

TUTORIAL 3 DE NETLOGO

Realizado Por: Michael Santiago Díaz 160002613

Entregado A: Ph.D Ángel Cruz

Universidad de los llanos

15 de noviembre de 2015

Este tutorial lo llevará a través del proceso de construcción de un modelo completo, construido por etapas y con cada paso explicado a lo largo del camino.

Agentes y procedimientos

En el Tutorial 2 aprendió a utilizar el centro de comando y los monitores de agentes para inspeccionar y modificar los agentes y hacer que ellos hagan cosas. Ahora está listo para aprender acerca del verdadero corazón de un modelo NetLogo: la ficha de procedimientos.

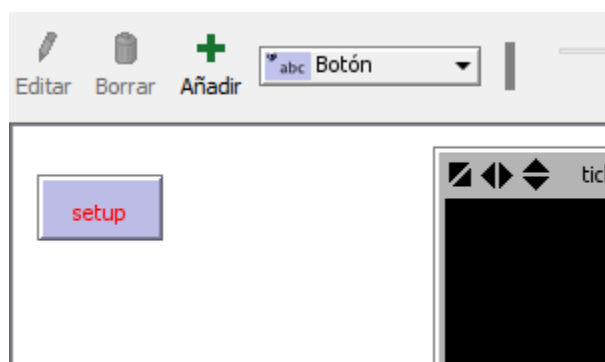
Usted ya ha utilizado diferentes tipos de agentes a los que se pueden dar comandos en NetLogo: parches, tortugas, enlaces, y el observador. Los parches son estacionarios y están arreglados en una cuadrícula. Las tortugas se mueven sobre esa cuadrícula. Los Enlaces conectan a dos tortugas. El observador (observer) supervisa todo lo que está pasando y hace todo aquello que las tortugas, los parches y los enlaces no pueden hacer por sí mismos.

Todos los cuatro tipos de agentes NetLogo pueden ejecutar comandos. Todos tres pueden además ejecutar "procedimientos" ("procedures"). Un procedimiento combina una serie de comandos NetLogo en un único nuevo comando definido por usted.

Ahora usted aprenderá a escribir procedimientos que hacen que las tortugas se muevan, coman, se reproduzcan y mueran. Usted también aprenderá cómo hacer monitores, sliders (controles deslizantes) y gráficas. El modelo que vamos a construir es un modelo simple de un ecosistema que parte del modelo de depredación Lobo Oveja del Tutorial 1.

Haciendo el botón setup

- Para iniciar un nuevo modelo, seleccione "New.^{en} el menú File. Luego empiece la creación de un botón setup (configuración):
- Haga clic en el ícono "Button.^{en} la parte superior de la ficha de la interfaz.
- Haga clic en donde usted desea que aparezca el botón dentro del área blanca vacía de la interfaz.
- Se abre un cuadro de diálogo para editar el botón. Escriba setup en la casilla marcada con "Commands".
- Pulse el botón OK cuando haya terminado; el cuadro de diálogo se cierra.



- Si desea ver el actual mensaje de error, haga clic en el botón.
- Ahora vamos a crear el procedimiento "setup", de modo que el mensaje de error desaparecerá:
- Cambiase a la pestaña Procedures.
- Escriba lo siguiente:

```

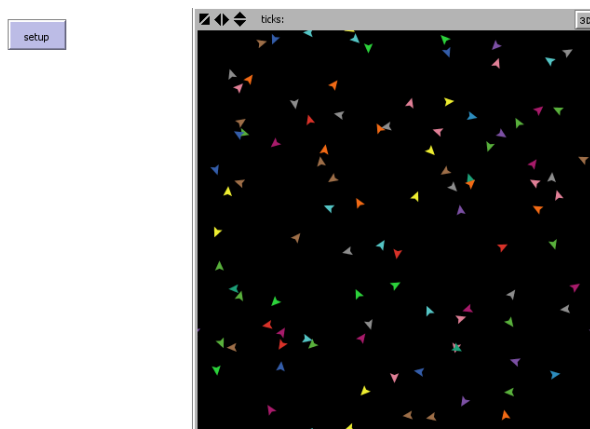
to setup
  clear-all
  create-turtles 100
  ask turtles[ setxy random-xcor random-ycor]
end

