TRABAJO ICSR
(SISTEMAS COMPLEJOS)

Raúl Varela Ferrando

02/05/2023

1. Introducción

Este trabajo consiste en el desarrollo de un modelo de NetLogo como representación de un sistema complejo. En mi caso, dado que todas las ideas que me surgían para este tipo de sistemas eran demasiado complejos como para implementarlos en este programa con mi capacidad, he decidido partir de un modelo ya existente en la librería de modelos, en concreto el modelo "Wolves Sheep Grass". En este modelo, como su propio nombre indica, tenemos un ecosistema de ovejas que se alimentan de césped y cuyos depredadores son los lobos, el cuál he expandido creando otro depredador, las águilas, para estudiar el impacto de este en el sistema. He creado dos tipos de sistemas, el primero en el que tenemos una cadena en la cual las águilas cazan a los lobos y a las ovejas, y los lobos solamente a las ovejas; y otro modelo en el que las águilas y los lobos tienen la misma probabilidad de comerse entre ellos. Es conveniente señalar que lo importante en estos sistemas no es el tipo de animal que he elegido ya que me he ceñido a las figuras disponibles para las tortugas, si no el concepto de cadena alimenticia que acabo de comentar.

2. Modelos

2.1 Modelo "Eagles Wolves Sheep Grass"

En este modelo como hemos comentado tenemos cuatro tipos de agentes, tres de ellos móviles (tortugas) y uno de ellos estático (parcelas):

- Grass: Es un agente de tipo parcela el cuál sirve como alimento a uno de los agentes de tipo tortuga, las ovejas (*sheep*).
- Sheep: Agente de tipo tortuga que sirve como alimento tanto a las águilas (eagles) como a los lobos (wolves).
- Wolves: Agente de tipo tortuga depredador de las ovejas que a su vez es alimento de las águilas.
- Eagles: Agente de tipo tortuga que es el eslabón superior de la cadena alimenticia, es decir, no tiene depredadores.

El funcionamiento del sistema sería el siguiente, en un inicio los agentes tipo parcela elegirán aleatoriamente un color entre el verde o el marrón, las parcelas color verde serán el césped (*Grass*), mientras que las parcelas de color marrón serán parcelas "vacías", las cuáles tendrán un contador que marcará el momento en el que puede volver a crecer césped en ellas. Los agentes tipo tortuga tienen funcionamientos similares, todas cuentan con una cantidad inicial de energía, la cuál gastarán al moverse o al reproducirse, o regenerarán al alimentarse en función de cuál sea el alimento que hayan ingerido. Para

asemejar nuestro sistema mejor a la realidad el alimento que más energía regenerará serán los lobos (*Wolves*), mientras el que menos será el césped.

En cada instante de tiempo haremos crecer césped en las parcelas cuyo contador haya llegado a cero, mientras que a las tortugas les haremos seguir una serie de órdenes según sean ovejas (*Sheep*), lobos (*Wolves*) o águilas (*Eagles*). Primero haremos que se muevan aleatoriamente, después que se alimenten de los otros respectivos agentes, y por último, que se reproduzcan, quedando con una cierta cantidad de energía según las órdenes que hayan podido cumplir, dado que no podrán alimentarse si no tienen cerca su respectivo alimento. Todas aquellas tortugas cuya energía quede en negativo después de seguir todo este proceso, desaparecerán de nuestro entorno. El código puede verse en el Anexo al final del documento

En la *Figura 1* se puede apreciar la interfaz de ejecución de nuestro modelo, donde aparecen todas las variables que hemos permitido controlar por parte del observador:

- initial-number-sheep / initial-number-wolves / initial-number-eagles: Como sus nombres indican, estas variables dan valor al número inicial de ovejas, lobos o águilas respectivamente.
- grass-regrowth-time: El tiempo necesario para que vuelva a crecer césped en una parcela vacía.
- eat-grass-gain / eat-sheep-gain / eat-wolf-gain: Representan la cantidad de energía que restauran al alimentarse de los distintos agentes.
- sheep-reproduce / wolf-reproduce / eagle-reproduce: Umbral de energía necesario para que las tortugas puedan reproducirse.

