

TALLER**SISTEMAS MULTIAGENTES**

Profesor: Ingrid Durley Torres & Jaime Alberto Guzmán Luna

Contenido del taller:

1. Introducción a NetLogo
2. Manejo Inicial
3. Ejemplo con estructuras de control y secuenciales
4. Actividad propuesta

NetLogo es un lenguaje de programación simple y adaptado al modelado/simulación de fenómenos en los que aparecen muchos individuos interactuando (como, por ejemplo, en los fenómenos habituales que se dan en la naturaleza, las sociedades, o muchas áreas de las ciencias).

El objetivo, A pesar de todas las ventajas, debe tenerse presente que el objetivo de NetLogo es proporcionar una herramienta de prototipado y desarrollo rápido de modelos de simulación, no de aplicaciones finales. Tiene como orientación principal la de modelar sistemas compuestos por individuos que interaccionan entre sí y con el medio, basándose en el paradigma de modelado por agentes: Un agente es un individuo sintético, autónomo y dotado de reglas o características que gobiernan su comportamiento y su capacidad de tomar decisiones. Los agentes interaccionan entre sí y con el medio ambiente obedeciendo un conjunto de reglas. Los agentes son flexibles y tienen capacidad de aprender y adaptar su comportamiento basándose en la experiencia. Esta capacidad requiere alguna forma de memoria. Los agentes incluso pueden tener reglas para modificar sus reglas de comportamiento.

INTRODUCCION

¿Cómo usarlo? Para descargar los ficheros de instalación de NetLogo, basta ir al sitio <http://ccl.northwestern.edu/netlogo/>.



Figura 1. Interfaz de la página de descargas NetLogo

Cuando ingresan encuentran dos versiones una instalable cuyo software funciona en varios sistemas operativos: Windows, MacOS, y Linux, siempre y cuando haya instalada una máquina virtual Java (JVM - Java Virtual Machine). Desde que NetLogo se actualizó a usar Java 8, en todas las versiones la descarga por defecto incluye la JVM necesaria y optimizada para NetLogo, y que no afectará a otras versiones instaladas en el sistema operativo.

Pero también existe una versión web, desde un navegador (está en desarrollo y todavía no presenta las mismas funcionalidades que la versión de escritorio, pero ya está muy avanzado).

Instalar NetLogo Lo primero que se debe saber es que la última versión instalable es la 6.1.1. y cuando se instala se descargan varias opciones (esto se habilita si en la instalación se generan las representaciones en el escritorio).

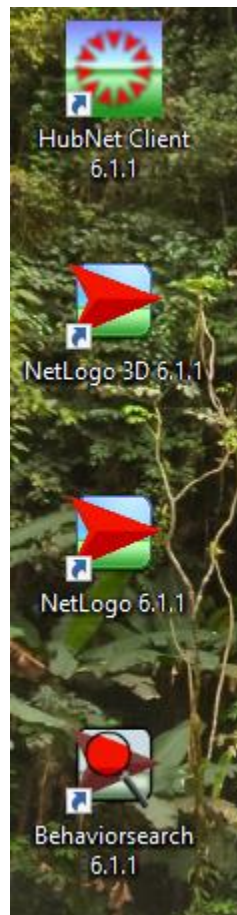


Figura 2. Interfaz de instalación vista escritorio

Para iniciar todo se usará el NetLogo 6.1.1, que corresponde a una versión 2D. Es decir, daremos click, sobre la opción “NetLogo 6.1.1”, y con ella se desplegará la siguiente ventana.