```

Tenga en cuenta que las líneas tienen diferentes sangrías. La mayoría de las personas encuentra útil sangrar su código como en este, sin embargo no es obligatorio. Esto hace más fácil leer y cambiar el código. Su procedimiento inició con la palabra `to` y terminó con la palabra `end`. Cada nuevo procedimiento que cree comenzará y terminará con estas dos palabras.

```
to setup
  clear-all
  create-turtles 100
  ask turtles [ setxy random-xcor random-ycor ]
end
```

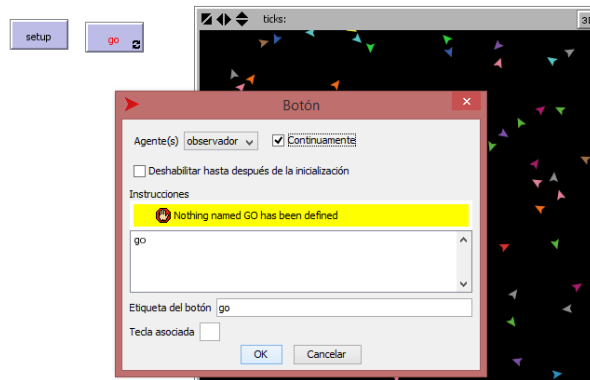
- Cuando haya terminado de escribir cambie a la interfaz y presione el botón `setup` que hizo anteriormente. Verá las tortugas dispersas alrededor del mundo:



Haciendo el botón `go`

Ahora haga un botón llamado `go`. Siga los mismos pasos que utilizó para hacer el botón `setup`, excepto:

- En `Commands` introduzca `go` en lugar de `setup`.
- Marque `forever`^{en} la casilla de verificación del diálogo de edición.



- A continuación, agregue un procedimiento Go en la pestaña de procedimientos (Procedures):

```
to go
  move-turtles
end
```

- Agregue el procedimiento move-turtles después del procedimiento go:

```
to go
  move-turtles
end

to move-turtles
  ask turtles[
    right random 360
    forward 1
  ]
end
```

PARCHES Y VARIABLES

- Regresemos al procedimiento setup. Podemos reescribirlo de la siguiente manera:

```
to setup
  clear-all
```

```
    setup-patches
    setup-turtles
end
```

- La nueva definición de setup se refiere a dos nuevos procedimientos. Para definir setup-patches añade lo siguiente:

```
to setup-patches
  ask patches [ set pcolor green ]
end
```

El procedimiento setup-patch establece en verde el color de cada parche al comenzar. (La variable del color de la tortuga es color; la del parche es pcolor.)

La única parte que permanecen en nuestro nuevo 'setup' que aún está indefinida es setup-turtles.

- Añade también este procedimiento:

```
to setup-turtles
  create-turtles 100
  ask turtles [ setxy random-xcor random-ycor ]
end
```

¿Advirtió que el nuevo procedimiento setup-turtles tiene la mayoría de los mismos comandos que el antiguo procedimiento setup?

```

to setup
  clear-all
  setup-patches
  setup-turtles
end

to go
  move-turtles
end

to move-turtles
  ask turtles [
    right random 360
    forward 1
  ]
end

to setup-patches
  ask patches [ set pcolor green ]
end

to setup-turtles
  create-turtles 100
  ask turtles [ setxy random-xcor random-ycor ]
end

```

- Vuelva a la pestaña de Interfaz.
- Pulse el botón setup.

Voilà! Aparece un exuberante paisaje NetLogo completo con las tortugas y los parches verdes:



VARIABLES DE TORTUGA

De modo que tenemos algunas tortugas corriendo en un paisaje, pero nada están haciendo con ello. Vamos a añadir algo de interacción entre las tortugas y los parches.