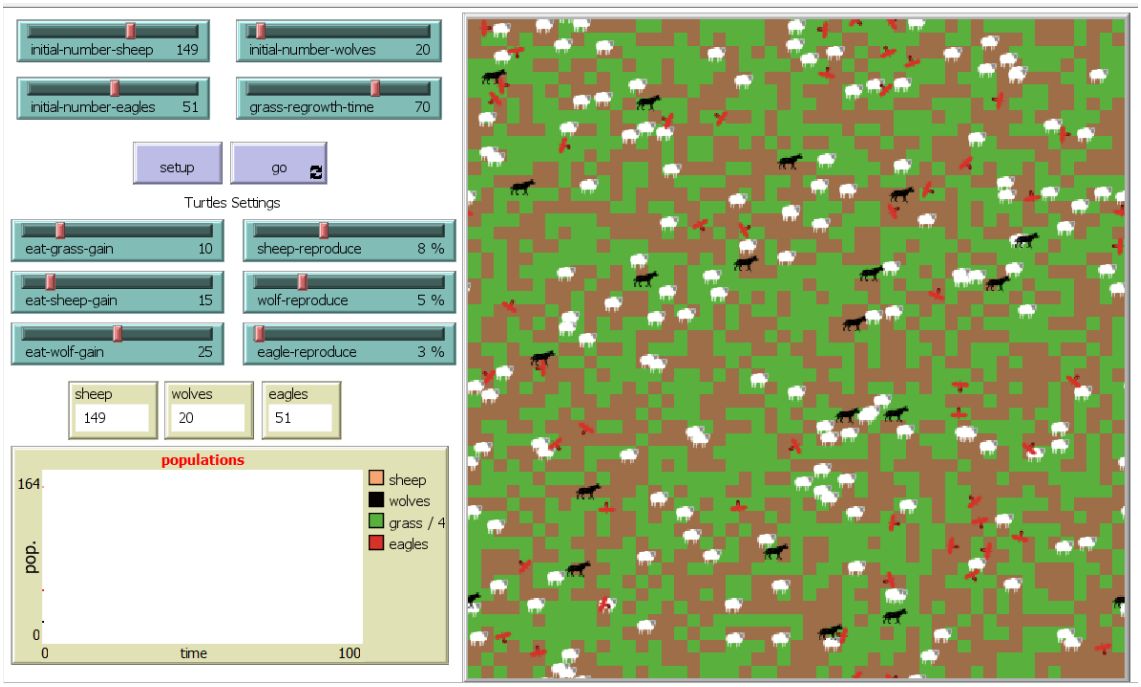

Figura 1: Interfaz de ejecución en NetLogo del modelo "Eagles Wolves Sheep Grass".

Como se puede observar, en la esquina inferior izquierda tenemos tres contadores en las que podemos visualizar la población de los distintos agentes tipo tortuga que tenemos en

cada instante, además de una gráfica para poder apreciar su comportamiento a lo largo del tiempo transcurrido desde el inicio.

2.2 Modelo “2.0 Eagles Wolves Sheep Grass”

Este modelo es similar al anterior con una pequeña diferencia, en este modelo es igual de probable que las águilas cacen a los lobos o viceversa, introduciendo así un factor de competitividad en el ecosistema. La energía obtenida para cada uno de ellos varía, a las cuáles les he asignado unos valores para que sean equitativas entre depredadores. Las variables a controlar serían prácticamente las mismas que en el primer modelo, solo que añadiendo la variable “eat-eagle-gain”, que representa la cantidad de energía ganada al cazar un águila, como era de esperar.

En la siguiente figura se puede observar la interfaz de ejecución de nuestro modelo, y el código, al igual que el anterior, se encuentra en el Anexo al final del documento.