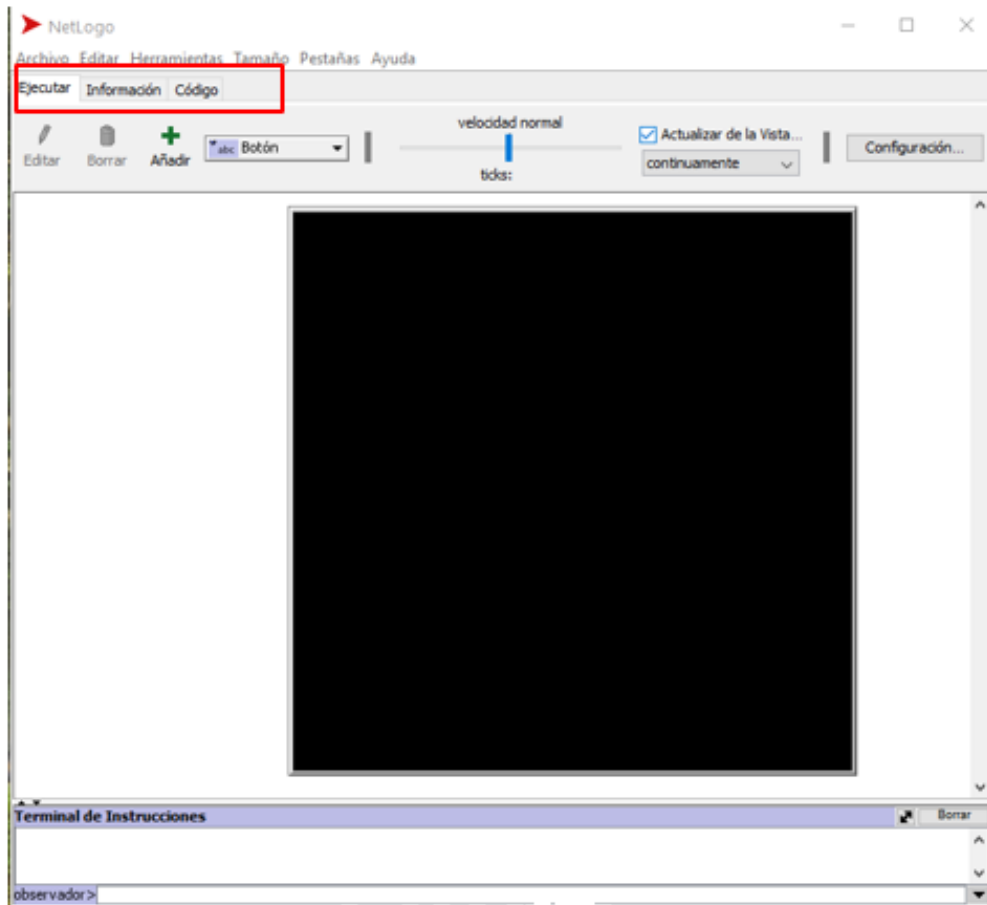


Figura 3. Interfaz inicial NetLogo

ACTIVIDAD 1. PROBANDO EL CENTRO DE COMANDOS

Es importante recordar los tipos de agentes con los que cuenta esta plataforma y los cuales se identificarán desde la misma interfaz de comandos:

- Agentes móviles (tortugas): son los agentes que se mueven por el mundo. El mundo es (en principio) 2D y está dividido en una malla de patches.
- Agentes inmóviles (patches): cada una de las divisiones cuadradas del mundo.
- Agentes conectores (links): agentes que conectan entre sí los agentes móviles (a modo de aristas de un grafo).
- Agente observador: no tiene localización, y puede interactuar con todos los elementos del mundo. De alguna forma, representa al superagente que puede controlar todas las demás componentes del mundo.

Una vez identificados es posible habilitar la línea de comandos para que revisemos algunos comportamientos propios de los agentes, desde ejecución de instrucciones inmediatas (on fly).

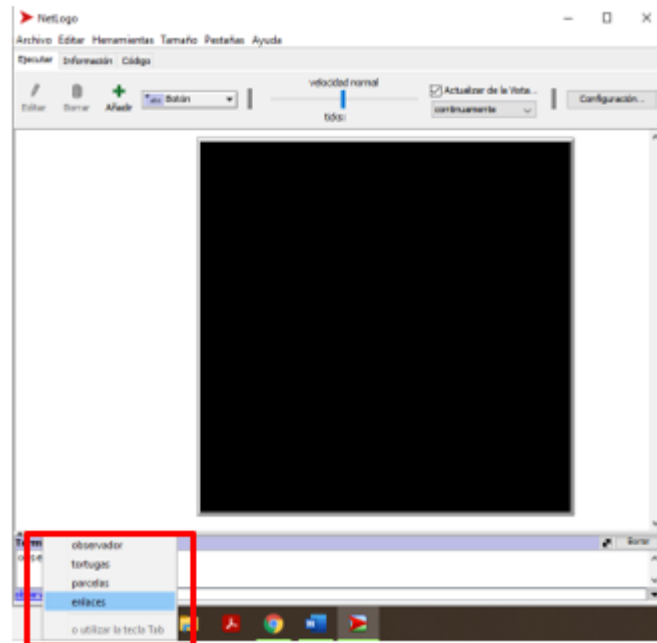


Figura 4. Habilitando la línea de comandos

Ahora se seleccionará el agente sobre el cual debemos ejecutar la instrucción.