Haremos a las tortugas comer "pasto" (los parches verdes), reproducirse y morir. La hierba crecerá gradualmente después de ser comida.

Necesitamos una manera de controlar cuando una tortuga se reproduce y cuándo muere. Vamos a determinarlo haciendo el seguimiento de la cantidad de "energía" que tiene cada una de las tortugas. Para hacer esto necesitamos añadir una nueva variable en la tortuga.

Ya ha visto la construcción de variables de la tortuga tales como color. Para crear una nueva variable de la tortuga, añadimos una declaración `turtles-own` (tortugas-propia) en la parte superior de la pestaña de procedimientos, antes que todos los procedimientos. Llamela `energy` (energía):

```
turtles-own [energy]
```

```
to go
  move-turtles
  eat-grass
end
```

Vamos a utilizar esta recientemente definida variable (`energy`) para permitirle a las tortugas comer.

- Cambie a la pestaña de Procedimientos.
- Reescriba el procedimiento `go` de la siguiente manera:

```
to go
  move-turtles
  eat-grass
end
```

- Agrega el nuevo procedimiento `eat-grass` (comer-pasto):

```
to eat-grass
  ask turtles [
    if pcolor = green [
```

```
        set pcolor black
        set energy (energy + 10) ]
    ]
end
```

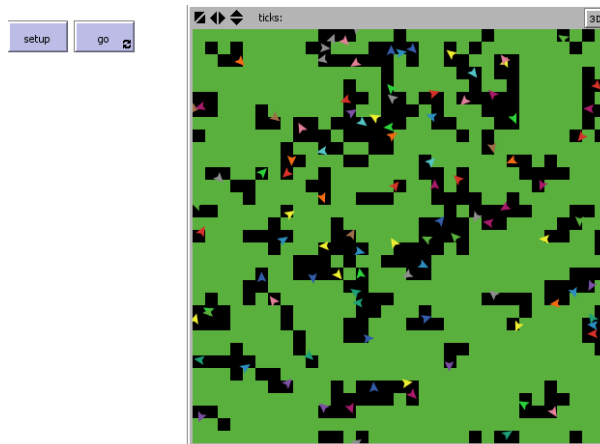
A continuación vamos a hacer que el movimiento de las tortugas utilice un poco de la energía de las tortugas.

- Reescriba move-turtles así:

```
to move-turtles
  ask turtles [
    right random 360
    forward 1
    set energy (energy - 1)
  ]
end
```

En la medida en que cada tortuga deambula pierde una unidad de energía a cada paso.

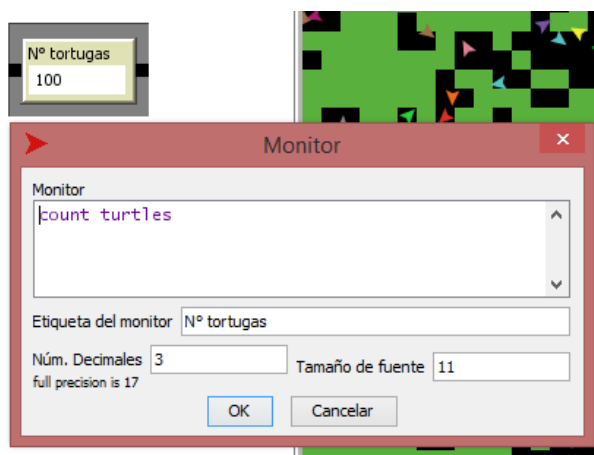
Ahora vaya a la interfaz y presione el botón setup y luego el botón go, verá los parches volverse negros a medida que las tortugas viajan sobre ellos.