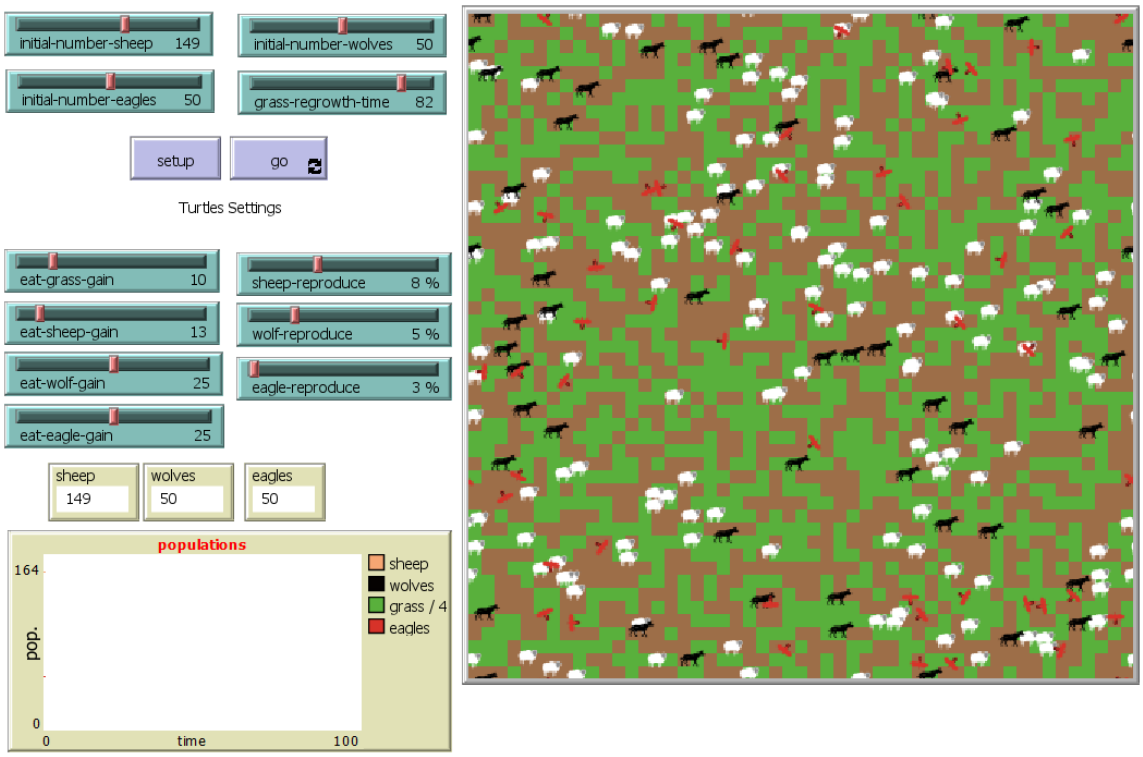

Figura 2: Interfaz de ejecución en NetLogo del modelo “2.0 Eagles Wolves Sheep Grass”.

3. Resultados

En este apartado haremos un estudio de algunas de las propiedades de nuestro sistema en función de las distintas variables que hemos enunciado anteriormente. Las magnitudes a estudiar son las distintas poblaciones de agentes, como era de esperar, de formar que experimentaremos diferencias en las proporciones de estas en nuestro ecosistema dependiendo del modelo y de los cambios aplicados.

3.1 Modelo “Eagles Wolves Sheep Grass”

Tras varias simulaciones en nuestro modelo hemos llegado a la conclusión que el estado estable de nuestro sistema viene dado por la siguiente gráfica:

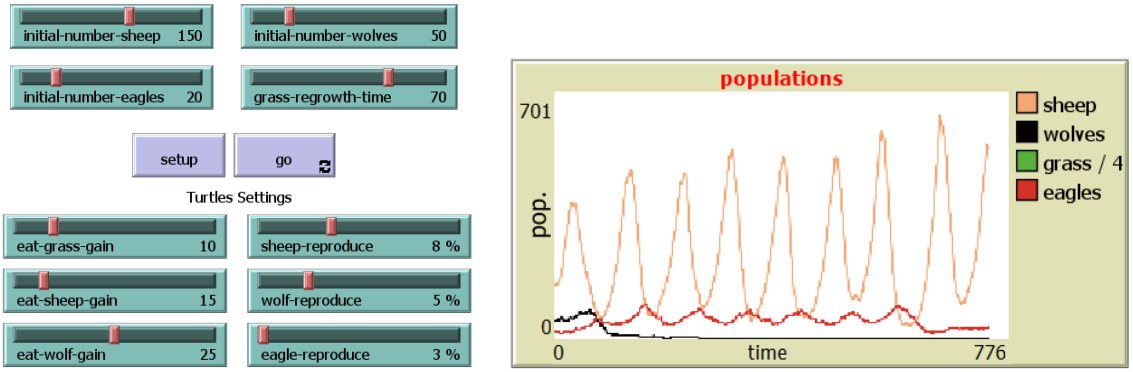

Figura 3: Estado estable del modelo “Eagles Wolves Sheep Grass”.

Los valores de las variables se han elegido de forma que se asemejen lo mejor posible al ecosistema real. El número inicial de ovejas debe ser elevado dado que es alimento de dos depredadores, mientras que el número de lobos debe ser menor. El número inicial de águilas se ha elegido menor aún que el de lobos dado que es depredador de ambas especies. Para la energía obtenida en cada caso he seguido el mismo principio, dado que el césped es alimento de las ovejas será el que menos energía proporcione, seguido de las ovejas, que es alimento de los lobos y las águilas, y por último, la energía que proporciona un lobo, que es mayor que las anteriores dado que existen en menor proporción. El umbral de reproducción de cada especie se ha elegido de la misma forma.

Como se puede observar, al haber dos depredadores, uno de los cuáles (los lobos) solo pueden alimentarse del primer eslabón de la cadena alimenticia, en todos los casos que hemos estudiado en los que llegamos a un estado estable, la población de lobos acaba tendiendo a cero, excepto en el caso de que no tengamos águilas, como era de esperar. Este modelo tiende a convertirse en un ecosistema dual presa-depredador, el cuál puede avanzar indefinidamente en el tiempo.

Como se puede observar las poblaciones se regulan entre ellas mismas, al haber un pico en la población de ovejas, las águilas tienen más alimento disponible y crece su población; como resultado, la población de ovejas decrece, llegando al estado inverso. En este punto las águilas disponen de poco alimento, por lo que van muriendo, permitiendo así a las ovejas reproducirse y llegar de nuevo a un pico en la población de ovejas. Los picos no son exactamente iguales debido a la aleatoriedad del movimiento de las tortugas, de ahí la asimetría.

