

Sistemas Multiagente

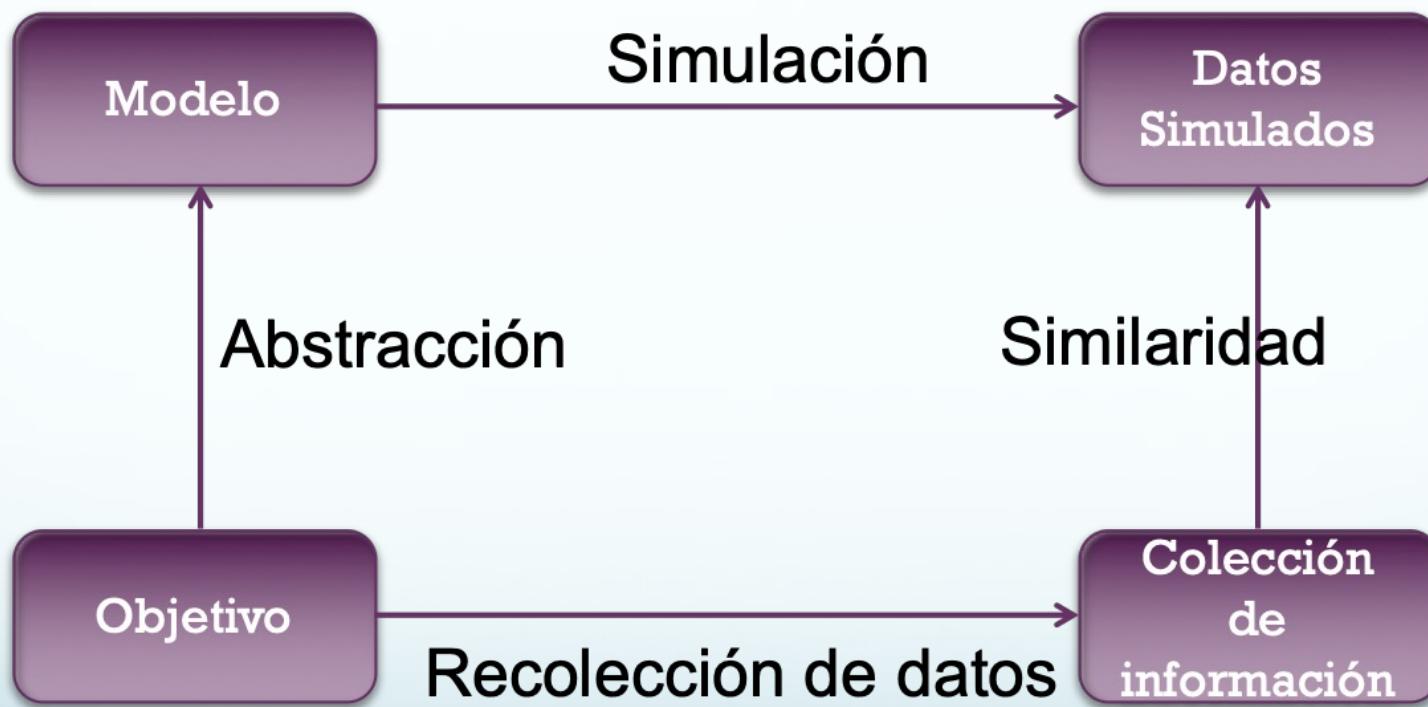
Tema 4
Simulación basada en agentes

Authors: Vicent Botti, Vicente Julián

¿Por qué simular?

- La parte del mundo real que se quiere examinar no es accesible, por ejemplo **el sistema no existe todavía**.
- Experimentar con **el sistema real es inviable** debido a diferentes circunstancias
- Es necesario **escalar en el tiempo o el tamaño** del sistema es demasiado grande o pequeño para ser observado
- El sistema real y su influencia por parte del entorno no son controlables
- **Útil como herramienta para entender y predecir**
 - Responder a las preguntas sobre “que pasaría si?”.

