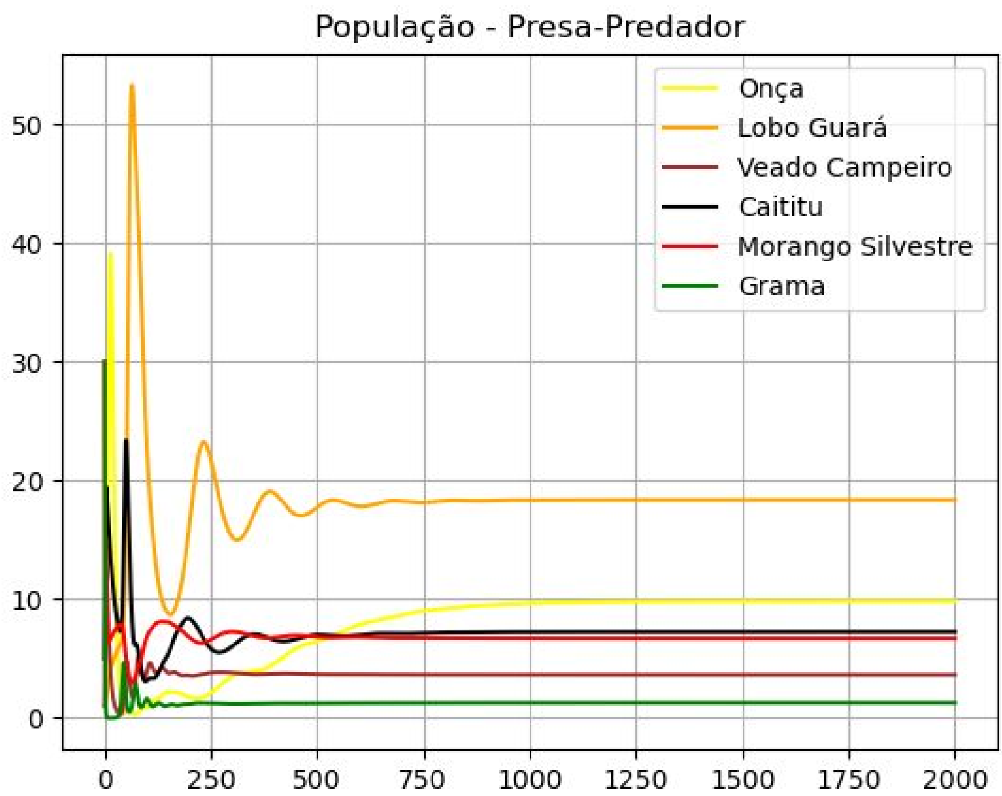
A tabela 1 apresenta os parâmetros utilizados para que o equilíbrio fosse atingido (p1 - p22 direita pra esquerda), no caso, para evitar o crescimento contínuo da morango e da grama foram definidos os termos logísticos k1 = 10 e k2 = 10, respectivamente, para que o crescimento pare nesses valores.

Tabela 1 - Parâmetros de crescimento/decrescimento

	A	B	C	D	E	F
A	-0,2	0,001	0,05	0	0	0
B	-0,001	0,1	0,001	0,01	0,005	0
C	-0,001	-0,001	-0,1	0	0	0,1
D	0	-0,001	0	-0,1	0,01	0,04
E	0	-0,005	0	-0,01	0,1	0
F	0	0	-0,1	0,01	0	0,1

Resultados e Discussão

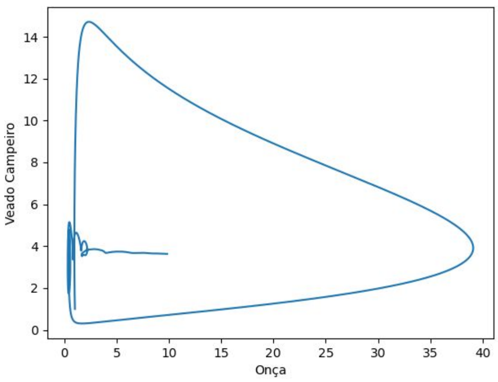
Caso 1 - Equilíbrio.