La emergencia de este estado depende básicamente de las variables que nos dan las condiciones iniciales, es decir, la cantidad de ovejas y del tiempo necesario para que vuelva a crecer el césped en una parcela vacía, dado que las cantidades iniciales de lobos y águilas lo único que varían es el tipo de depredador que perdura en el tiempo, pero siempre llegamos al estado estable que hemos descrito anteriormente.

En el caso de tener muy pocas ovejas en el estado inicial hace que los depredadores no tengan alimento y en la gran mayoría de casos, acabamos en un ecosistema sin depredadores como aparece en la siguiente figura.

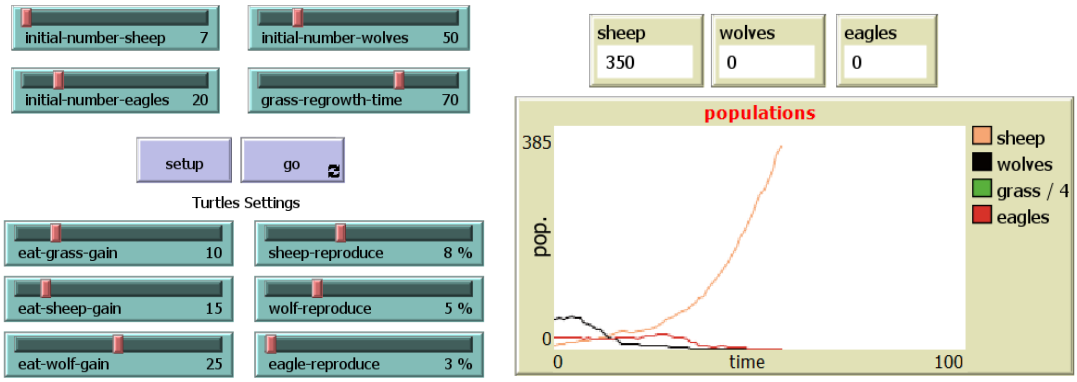

Figura 4: Ecosistema sin depredadores.

Si dejamos fijas las cantidades iniciales de los distintos agentes de tipo tortuga y variamos el tiempo necesario para que el césped vuelva a crecer en las parcelas vacías llegamos a un punto en el que, si este tiempo es demasiado pequeño (menor que 30 aproximadamente), las ovejas crecen a un ritmo prácticamente exponencial, acabando en un ecosistema con sobrepoblación de estas en un periodo de tiempo muy corto. En la siguiente figura se pueden apreciar los valores asignados y la gráfica resultante en este caso.

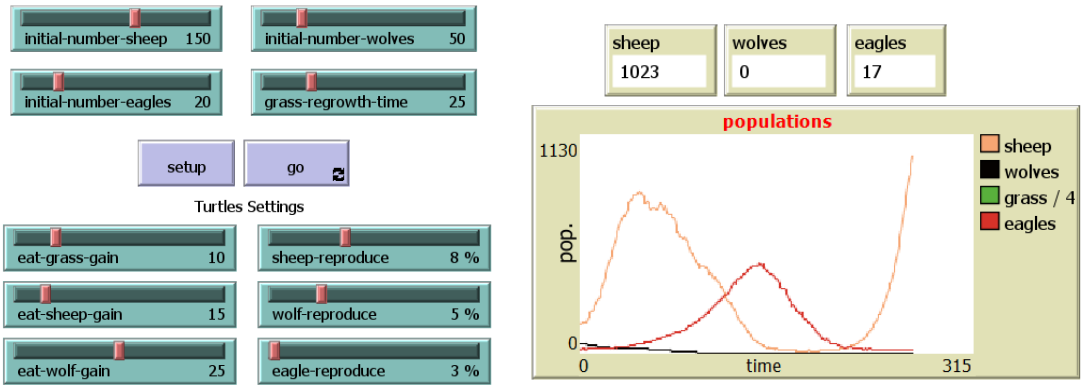

Figura 5: Ecosistema con sobrepoblación de ovejas.

En el caso de aumentar mucho la energía que aporta el comer césped a las ovejas llegamos exactamente al mismo caso, una sobrepoblación de ovejas, al igual que si reducimos demasiado el umbral necesario para que se reproduzcan las ovejas, por lo que no adjuntaré ninguna gráfica asociada a estos casos.

Otra situación sería si aumentásemos la energía obtenida al cazar una oveja, esto va aportando cada vez más inestabilidad al sistema, manteniéndose estable durante menos tiempo y con picos de población más agudos, llegando finalmente a un estado en el que se han extinguido todos los agentes de tipo tortuga, como se muestra a continuación.