- a. Para crear tortugas, usamos la primitiva `create-turtles` (ó `crt` en abreviatura), seguido del número de tortugas que queremos crear. Ejemplo:

`crt 50 (observador)`

- b. Como no se observan bien las tortugas creadas, se va a ordenar que las tortugas, recién creadas, se muevan 10 pasos adelante.

`ask turtles [fd 10] (observador)`

- c. Ahora cree nuevamente 50 tortugas, pero esta vez la orden de que se mueva se va a dar directamente sobre las tortugas. *Ask*, es para dar ordenes desde el observador a un agente específico.

`[fd 10] (tortugas)`

- d. Ahora probemos: ¿qué pasa, que indica?

`if (random 3) = 1 [die] (tortugas)`

- e. Ahora se realizarán algunas tareas sobre el agente *parcelas*.

`if pxcor > 5 [set pcolor blue](parcelas)`

- f. Finalmente, se ejecuta

`ca (observador)`

ACTIVIDAD 2. REALIZANDO PROCEDIMIENTOS

Una **instrucción** es una acción que un agente debe llevar a cabo, lo que resulta en algún efecto. Un **reportero provee instrucciones para calcular un valor**, que luego el agente "informa" a quien lo solicitó.

Normalmente, un nombre de instrucción comienza con un verbo, como "**crear**", "**morir**", "**saltar**", "**inspeccionar**" o "**borrar**". La mayoría de los nombres de los reporteros son sustantivos o frases nominales.

Las instrucciones y reporteros integrados en NetLogo se llaman primitivas. El diccionario de NetLogo tiene una lista completa de instrucciones y reporteros integrados. Las instrucciones y reporteros que usted define se llaman procedimientos. Cada procedimiento tiene un nombre, precedido por la palabra clave **to** o **to-report**, dependiendo de si se trata de un **procedimiento de instrucciones** o un **procedimiento de reportero**. La palabra clave **end** es el final de las instrucciones en el procedimiento. Una vez que define un procedimiento, puede usarlo en cualquier otro lugar de su programa.

ACTIVIDAD 2.1 Creando Botones

Lo primero, por ahora es crear un primer programa. Para ello se selecciona archivo nuevo y la ruta, como se hace con cualquier archivo tradicional.

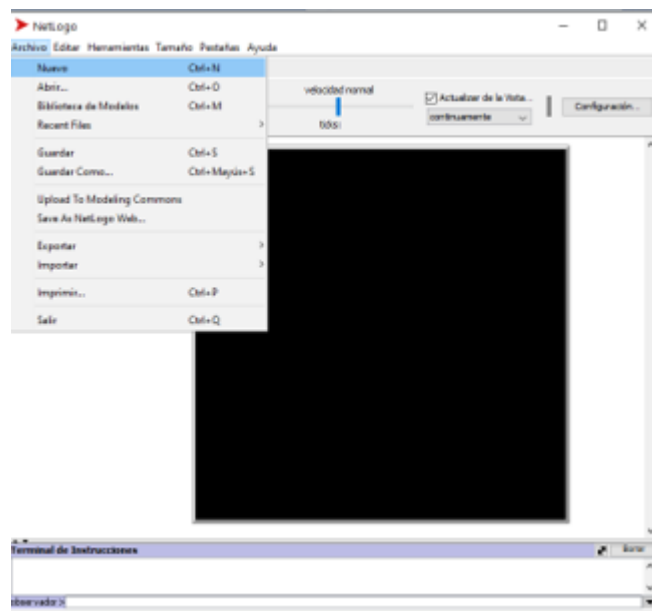


Figura 5. Interfaz nuevo archivo

Posteriormente, desde la ventana ejecutar (ver figura 3), se deben adicionar los botones de control de configuración básicos:

- **SETUP**: Botón que ejecuta una acción
- **GO**: Botón que repite la misma acción hasta parar “continuamente”

Pasos para la adición del botón setup:

- Haga clic en el ícono "Añadir" en la barra de herramientas en la parte superior de la pestaña Interfaz.