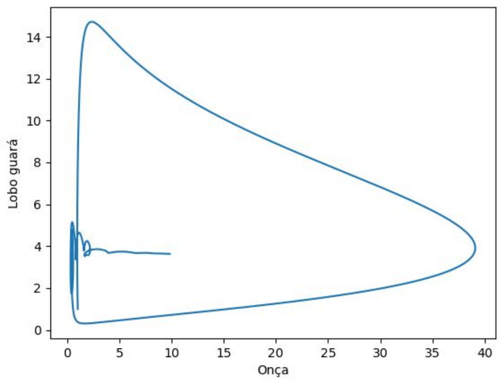


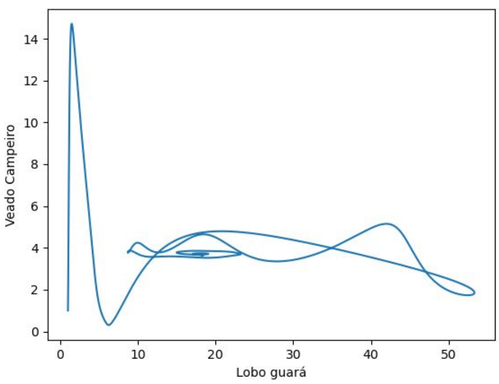
Apesar de um começo instável, o equilíbrio foi atingido depois de t = 1000. Note que, no começo, o decrescimento espontâneo da grama causou o crescimento dos caititus, no caso, os veados também se alimentam da grama, no entanto, por serem a presa mais suscetível à caça da onça e dos lobos, fazendo com que a taxa de crescimento não desse conta da sua caçada, e quase os extinguindo. Logo após esse desequilíbrio, houve a queda desses predadores primários e o crescimento dos herbívoros, que resultaram então, no equilíbrio.

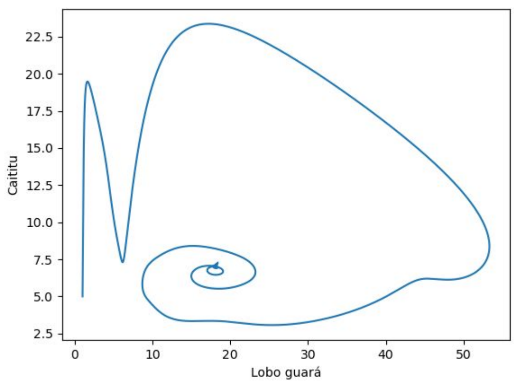
Abaixo se encontram os gráficos *Phase-Space* das populações de presas (Eixo Y) e seus respectivos predadores (Eixo X) durante o intervalo da simulação do ecossistema em equilíbrio, discutida acima.

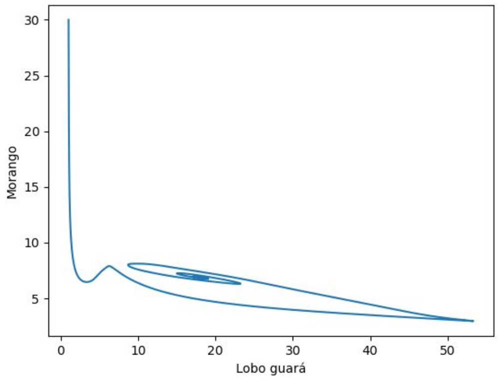
Podemos perceber que os gráficos obtidos não possuem um formato cíclico bem definido pois, a convergência na simulação do ecossistema equilibrado se dá de maneira abrupta, somado ao fato do ecossistema ser composto por 6 populações, porém ainda fica nítido o fato de que as populações de presa e predador convergem em um equilíbrio.