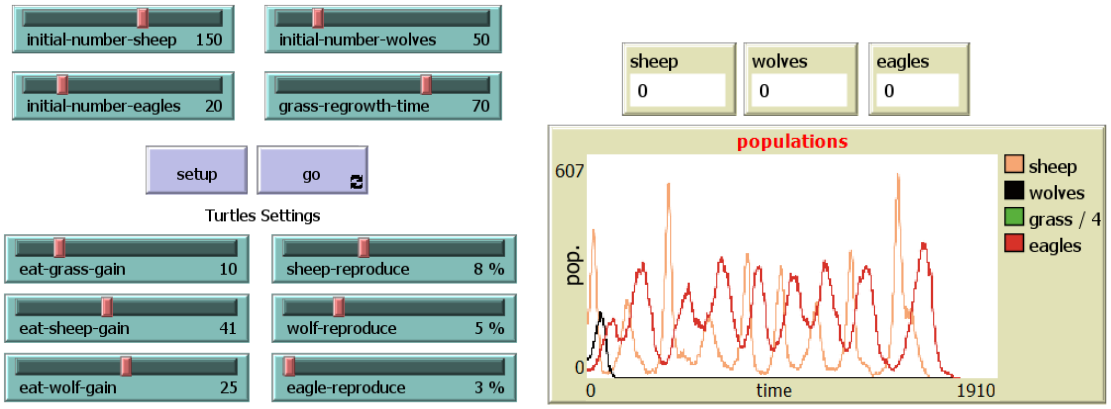

Figura 6: Ecosistema extinguido.

Por último, he estudiado el efecto de reducir el umbral de energía necesario para reproducirse tanto de los lobos como de las águilas. En el primer caso no tenemos efecto, dado que al haber existencia de águilas, los lobos tienden a desaparecer por mucho que fomentemos su reproducción, mientras que en el caso de las águilas acabamos en un ecosistema sin depredadores al igual que en el caso de tener pocas ovejas inicialmente.

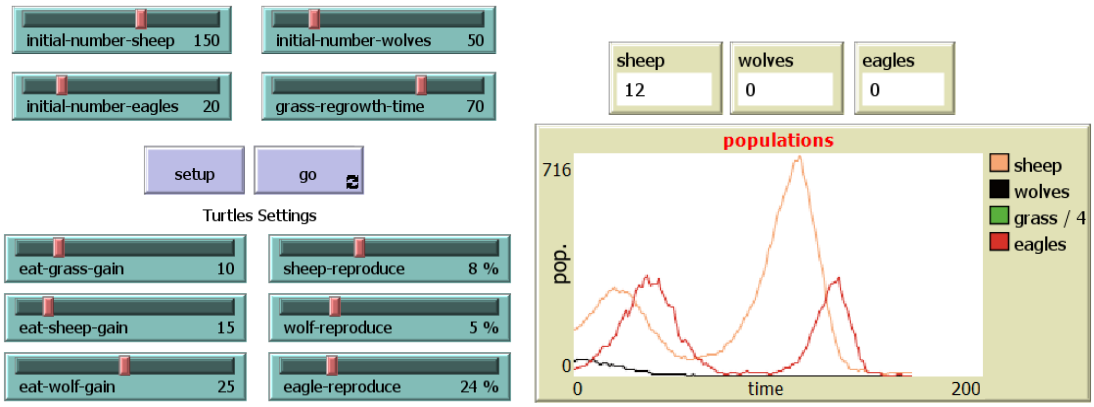

Figura 7: Ecosistema sin depredadores.

3.2 Modelo “2.0 Eagles Wolves Sheep Grass”

En este modelo, eligiendo valores para algunas variables distintos a los del anterior modelo para que haya igualdad de condiciones para ambos depredadores, llegamos a la conclusión de que el estado estable de nuestro modelo es el siguiente:

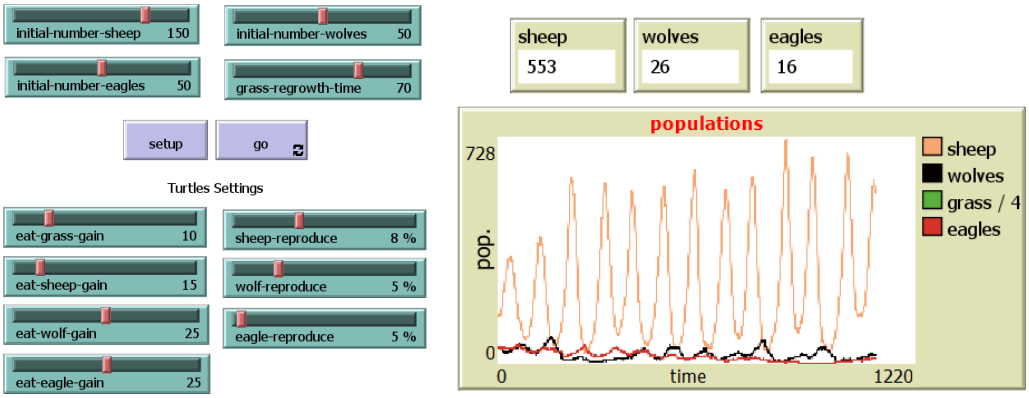

Figura 8: Estado estable del modelo "2.0 Eagles Wolves Sheep Grass".