- En el menú al lado de Añadir, seleccione Botón (si aún no está seleccionado).
- Haga clic donde desee que aparezca el botón en el área blanca vacía de la pestaña Interfaz.
- Se abre un cuadro de diálogo para editar el botón. Escriba setup en el cuadro etiquetado "Instrucciones".
- Presione el botón OK cuando haya terminado; el cuadro de diálogo se cierra.

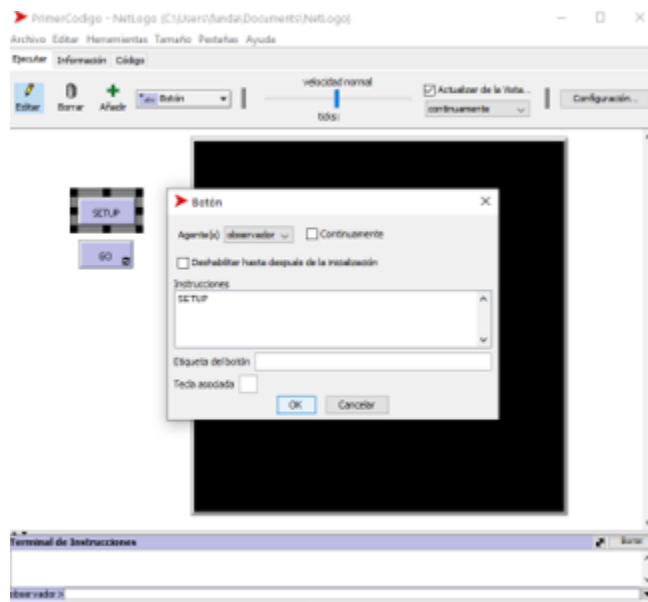


Figura 6. Interfaz añadir el botón SETUP

De igual forma se adiciona el botón “GO” pero acá existen dos cambios, para hacerlo: i) En instrucción se ubica GO, ya no SETUP. Y ii) se habilita la casilla de continuamente. Esto para que una vez se haga “click” sobre él, inicie el proceso de activación de las instrucciones de ese botón y solo se detienen, hasta que nuevamente se de “click” sobre él.

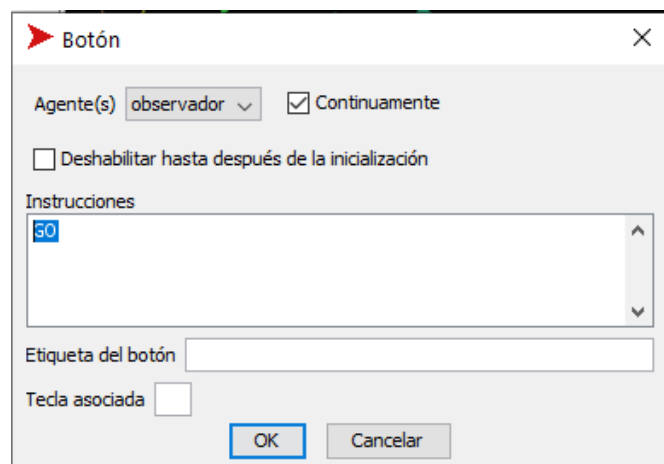


Figura 7. Interfaz botón GO

ACTIVIDAD 2.2 Creando Código a los Botones

Una vez realizado esto y si no existen instrucciones de código puede que las letras de ambos botones salgan en color rojo. Ahora, se configurarán ambos botones, desde la pestaña “código” (Ver figura 3). Para ello sobre escribimos el siguiente código (los punto y coma, se usan para indicar comentarios y con ellos, se ha definido lo que significa para NetLogo, cada instrucción).

Ahora en la pestaña de código **CODIGO 1**

```
to setup ;;para configurar
  clear-all ;;limpiar todo
  create-turtles 10 ;;crea 10 tortugas
  reset-ticks ;;redefinir pasos (de tiempo)
end ;; fin del procedimiento

to go ;; para ejecutar (correr el procedimiento)
  ask turtles [ ;; pedir a las tortugas
    fd 1 ;; avanzar 1 paso
    rt random 10 ;; girar a la derecha aleatoriamente entre 0 y 9
    lt random 10 ;; girar a la izquierda aleatoriamente entre 0 y 9
  ]
  tick
end
```

ACTIVIDAD 2.3 Asignando comportamiento de aspecto al agente celdas

Ahora para mejorar la interfaz se adicionará una orden para que las celdas, sean de color verde, para ello, insertaremos la siguiente instrucción en el botón *setup*:

ask patches[set pcolor green]

Esta instrucción puede ubicarse despues de crear las tortugas o incluso antes, pero nunca fuera del botón *end* del *setup*.