MONITORES

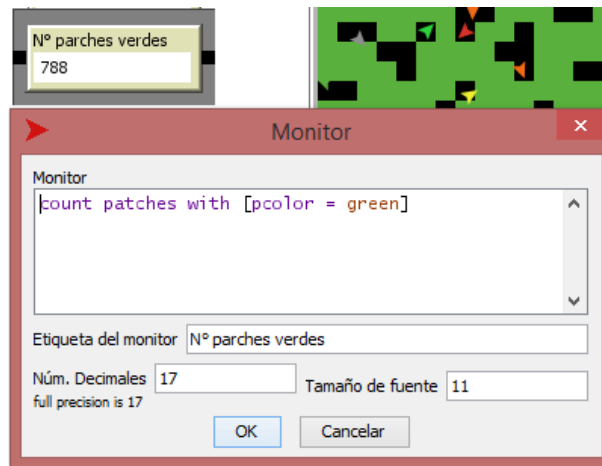
A continuación creará en la interfaz dos monitores con la barra de herramientas. (Usted los hace tal como a los controles deslizantes (sliders) y a los botones, usando el icono del monitor de la barra de herramientas.) Vamos a hacer ahora el primer monitor.

- Cree un monitor, utilizando el icono del monitor de la barra de herramientas, haga clic en un lugar abierto de la interfaz.
- Aparecerá un cuadro de diálogo.
- En el cuadro de diálogo escriba: count turtles (contar las tortugas, ver imagen inferior).
- Pulse el botón OK para cerrar el cuadro de diálogo.



Vamos a hacer ahora el segundo monitor:

- Cree un monitor, utilizando el icono del monitor de la barra de herramientas y haga clic en un lugar abierto de la interfaz.
- Aparecerá un cuadro de diálogo.
- En la sección Reportero del cuadro de diálogo escriba: count patches with [pcolor = green] (ver imagen inferior).
- En la sección Display name del cuadro de diálogo escriba: green patches
- Pulse el botón OK para cerrar el cuadro de diálogo.



INTERRUPTORES Y ETIQUETAS

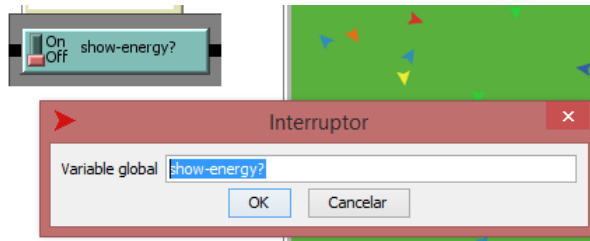
- Para crear un interruptor, haga clic en el interruptor de la barra de herramientas (en la pestaña Interfaz) y haga clic en un punto abierto en la interfaz.

Aparecerá un cuadro de diálogo.

- En la sección de variable Global del cuadro de diálogo escriba: show-energy? No olvide de incluir el signo de interrogación en el nombre. (Vea la imagen a continuación.)

```
to eat-grass
  ask turtles [
    if pcolor = green [
      set pcolor black
      set energy (energy + 10)
    ]
    ifelse show-energy?
      [ set label energy ]
      [ set label "" ]
  ]
end
```

Cuando el interruptor está encendido, verá la energía de cada tortuga incrementarse cada vez que come hierba. También verá su energía disminuir cada vez que se mueve.



MAS PROCEDIMIENTOS

Ahora nuestras tortugas están comiendo; vamos a hacer que también se reproduzcan y mueran. Además vamos a hacer que la hierba rebrote. Ahora mismo vamos a añadir estos tres comportamientos haciendo tres procedimientos separados. Uno para cada comportamiento.

- Vaya a la pestaña de procedimientos (Procedures).
- Reescribir el procedimiento `go` de la siguiente manera:

```
to go
  move-turtles
  eat-grass
  reproduce
  check-death
  regrow-grass
end
```

- Añada los procedimientos para `reproduce`, `check-death` y `regrow-grass`, como se indica a continuación:

```
to reproduce ;; reproducirse
  ask turtles [
    if energy > 50 [
      set energy energy - 50
      hatch 1 [ set energy 50 ]
    ]
  ]
end

to check-death ;; verificar muerte
```

```

ask turtles [
  if energy <= 0 [ die ]
]
end

to regrow-grass ;; rebrotar pasto
ask patches [
  if random 100 < 3 [ set pcolor green ]
]
end

```

- Vaya a la Interfaz y presione los botones setup y go.



