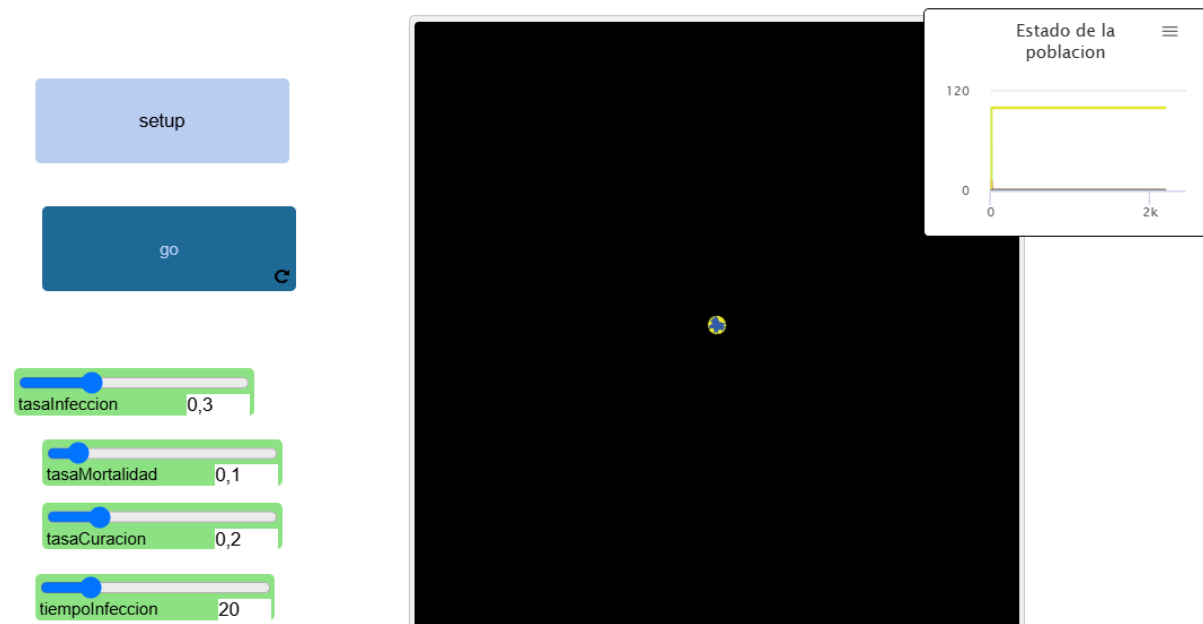


Modelos Netlogo

Programación III
Luisa Maria Reyes

1. Modelo de Propagación de Epidemias

- Propósito: Observar la propagación de un virus en una población.
- Agentes: Población, que se puede dividir en población vulnerable como adultos mayores o niños, fuentes de contagio, como agua, alimentos, etc. agentes como médicos, que se encargaran de encontrar una cura.
- Entorno: Comunidad o región, donde solo hay una fuente de agua disponible y escasos recursos.
- Reglas: La comunidad está en cuarentena, hay curas limitadas y casi no hay servicios de atención.
- Parámetros ajustables: velocidad de propagación, índice de mortalidad, eficiencia de una vacuna o una cura.
- Resultados esperados: Observar cuántos agentes de la población se contagian en un intervalo de tiempo, que tan eficiente es una cura, y que secuelas puede dejar en los agentes ya curados.



globals [] ;; Eliminamos tasaInfeccion, tasaMortalidad, tasaCuracion y tiempoInfeccion de la lista globals

```
turtles-own [  
  estado      ;; Estado de la tortuga: sano, infectado, curado, fallecido, medico  
  vulnerabilidad ;; Vulnerabilidad de la tortuga: alta o baja  
  tiempoInfectado ;; Tiempo que una tortuga ha estado infectada  
]
```

```
to setup  
  clear-all
```

```
; Crear las tortugas (personas) con diferentes estados iniciales
```

```
create-turtles 100 [  
  ifelse random 2 = 0 [  
    set vulnerabilidad "alta"  
  ] [  
    set vulnerabilidad "baja"  
  ]  
  set estado "sano"  
  set tiempoInfectado 0  
  set color green  
]
```

```
; Infectar a una tortuga inicial para comenzar la propagación
```

```
ask one-of turtles [  
  set estado "infectado"  
  set color red  
  set tiempoInfectado 0  
]
```

```
; Crear médicos
```

```
create-turtles 5 [  
  set estado "medico"  
  set color blue  
]
```

```
reset-ticks  
end
```

```
to go  
  infectar  
  evolucionarInfeccion  
  curar  
  aplicarCuras  
  actualizarDatos  
  tick  
end
```

```

to infectar
  ask turtles with [estado = "infectado"] [
    let vecinos turtles in-radius 1
    ask vecinos with [estado = "sano"] [
      if random-float 1 < tasaInfeccion [
        set estado "infectado"
        set tiempoInfectado 0
        set color red
      ]
    ]
  ]
end