Simulación como un Método



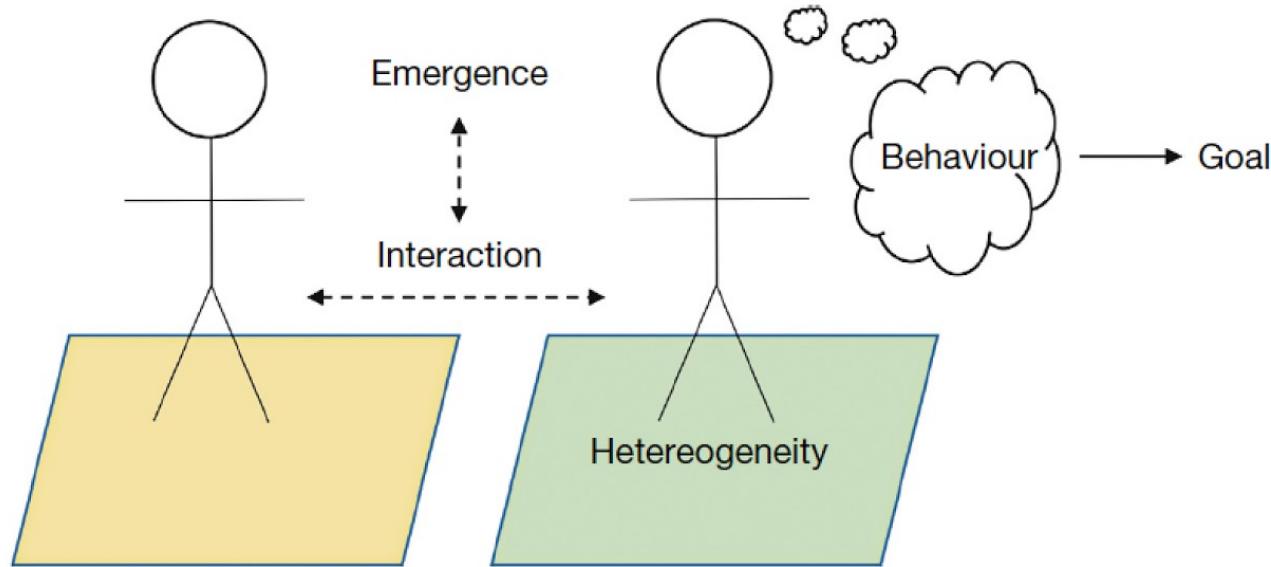
Simulación. Evolución

- **Dinámica de sistemas y modelos mundiales:** Extensos sistemas de ecuaciones para seguir la trayectoria de variables a través del tiempo:
 - Ejemplo: ecuaciones sobre la evolución de la economía global. Club de Roma (Meadows, 1974).
- **Simulación de colas:** modeliza el paso de unidades a través de procesos para predecir el resultado típico.
 - Ejemplo: tiempo de espera de compradores en una cola, tiempo que coches de policía tardan en llegar a una emergencia. <http://www.arenasimulation.com/>
- **Autómatas celulares:** las propiedades de un sistema pueden ser modeladas simulando las interacciones entre las unidades componentes (moléculas, individuos) situadas en una extensa cuadricula rectangular, en la que cada celda se encuentra en una situación (número reducido) y los cambios suceden según unas reglas que dependen de las situaciones de los inmediatos vecinos de la celda considerada.
 - Ejemplo: juego de la vida (<http://www.math.com/students/wonders/life/life.html>)
- **Sistemas multiagente:** se diseña, se pone en funcionamiento y analiza los resultados formados por un entorno de base, sobre el cual se distribuyen y se desplazan una multitud de agentes de diversos tipos.