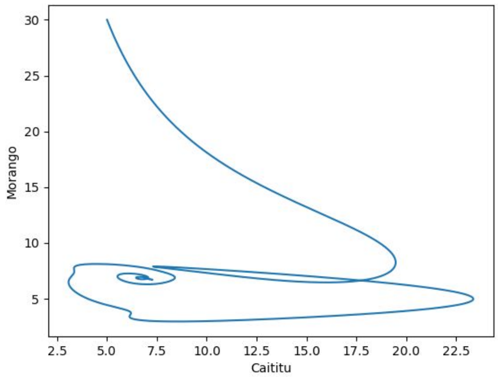


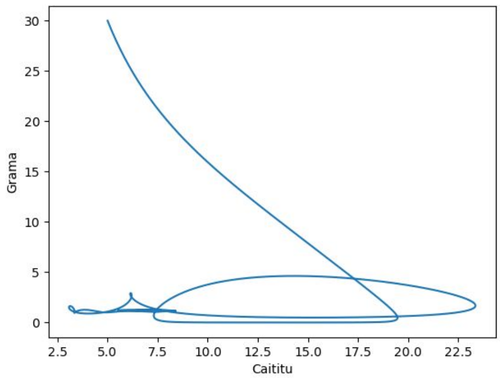


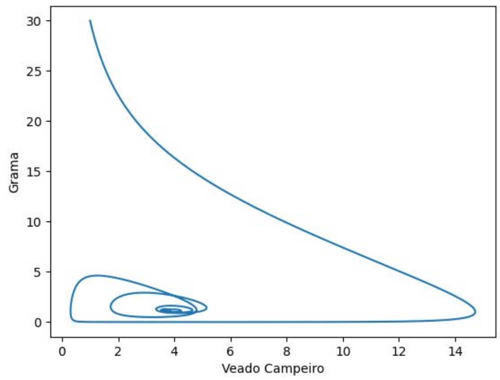




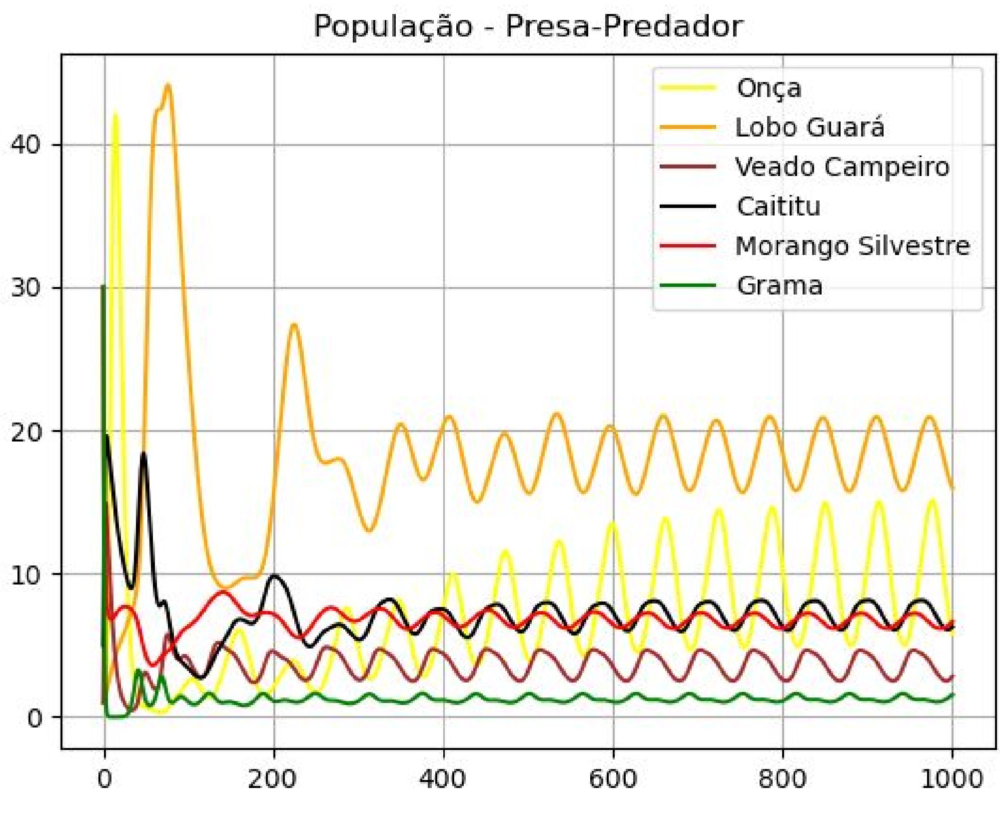








Caso 2 - Mudança de Estações

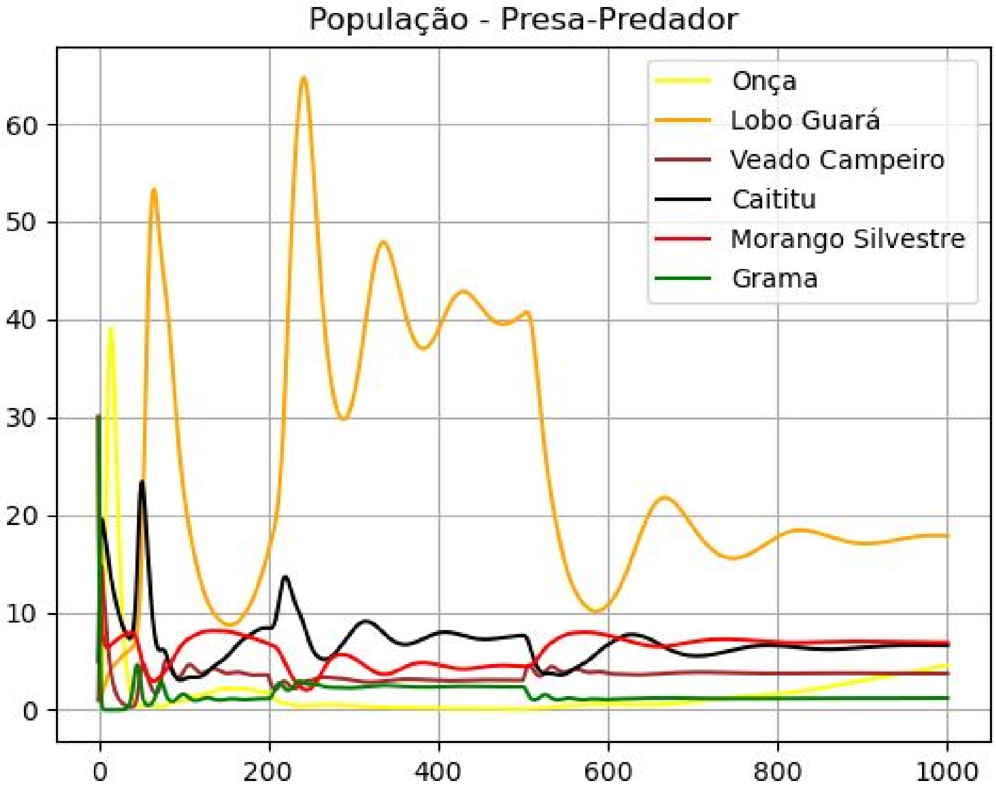


Foram simuladas estações que são expressas através de uma função seno com argumentos de duração de 0.1 e efetividade da chuva de 0.1 absorvida pelas plantas em relação ao tempo do sistema. Na equação abaixo pode ser observado, quando a função possui valor positivo, a quantidade de chuva, tem uma influência positiva sobre a vegetação, chegando a um valor máximo da amplitude para a grama e um terço da amplitude para o morango e uma influência negativa de mesma magnitude quando a função é negativa.