En este caso, a diferencia del anterior se puede observar como las poblaciones de ambos depredadores perduran en el tiempo compitiendo entre ellos, de forma que en este modelo sí conseguimos un sistema sustancialmente diferente del presentado en el modelo por defecto de NetLogo.

Estudiando el efecto de la población inicial de lobos o de águilas llegamos a la conclusión que por debajo de un umbral (aproximadamente 10), se acaban extinguiendo para ambos casos, quedando el ecosistema dual que representamos en el estado estable del anterior modelo. Adjuntaré solo la gráfica correspondiente al caso de las águilas ya que como he comentado, al reducir la población de lobos llegamos al estado complementario.

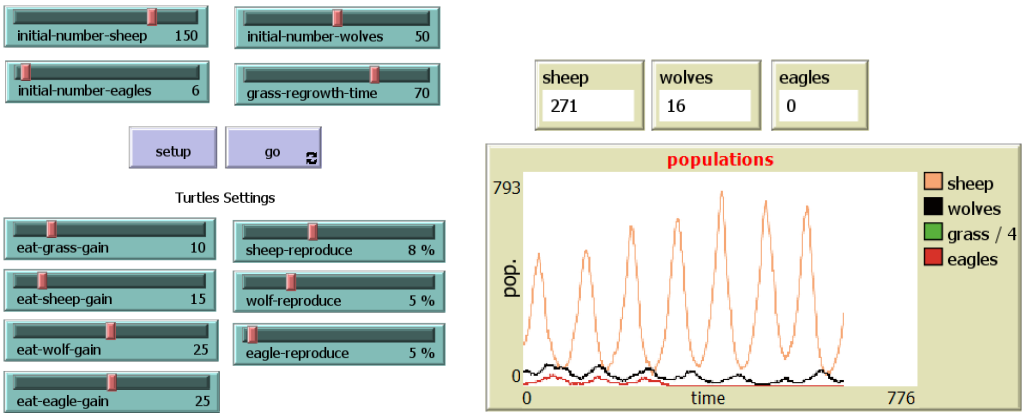

Figura 8: Ecosistema dual presa-depredador.

Para el caso de disminuir mucho la población de ovejas no tenemos ningún cambio sustancial, acabando siempre en el estado estable del sistema. Aunque cabe señalar que en el caso de no tener ovejas, el ecosistema no perdura, es decir, un ecosistema en el que los lobos y las águilas se cacen entre ellos no es viable en este modelo.

Al variar el tiempo necesario para que el césped vuelva a crecer llegamos a los mismos resultados que teníamos anteriormente, si lo reducimos demasiado acabamos en un ecosistema con sobrepoblación de ovejas (al igual que pasaría si aumentamos la energía obtenida al alimentarse del césped), mientras que si lo aumentamos demasiado acabamos en un ecosistema donde todas las poblaciones se han extinguido debido a que las ovejas no tienen alimento suficiente, y como consecuencia, los siguientes eslabones de la cadena

alimenticia tampoco (el mismo resultado que obtenemos al disminuir la energía obtenida al alimentarse del césped).

Al estudiar el efecto que tiene en nuestro sistema variar la energía obtenida al alimentarse de una oveja obtenemos los siguientes resultados:

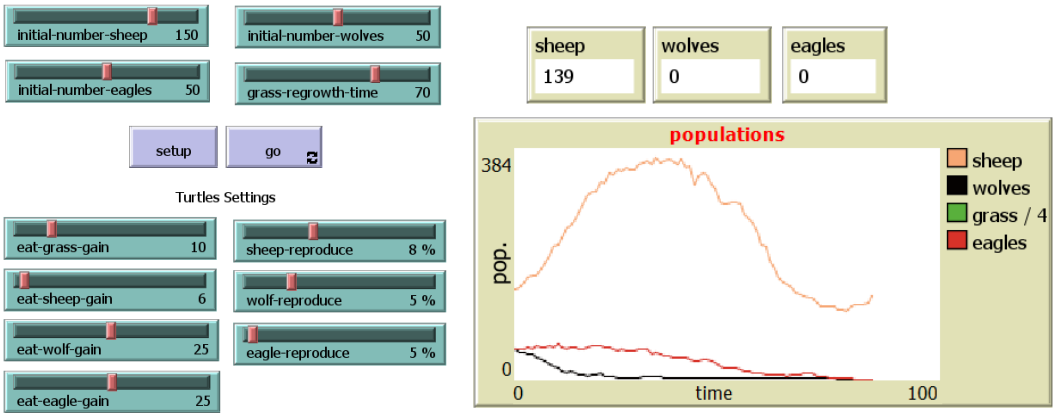

Figura 10: Ecosistema sin depredadores.