Para validar su correcto funcionamiento, se puede regresar a la pestaña “Ejecutar” y nuevamente se activan los botones de control *setup* y *go*. Lo que ahora se puede observar, es una imagen similar a la señalada en la Figura 8.

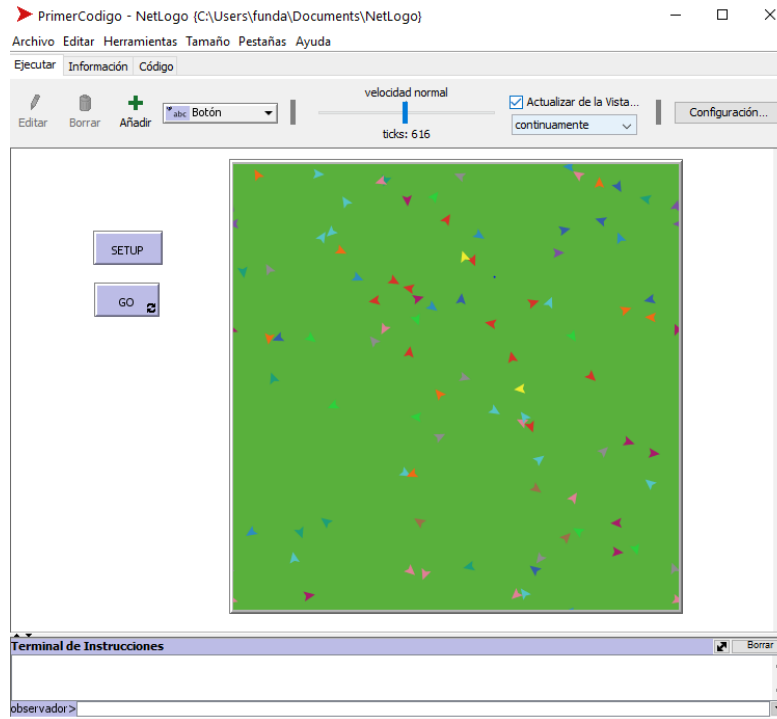


Figura 8. Interfaz PrimerCódigo

ACTIVIDAD 2.4 asignando comportamiento movimiento al agente tortuga

Ahora se otorgarán comportamientos a las agentes tortugas. Para ello, por ejemplo, se hará la simulación más iterativa, creamos un botón que se llame *mover tortuga* e incluimos el siguiente código:

CODIGO 2

```
to movertortuga
```

```
  ask turtles [;; decimos a cada elemento de turtles que realice las siguientes acciones dentro
    forward 1 ;; desplazar una parcela en donde está dirigida la tortuga.
```

```
    set heading random 360 ;; se le dice a turtle que gire a una dirección aleatoria de 360
```

```
    if (pcolor = green)[;; Y en la parcela que se desplaza preguntamos si es de color verde,
      set pcolor blue ;; si es verde que la establezca azul.
```

```
  ]
```

```
]
```

```
end
```

RECUERDE GUARDAR su código. Nuevamente, volvemos a la pestaña “Ejecutor” y observamos....No funciona...Por?

Debemos cambiar el código del botón GO por:

```
to go ;; para ejecutar (correr el procedimiento)
```

```
  movertortuga
```

```
  tick
```

```
end
```


Ahora nuevamente se va a la pestaña ejecutar y si los cambios son muy rápidos se sugiere, modificar la velocidad de la simulación.