$$influencia \ = \ (efetividade) \cdot sin((duração) \cdot t)$$

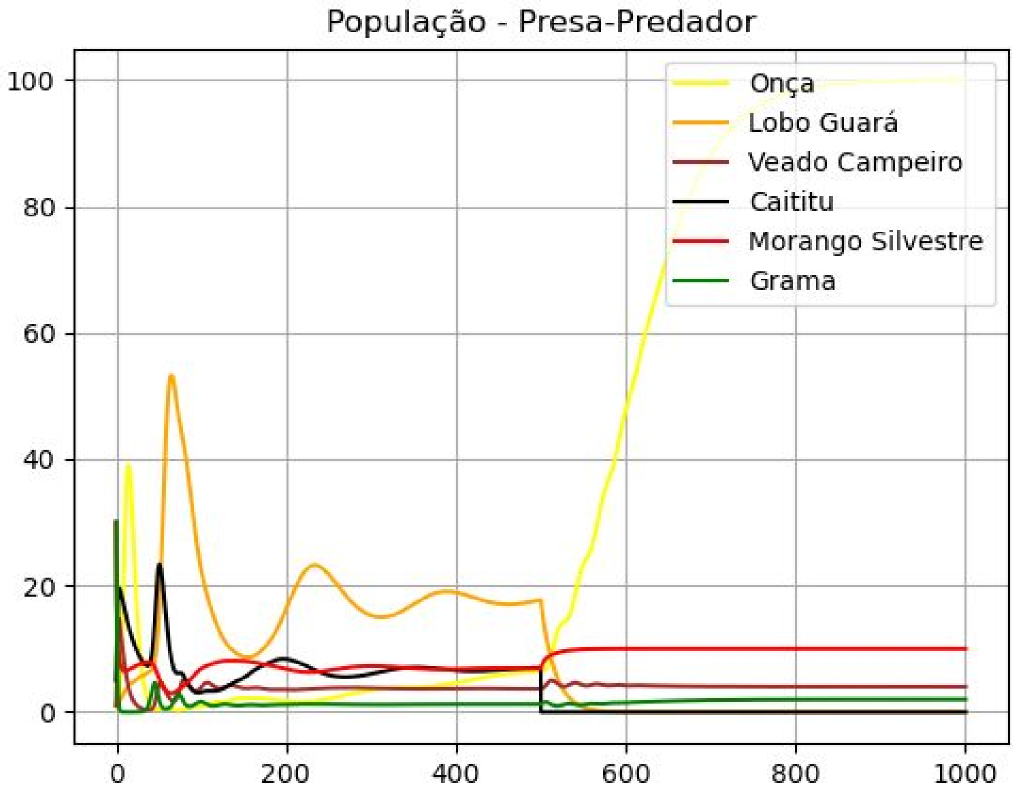
Analisando o gráfico podemos ver que todas as espécies possuem um comportamento senoidal após um tempo de simulação, com uma grande variação das onças e dos lobos guarás.

Caso 3 - Temporada de caça



Para a temporada de caça foi estabelecido um intervalo de tempo em que ocorre a caça, e, durante esse intervalo, há um decrescimento do animal caçado, que no caso, é o veado campeiro. Foi definido que entre TMax/5 e TMax/2 havia a caça, no gráfico observamos que entre t = 200 e t = 500 há o decrescimento dos veados, e, consequentemente, o decrescimento de seus predadores, que ao longo do tempo, foram caindo em quantidade; logo após essa temporada, o sistema volta a se estabilizar.

Caso 4 - Extinção abrupta



Para esta simulação foi definido que em TMax/2 ocorreria a extinção abrupta dos caititus.

Com a extinção abrupta em t = 500 dos caititus, a população de morangos cresce pois perdeu um de seus predadores, a da grama também mas logo se estabiliza pois o parâmetro no qual era consumida pelos caititus era pequeno. Com a perda de sua principal fonte de alimento a população de lobo guarás decresce até ser extinta pois o parâmetro no qual os caititus eram predados é muito maior que o da predação do veados.