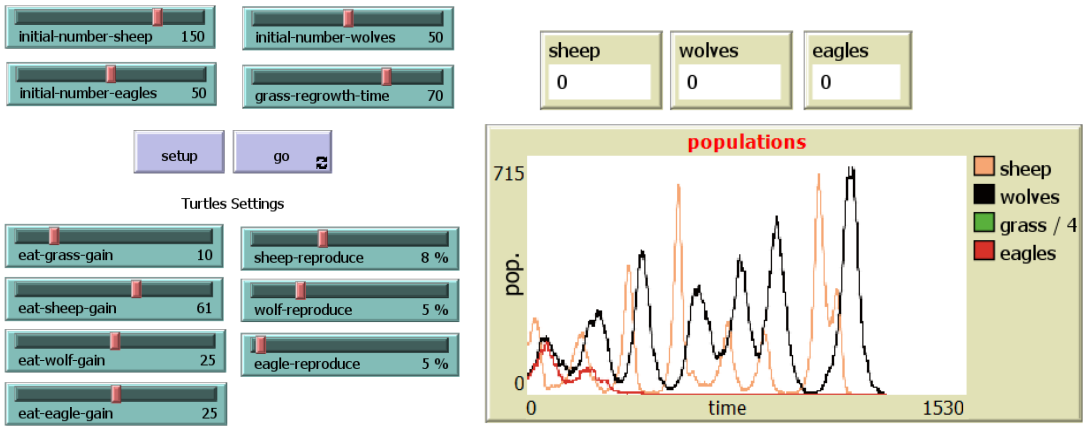

Figura 11: Ecosistema extinguido.

Al disminuirlo demasiado los depredadores no obtienen energía suficiente para reproducirse y se acaban extinguiendo (Figura 10), mientras que al aumentarlo demasiado, aportamos mucha inestabilidad al sistema, llegando a picos de población muy agudos, y acabando en la extinción de todos los agentes del sistema de tipo tortuga.

Por último, al variar tanto la energía obtenida al alimentarse de un lobo o de un águila, como al variar el umbral de reproducción para los lobos o para las águilas, estamos favoreciendo la existencia de uno de los depredadores frente al otro, llegando siempre al sistema dual presa-depredador que hemos visto anteriormente, por lo que no adjuntaré ninguna gráfica para estos casos para evitar la redundancia.

Anexo:

Código “Eagles Wolves Sheep Grass”

```
globals [ max-sheep max-wolves max-eagles]

breed [ sheep a-sheep ]
breed [ wolves wolf ]
breed [ eagles eagle]

turtles-own [ energy ]

patches-own [ countdown ]

to setup
  clear-all
  set max-sheep 1000
  set max-wolves 1000
  set max-eagles 1000
  ask patches [
    set pcolor one-of [ green brown ]
    ifelse pcolor = green [ set countdown grass-regrowth-time ]
    [ set countdown random grass-regrowth-time ]
    ]

  create-sheep initial-number-sheep
  [
    set shape  "sheep"
    set color white
    set size 1.5
    set label-color blue - 2
    set energy random (2 * eat-grass-gain)
    setxy random-xcor random-ycor
  ]

  create-wolves initial-number-wolves
  [
    set shape "wolf"
    set color black
    set size 2
    set energy random (2 * eat-sheep-gain)
    setxy random-xcor random-ycor
  ]

  create-eagles initial-number-eagles
  [
    set shape "hawk"
    set color red
    set size 1.5
    set energy random (2 * eat-wolf-gain)
    setxy random-xcor random-ycor
  ]

  reset-ticks
end
```

```
to go
  if not any? turtles [ stop ]
  if not any? wolves and not any? eagles [ user-message "No predators in the ecosystem." stop ]
  if count sheep > max-sheep [ user-message "The sheep have inherited the earth" stop ]
  if count wolves > max-wolves [ user-message "The wolves have inherited the earth" stop ]
  if count eagles > max-eagles [ user-message "The eagles have inherited the earth" stop ]
  ask sheep [
    move
    set energy energy - 1
    eat-grass
    death
    reproduce-sheep
  ]
  ask wolves [
    move
    set energy energy - 1
    eat-sheep
    death
    reproduce-wolves
  ]
  ask eagles [
    move
    set energy energy - 1
    eat-sheep
    eat-wolf
    death
    reproduce-eagles
  ]

  ask patches [ grow-grass ]



  tick
end

to move
  rt random 50
  lt random 50
  fd 1
end

to eat-grass
  if pcolor = green [
    set pcolor brown
    set energy energy + eat-grass-gain
  ]
end

to reproduce-sheep  ; sheep procedure
  if random-float 100 < sheep-reproduce [
    set energy (energy / 2)
    hatch 1 [ rt random-float 360 fd 1 ]
  ]
end

to reproduce-wolves  ; wolf procedure
  if random-float 100 < wolf-reproduce [
    set energy (energy / 2)
    hatch 1 [ rt random-float 360 fd 1 ]
  ]
end



to reproduce-eagles  ; wolf procedure
  if random-float 100 < eagle-reproduce [
    set energy (energy / 2)
    hatch 1 [ rt random-float 360 fd 1 ]
  ]
end

to eat-sheep  ; wolf procedure
  let prey one-of sheep-here
  if prey != nobody  [
    ask prey [ die ]
    set energy energy + eat-sheep-gain
  ]
end
```

```
to eat-wolf  ; wolf procedure
  let prey one-of wolves-here
  if prey != nobody  [
    ask prey [ die ]
    set energy energy + eat-wolf-gain
  ]
end

to death
  if energy < 0 [ die ]
end


to grow-grass
  if pcolor = brown [
    ifelse countdown <= 0
      [ set pcolor green
        set countdown grass-regrowth-time ]
      [ set countdown countdown - 1 ]
  ]
end
```