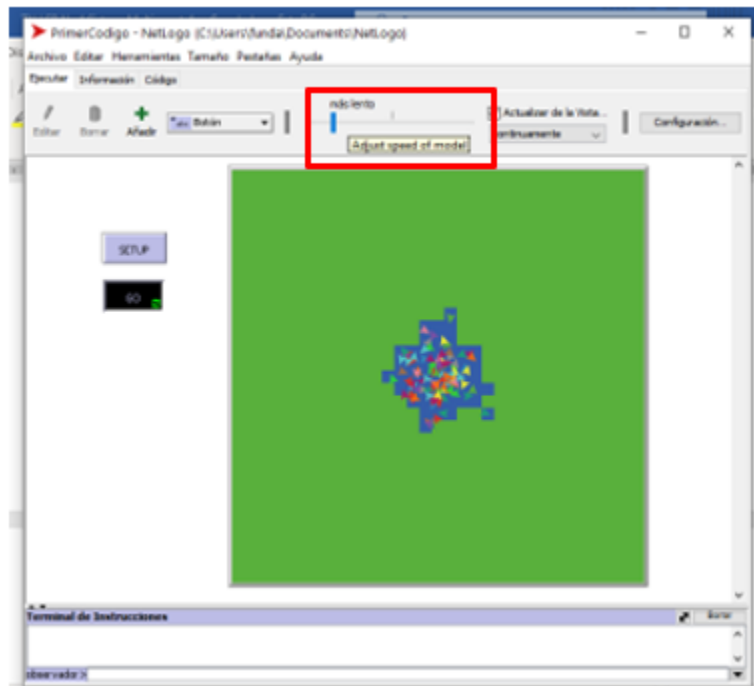


Figura 9. Ajuste velocidad de simulación a más lento

ACTIVIDAD 2.5 asignando comportamiento muerte al agente tortuga

Ahora se va a otorgar otro comportamiento, que las **tortugas mueran**, para ello se tiene una resistencia de 10 y por cada movimiento que realice cada tortuga, se disminuye un punto esa resistencia, pero si come en un pasto verde, la resistencia no baja (se mantiene), luego incluiremos ese comportamiento en el siguiente código:

CODIGO 3

- como primera línea de código, se incluirá:
turtles-own[resistencia];; Establecemos un atributo a las tortugas llamado resistencia.
- En el *setup* inicializamos el valor de resistencia a 10. Esto se hace inmediatamente después de crear las tortugas *set resistencia 10*:

```

to setup ;;para configurar
  clear-all ;;limpiar todo
  ask patches[set pcolor green ]
  create-turtles 100 ;;crea 100 tortugas
  reset-ticks ;;redefinir pasos (de tiempo)
end ;; fin del procedimiento

```
- Ahora se reescribirá el código de *mover tortuga* a:

```

to mover tortuga
  ask turtles [

```

set resistencia 10;;Establecemos a la tortuga una resistencia de 10

```
forward 1  
set resistencia resistencia - 1  
set heading random 360  
if (pcolor = green)[  
  set pcolor blue  
  set resistencia resistencia + 1  
]  
if(resistencia = 0)[  
  die  
]  
]  
end
```

CODIGO 5. Código completo con los 3 procedimientos

```
turtles-own[resistencia]

to setup ;;para configurar
    clear-all ;;limpiar todo
    ask patches[set pcolor green ]
    reset-ticks ;;redefinir pasos (de tiempo)
    create-turtles 100 ;;crea 100 tortugas
end ;; fin del procedimiento

to go ;; para ejecutar correr el procedimiento)
    asignarresistencia
    movertortuga
    tick
end

to movertortuga
    ask turtles [
    set resistencia 10
    forward 1
    set resistencia resistencia - 1
    set heading random 360
    if (pcolor = green)[
        set pcolor blue
        set resistencia resistencia + 1
    ]
    if(resistencia = 0)[
        die
    ]
    ]
end
```

ACTIVIDAD 2.6 Monitores

A continuación, creará dos monitores en la pestaña Ejecutar, con la barra de herramientas. (Los hará como botones y controles deslizantes, usando el ícono Añadir en la barra de herramientas.)

- Cree un monitor haciendo clic en el ícono Añadir en la barra de herramientas, seleccionando Monitor al lado y haciendo clic en un lugar abierto en la interfaz.
- Aparecerá un cuadro de diálogo: En el tipo de diálogo: `count turtles` (ver Figura 10).
- Presione el botón OK para cerrar el diálogo.

turtles es un "conjunto de agentes", el conjunto de todas las tortugas. *count* nos dice cuántos agentes hay en ese conjunto.