Os veados, por sua vez, perdem um de seus principais predadores e acaba se estabilizando já que a onça passa a se alimentar deles sem competição dos lobos, fazendo com que a população de onça cresça muito antes de estabilizar novamente.

Anexo 1 - Implementação em Python 3

```
import numpy as np
import matplotlib.pyplot as plt
import math

# Modelo Lotka-Volterra
DT = 0.01 # Passo para cada atualização
TMax = 1000 # Tempo total de simulação

#valor inicial das populações (Respectivas cores nos gráficos)
onca = 1 # Amarelo
veado = 1 # Marrom
lobog = 1 # Laranja
caititu = 5 # Preto
morango = 30  # Vermelho
grama = 30 # Verde

# Parâmetros
p1 = 0.2  # Decrescimento da onça
p2 = 0.001 # onca come lobo
p3 = 0.05  # onca come veado
p4 = 0.1  # Decrescimento do lobo
p5 = 0.001  # Lobo é comido pela onça
p6 = 0.001  # Lobo come veado
p7 = 0.01  # Lobo come caititu
p8 = 0.005  # Lobo come morango
p9 = 0.1  # Decrescimento do veado
p10 = 0.001 # Veado é comido pela onça
p11 = 0.001 # Veado é comido pelo lobo
p12 = 0.1 # Veado come grama
p13 = 0.1 # Decrescimento do caititu
p14 = 0.001 # Caititu é comido pelo lobo
p15 = 0.01 # Caititu come morango
p16 = 0.04 # Caititu come grama
p17 = 0.5 # Crescimento do morango
p18 = 0.005 # Morango é comido pelo lobo
p19 = 0.01 # Morango é comido pelo caititu
p20 = 0.5 # Crescimento da grama
p21 = 0.1 # Grama é comido pelo veado
```

```
p22 = 0.01 # Grama é comido pelo caititu
k1 = 10 # Termologistico do crescimento do morango
k2 = 10 # Termo logistico do crescimento da grama

# Vetores de cada animal com os valores iniciais de cada população
onca_array = [onca]
lobog_array = [lobog]
veado_array = [veado]
caititu_array = [caititu]
morango_array = [morango]
grama_array = [grama]

# Inicia vetor do tempo com 0
t_array = [0]

# Matriz de adjacência com as constantes
m                                                                        =
[[p1,p2,p3,0,0,0],[p5,p4,p6,p7,p8,0],[p10,p11,p9,0,0,p12],[0,p14,0,p13,
p15,p16],[0,p18,0,p19,p17,0],[0,0,p21,p22,0,p20]]

# Simulação de estação e quantidade de chuva (Afeta 100% a grama e 33%
o morango)
estacao = 1 # 0 para não considerar e 1 para considerar
duracao = 0.1 # Extensão da estação
efetividade = 0.1 # efetividade de chuva

# Simulação de temporada de caça aos veados
temporada = 0 # 0 para não considerar e 1 para considerar
caca = 0.1 # Efetividade da caça
comeco = TMax/5 # Dia de ínicio
fim = TMax/2 # Dia de término

# Simulação da extinção abrupta dos caititus
extincao = 0 # 0 para não considerar e 1 para considerar
data = TMax/2 # Data de extinsão da espécie

# Laço para atualizar as populações de cada integrante da cadeia
for t in np.arange(DT, TMax, DT):
```

```
    dt_onca = onca * (-m[0][0] + m [0][1] * lobog + m[0][2] * veado) * DT #A0 - Eq Variação populacional da onça
    dt_loboguara = lobog * (-m[1][1] - m[1][0] * onca + m[1][2] * veado + m[1][3] * caititu + m[1][4] * morango) * DT #B0 - Eq Variação populacional do loboguará

    if(temporada == 1 and t >= comeco and t <= fim):
        dt_veado = veado * (-m[2][2] - m[2][0] * onca - m[2][1] * lobog + m[2][5] * grama - caca) * DT #C0 - Eq Variação populacional do veado considerando a caça