Código “2.0 Eagles Wolves Sheep Grass”

```
globals [ max-sheep max-wolves max-eagles]

breed [ sheep a-sheep ]
breed [ wolves wolf ]
breed [ eagles eagle]

turtles-own [ energy ]
patches-own [ countdown ]

to setup
  clear-all
  set max-sheep 1000
  set max-wolves 1000
  set max-eagles 1000
  ask patches [
    set pcolor one-of [ green brown ]
    ifelse pcolor = green [ set countdown grass-regrowth-time ]
    [ set countdown random grass-regrowth-time ]
    ]

  create-sheep initial-number-sheep
  [
    set shape  "sheep"
    set color white
    set size 1.5
    set label-color blue - 2
    set energy random (2 * eat-grass-gain)
    setxy random-xcor random-ycor
  ]


  create-wolves initial-number-wolves
  [
    set shape "wolf"
    set color black
    set size 2
    set energy random (2 * eat-sheep-gain)
    setxy random-xcor random-ycor
  ]
```

```
  create-eagles initial-number-eagles
  [
    set shape "hawk"
    set color red
    set size 1.5
    set energy random (2 * eat-wolf-gain)
    setxy random-xcor random-ycor
  ]

  reset-ticks
end

to go
  if not any? turtles [ stop ]
  if not any? wolves and not any? eagles [user-message "No predators in this ecosystem." stop ]
  if count sheep > max-sheep [ user-message "The sheep have inherited the earth" stop ]
  if count wolves > max-wolves [ user-message "The wolves have inherited the earth" stop ]
  if count eagles > max-eagles [ user-message "The eagles have inherited the earth" stop ]
  ask sheep [
    move
    set energy energy - 1
    eat-grass
    death
    reproduce-sheep
  ]
  ask wolves [
    move
    set energy energy - 1
    eat-sheep
    eat-eagle
    death
    reproduce-wolves
  ]
  ask eagles [
    move
    set energy energy - 1
    eat-sheep
    eat-wolf
    death
    reproduce-eagles
  ]

  ask patches [ grow-grass ]

  tick
end

to move
  rt random 50
  lt random 50
  fd 1
end

to eat-grass
  if pcolor = green [
    set pcolor brown
    set energy energy + eat-grass-gain
  ]
end


to reproduce-sheep
  if random-float 100 < sheep-reproduce [
    set energy (energy / 2)
    hatch 1 [ rt random-float 360 fd 1 ]
  ]
end
```

```
to reproduce-wolves
  if random-float 100 < wolf-reproduce [
    set energy (energy / 2)
    hatch 1 [ rt random-float 360 fd 1 ]
  ]
end

to reproduce-eagles
  if random-float 100 < eagle-reproduce [
    set energy (energy / 2)
    hatch 1 [ rt random-float 360 fd 1 ]
  ]
end

to eat-sheep
  let prey one-of sheep-here
  if prey != nobody  [
    ask prey [ die ]
    set energy energy + eat-sheep-gain
  ]
end

to eat-wolf  ; wolf procedure
  if random-float 100 < 50 [ let prey one-of wolves-here
  if prey != nobody  [
    ask prey [ die ]
    set energy energy + eat-wolf-gain
  ]
  ]
end

to eat-eagle  ; wolf procedure
  if random-float 100 < 50 [ let prey one-of eagles-here
  if prey != nobody  [
    ask prey [ die ]
    set energy energy + eat-eagle-gain
  ]
  ]
end

to death
  if energy < 0 [ die ]
end

to grow-grass
  if pcolor = brown [
    ifelse countdown <= 0
      [ set pcolor green
        set countdown grass-regrowth-time ]
      [ set countdown countdown - 1 ]
  ]
end
```