Observación Las tortugas ahora mueren y se reproducen, además el pasto vuelve a crecer.

GRAFICACIÓN

- Cambiemos setup para llamar al nuevo procedimiento, do-plots (hacer gráficas), que estamos a punto de añadir:

```
to setup
  clear-all
  setup-patches
  setup-turtles
  do-plots
end
```

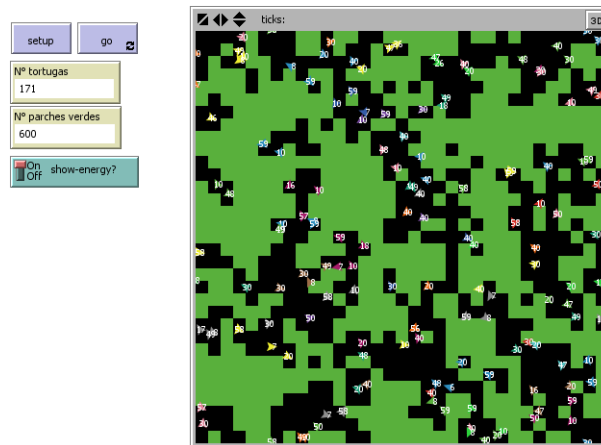
- Además, cambiemos go para llamar al procedimiento do-plots:

```
to go
  move-turtles
  eat-grass
  reproduce
  check-death
  regrow-grass
  do-plots
end
```

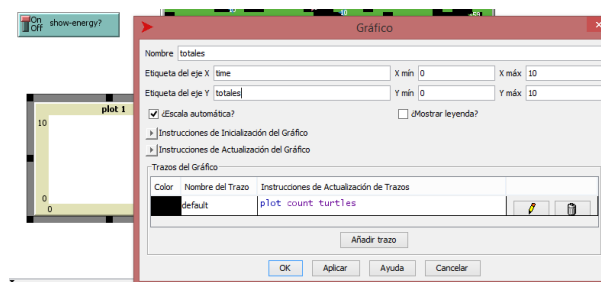
- Ahora agregue el nuevo procedimiento. Lo que estamos graficando será el número de tortugas y el número de parches verdes contra el tiempo. En cada paso de tiempo (una ejecución simple a través del procedimiento go) estos valores se añaden a la gráfica.

```
to do-plots
  set-current-plot "Totals"
  set-current-plot-pen "turtles"
  plot count turtles
  set-current-plot-pen "grass"
  plot count patches with [pcolor = green]
end
```

- Cree un gráfico, utilizando el icono plot de la barra de herramientas y haga clic en un lugar abierto en la interfaz.
- Establezca su nombre como "Totals" (ver imagen inferior)
- Establezca el eje de las X con la etiqueta "time"
- Establezca el eje Y con etiqueta "total"



- Con el cuadro de diálogo de Plot (Gráfica) aun abierto, pulse el botón 'Create' en el diálogo de Plot para crear una nueva curva.
- Introduzca el nombre de esta curva como "turtlesz pulse OK en el diálogo de .Enter Pen Name". (ver imagen inferior)
- Pulse nuevamente el botón 'Crear' en el cuadro de diálogo de nuevo, para crear una segunda curva nueva.
- Introduzca el nombre de esta curva como "grassz pulse OK en el cuadro de diálogo .Enter Pen Name". (ver imagen inferior)
- Seleccione el color de la curva y cámbielo a verde.
- Seleccione OK en el cuadro de diálogo de la Plot (Gráfica).