    else:
        dt_veado = veado * (-m[2][2] - m[2][0] * onca - m[2][1] * lobog + m[2][5] * grama) * DT #C0 - Eq Variação populacional do veado

    if(extincao == 1 and t == data):
        caititu = 0 # Acaba a população de caititus
        dt_caititu = 0
    else:
        dt_caititu = caititu * (-m[3][3] - m[3][1] * lobog + m[3][4]* morango + m[3][5] * grama) * DT #D0 - Eq Variação populacional do caititu

    if(estacao == 1): # Eq de variação considerando as estações
        dt_morango = morango * (m[4][4] - (m[4][4] * morango) / k1 - m[4][1] * lobog - m[4][3] * caititu + (efetividade/3)*math.sin(duracao*t)) * DT #E0 - Eq Variação populacional do morango
        dt_grama = grama * (m[5][5] - (m[5][5] * grama) / k2 - m[5][2] * veado - m[5][3] * caititu + efetividade*math.sin(duracao*t)) * DT #F0 - Eq Variação populacional da grama
    else:
        dt_morango = morango * (m[4][4] - (m[4][4] * morango) / k1 - m[4][1] * lobog - m[4][3] * caititu) * DT #E0 - Eq Variação populacional do morango
        dt_grama = grama * (m[5][5] - (m[5][5] * grama) / k2 - m[5][2] * veado - m[5][3] * caititu) * DT #F0 - Eq Variação populacional da grama

    # Atualização da populações
    onca += dt_onca
```

```
    lobog += dt_loboguara
    veado += dt_veado
    caititu += dt_caititu
    morango += dt_morango
    grama += dt_grama

    # Colocando os valores de populações nos vetores de cada espécie
    onca_array.append(onca)
    lobog_array.append(lobog)
    veado_array.append(veado)
    caititu_array.append(caititu)
    morango_array.append(morango)
    grama_array.append(grama)

    t_array.append(t)

# Plot dos graficos de cada população com sua respectiva cor
plt.plot(t_array, onca_array, color='yellow' )
plt.plot(t_array, lobog_array, color='orange' )
plt.plot(t_array, veado_array,color='brown' )
plt.plot(t_array, caititu_array,color='black' )
plt.plot(t_array, morango_array,color='red')
plt.plot(t_array, grama_array,color='green' )

plt.grid(True) # Grid no gráfico
plt.title('População - Presa-Predador') # Título do gráfico
plt.legend(['Onça', 'Lobo Guará', 'Veado Campeiro', 'Caititu', 'Morango
Silvestre', 'Grama'], loc = 'upper right') # Define a legenda do
gráfico
plt.show()

# Plot dos gráficos Phase - Space de cada relação predador - presa
""" plt.plot(onca_array,veado_array)
plt.xlabel('Onça')
plt.ylabel('Veado Campeiro')
plt.show()
plt.plot(onca_array,veado_array)
plt.xlabel('Onça')
plt.ylabel('Lobo guará')
plt.show()
```

```
plt.plot(lobog_array,veado_array)
plt.xlabel('Lobo guará')
plt.ylabel('Veado Campeiro')
plt.show()
plt.plot(lobog_array,caititu_array)
plt.xlabel('Lobo guará')
plt.ylabel('Caititu')
plt.show()
plt.plot(lobog_array,morango_array)
plt.xlabel('Lobo guará')
plt.ylabel('Morango')
plt.show()
plt.plot(caititu_array,morango_array)
plt.xlabel('Caititu')
plt.ylabel('Morango')
plt.show()
plt.plot(caititu_array,grama_array)
plt.xlabel('Caititu')
plt.ylabel('Grama')
plt.show()
plt.plot(veado_array,grama_array)
plt.xlabel('Veado Campeiro')
plt.ylabel('Grama')
plt.show() """
```