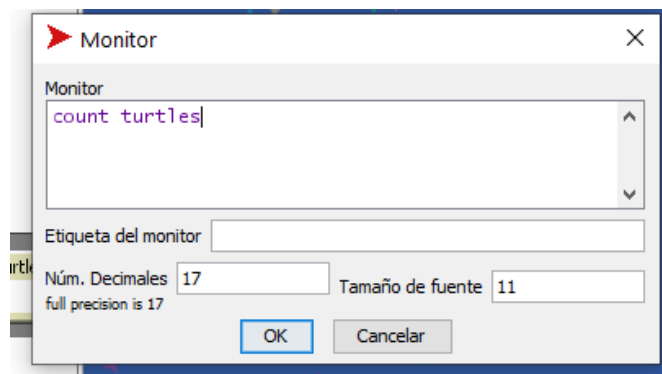


Figura 10. Monitor contar tortugas

Hagamos el segundo monitor ahora:

- Cree un monitor haciendo clic en el ícono Añadir en la barra de herramientas, seleccionando Monitor al lado y haciendo clic en un lugar abierto en la Interfaz.
- Un cuadro de diálogo aparecerá. En la sección Monitor del cuadro de diálogo, escriba:
count patches with [pcolor = green] (ver Figura 11).
- En la sección Etiqueta del monitor del cuadro de diálogo, escriba: *green patches*
- Presione el botón OK para cerrar el cuadro de diálogo.

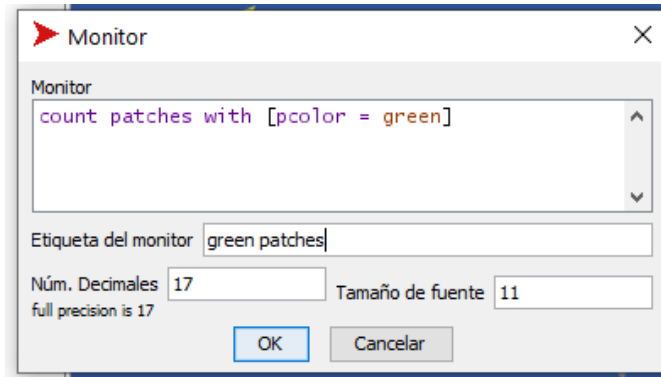


Figura 11. Monitor contar patches verdes


Aquí estamos usando conteo nuevamente para ver cuántos agentes hay en un conjunto de agentes. *patches* es el conjunto de todas las parcelas, pero no solo queremos saber cuántas parcelas hay en total, queremos saber cuántas de ellas son verdes. Eso es lo que *with* hace; hace un conjunto de agentes más pequeños de aquellos agentes para quienes la condición entre corchetes es verdadera. La condición es *pcolor = green*, por lo que nos da solo las parcelas verdes.

Ahora tenemos dos monitores que reportarán cuántas tortugas y parcelas verdes tenemos, para ayudarnos a rastrear lo que sucede en nuestro modelo. A medida que el modelo se ejecuta, los números en los monitores cambiarán automáticamente.

ACTIVIDAD 2.7 Graficando

crear un gráfico en la pestaña Ejecutar y colocar algunas instrucciones dentro de él. Las instrucciones que ponemos en las gráficas se ejecutarán automáticamente cuando nuestro procedimiento de *setup* llame a *reset-ticks* y cuando nuestro procedimiento *go* funcione.

- Cree un gráfico haciendo clic en el ícono Añadir en la barra de herramientas, seleccionando Gráfico al lado y haciendo clic en un lugar abierto en la Interfaz.
- Establezca su nombre en "Totales" (ver Figura 12)
- Establezca la etiqueta del eje X en "time"
- Establezca la etiqueta del eje Y en "totales"
- Cambie el nombre del trazo por defecto "default" por "turtles".
- Ingrese el conteo de las tortugas debajo de los "Instrucciones de Actualización de Trazos": *plot count turtles*
- Presione el botón "Añadir Trazo".
- Cambie el nombre del nuevo pen a "grass".
- Ingrese el conteo de las patches debajo de los "Instrucciones de Actualización de Trazos": *plot count patches with [pcolor = green]*
- Cuando termine, se dará OK

 Gráfico ✕

Nombre

Etiqueta del eje X X mín X máx







Etiqueta del eje Y Y mín Y máx

☒ ¿Escala automática? ☐ ¿Mostrar leyenda?

☐ Instrucciones de Inicialización del Gráfico

☐ Instrucciones de Actualización del Gráfico

Trazos del Gráfico

Color	Nombre del Trazo	Instrucciones de Actualización de Trazos		
	turtles	<code>plot count turtles</code>		
	grass	<code>plot count patches with [pcolor = green]</code>	