CONTADOR DE TICS

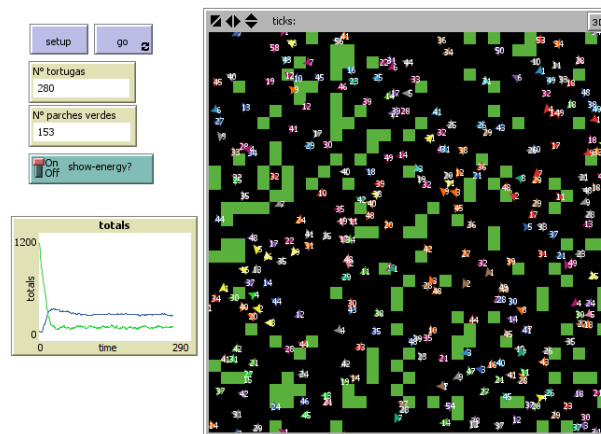
- Cambie el procedimiento go:

```

to go
  if ticks >= 500 [ stop ]
  move-turtles
  eat-grass
  reproduce
  check-death
  regrow-grass
  tick
  do-plots
end

```

- Ahora configure y ejecute el modelo.



ALGUNOS DETALLES ADICIONALES

- Cree una variable slider llamada "numero" usando el ícono monitor de la barra de herramientas y haga clic en un lugar limpio de la interfaz. Intente cambiando los valores mínimo y máximo en el slider.
- Luego, dentro de setup-turtles, en lugar de create-turtles 100 usted puede escribir:

```

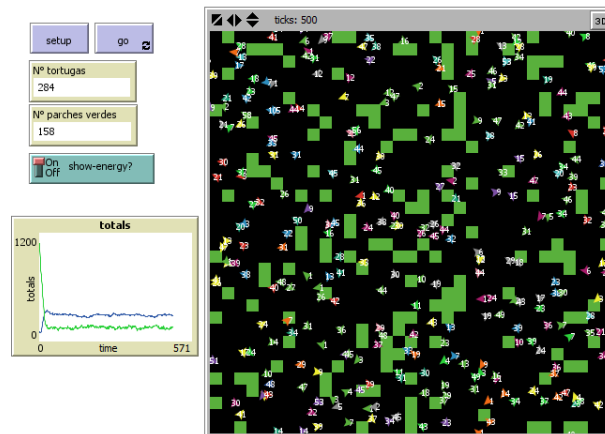
to setup-turtles
  create-turtles number

```

```
ask turtles [ setxy random-xxcor random-ycor ]  
end
```

- Y dentro de reproduce haga este cambio:
-

```
to reproduce  
  ask turtles [  
    if energy > birth-energy [  
      set energy energy - birth-energy  
      hatch 1 [ set energy birth-energy ]  
    ]  
  ]  
end
```



EL CODIGO COMPLETO DE LA SIMULACIÓN

```
turtles-own [energy]
```

```
to setup  
  clear-all  
  setup-patches  
  setup-turtles  
  do-plots  
  reset-ticks
```



```

end

to go
  if ticks >= 500 [ stop ]
  move-turtles
  eat-grass
  reproduce
  check-death
  regrow-grass
  tick
  do-plots
end

to move-turtles
  ask turtles [
    right random 360
    forward 1
    set energy energy - 1 ]
end

to setup-patches
  ask patches [ set pcolor green ]
end

to setup-turtles
  create-turtles number
  ask turtles [ setxy random-xcor random-ycor ]
end

to eat-grass
  ask turtles [
    if pcolor = green [
      set pcolor black
      set energy (energy + 10)
    ]
    ifelse show-energy?
    [ set label energy ]
    [ set label "" ]
  ]
end

```

```

to reproduce
  ask turtles [
    if energy > birth-energy [
      set energy energy - birth-energy
      hatch 1 [ set energy birth-energy ]
    ]
  ]
end

to check-death ;; verificar muerte
  ask turtles [
    if energy <= 0 [ die ]
  ]
end

to regrow-grass ;; rebrotar pasto
  ask patches [
    if random 100 < 3 [ set pcolor green ]
  ]
end

to do-plots
  set-current-plot "Totals"
  set-current-plot-pen "turtles"
  plot count turtles
  set-current-plot-pen "grass"
  plot count patches with [pcolor = green]
end

```