```

```

to evolucionarInfeccion
  ask turtles with [estado = "infectado"] [
    set tiempoInfectado tiempoInfectado + 1
    if tiempoInfectado > tiempoInfeccion [
      if random-float 1 < tasaMortalidad [
        set estado "fallecido"
        set color black
      ]
    ]
  ]
end

```

```

to curar
  ask turtles with [estado = "infectado"] [
    if random-float 1 < tasaCuracion [
      set estado "curado"
      set color yellow
      ifelse random 2 = 0 [
        set vulnerabilidad "alta"
      ] [
        set vulnerabilidad "baja"
      ]
    ]
  ]
end

```

```

to aplicarCuras
  ask turtles with [estado = "medico"] [
    let infectadosCercanos turtles in-radius 2 with [estado = "infectado"]
    ask infectadosCercanos [
      if random-float 1 < tasaCuracion [
        set estado "curado"
        set color yellow
      ]
    ]
  ]
end

```

```
]
]
end
```

to actualizarDatos

```
set-current-plot "Estado de la poblacion"
```

```
set-current-plot-pen "Sanos"
```

```
plot count turtles with [estado = "sano"]
```

```
set-current-plot-pen "Infectados"
```

```
plot count turtles with [estado = "infectado"]
```

```
set-current-plot-pen "Curados"
```

```
plot count turtles with [estado = "curado"]
```

```
set-current-plot-pen "Fallecidos"
```

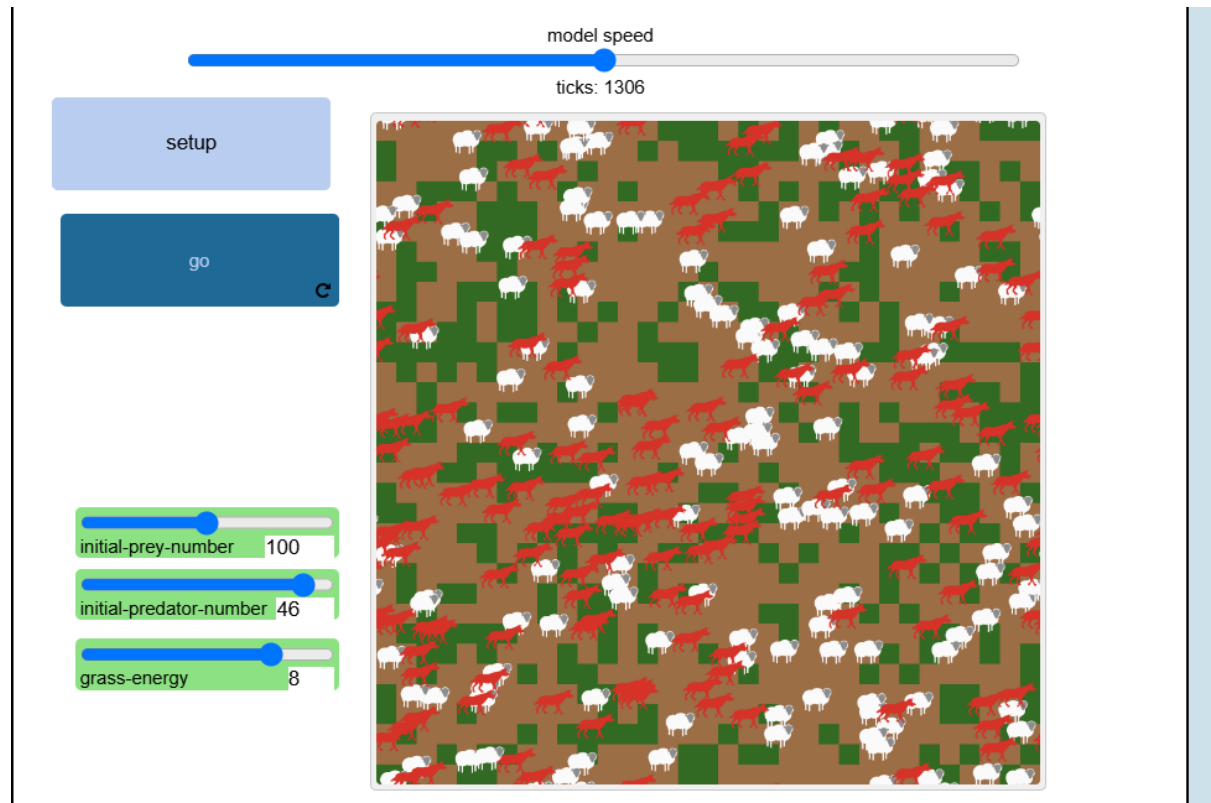
```
plot count turtles with [estado = "fallecido"]
```

end

2. Modelo de Ecosistema de Depredador-Presa

- Propósito: Entender la interacción entre depredador y presa y cómo esto afecta el ecosistema
- Agentes: Presas y depredadores
- Entorno: Ecosistema de diferentes hábitats, recursos limitados etc, que afectan la relación y supervivencia de las presas y depredadores.
- Reglas: Tienen que seguir la cadena alimenticia, y el número de depredadores depende de las presas.
- Parámetros ajustables: Numero de presas y depredadores, características de ecosistema,

- Resultados esperados: Representación de las poblaciones de depredadores y presas, y entender factores que influyen en esta población y el ecosistema.



globals [] ;; Ninguna variable global manual aquí

breed [preys prey]

breed [predators predator]

preys-own [energy]

predators-own [energy]

;; Configuración inicial del modelo

to setup

clear-all

setup-patches

setup-preys

setup-predators

reset-ticks

end

;; Configuración de las áreas con recursos

to setup-patches

ask patches [

set pcolor green - 2 ;; Color para representar vegetación o hábitats

```
]
end
```

```
:: Inicializar las presas
to setup-preys
  create-preys initial-prey-number [
    set shape "sheep"
    set color white
    set size 1.5
    setxy random-xcor random-ycor
    set energy random 10 + 10 ;; Energía inicial aleatoria
  ]
end
```

```
:: Inicializar los depredadores
to setup-predators
  create-predators initial-predator-number [
    set shape "wolf"
    set color red
    set size 2
    setxy random-xcor random-ycor
    set energy random 20 + 20 ;; Energía inicial aleatoria
  ]
end
```

```
:: Acción que ejecuta el modelo en cada "tick"
to go
  if not any? preys [ stop ] ;; El modelo termina si no quedan presas
  ask preys [
    prey-move
    eat-grass
    reproduce-preys
    lose-energy
    check-death
  ]
  ask predators [
    predator-move
    eat-prey
    reproduce-predators
    lose-energy
    check-death
  ]
  regrow-grass ;; La vegetación vuelve a crecer
  tick
end
```

```
:: Movimiento de las presas
to prey-move
```

```
    rt random 50 - random 50
    fd 1
end
```

```
:: Las presas comen hierba para obtener energía
to eat-grass
  if pcolor = green - 2 [
    set pcolor brown
    set energy energy + grass-energy
  ]
end
```

```
:: Reproducción de las presas
to reproduce-preys
  if energy > 20 [
    set energy energy / 2
    hatch-preys 1 [
      rt random-float 360
      fd 1
    ]
  ]
end
```

```
:: Movimiento de los depredadores
to predator-move
  rt random 50 - random 50
  fd 1
end
```

```
:: Los depredadores cazan presas para obtener energía
to eat-prey
  let target one-of preys-here
  if target != nobody [
    ask target [
      die
    ]
    set energy energy + 10
  ]
end
```

```
:: Reproducción de los depredadores
to reproduce-predators
  if energy > 40 [
    set energy energy / 2
    hatch-predators 1 [
      rt random-float 360
      fd 1
    ]
  ]
end
```

```
]
end
```

```
:: Las presas y depredadores pierden energía con el tiempo
to lose-energy
  set energy energy - 1
end
```

```
:: Verifica si una presa o depredador debe morir por falta de energía
to check-death
  if energy <= 0 [
    die
  ]
end
```

```
:: Regeneración de la vegetación en las áreas donde no hay presas
to regrow-grass
  ask patches with [pcolor = brown] [
    if random-float 100 < 5 [
      set pcolor green - 2
    ]
  ]
end
```