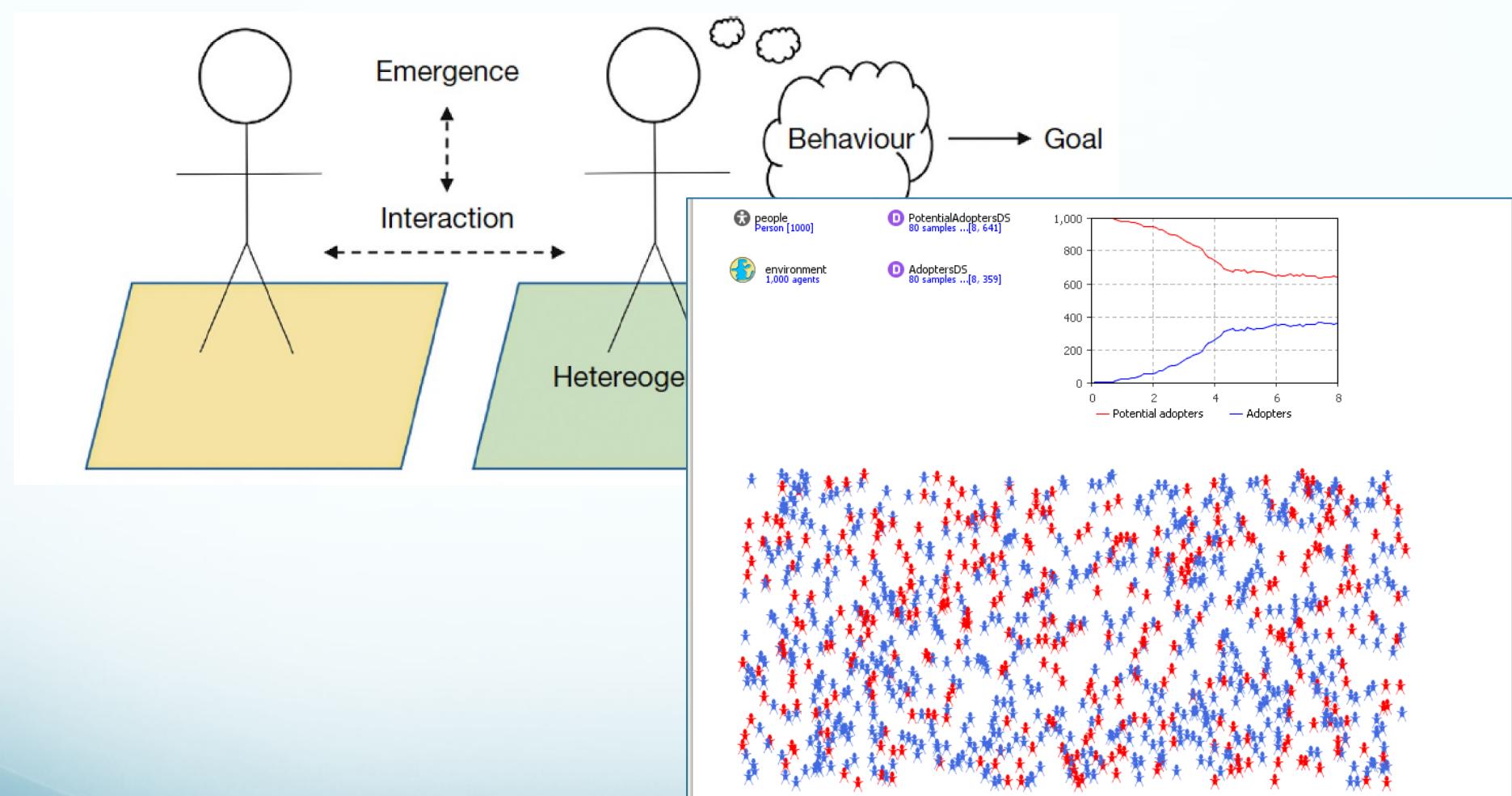
Simulación basada en agentes: ¿Qué es?

- Emerge como un soporte ante la alternativa de la simulación de sistemas complejos (Drogoul et al., 2003)
- La Simulación basada en agentes intenta facilitar dos tareas (Bazzan,2005):
 - comprender el sistema y
 - hacer predicciones cualitativas y cuantitativas en relación al futuro.
- **Un modelo de simulación basado en agentes consiste de un conjunto de agentes los cuales encapsulan el comportamiento de varios individuos.**
- La ejecución del modelo emula el comportamiento global del sistema.

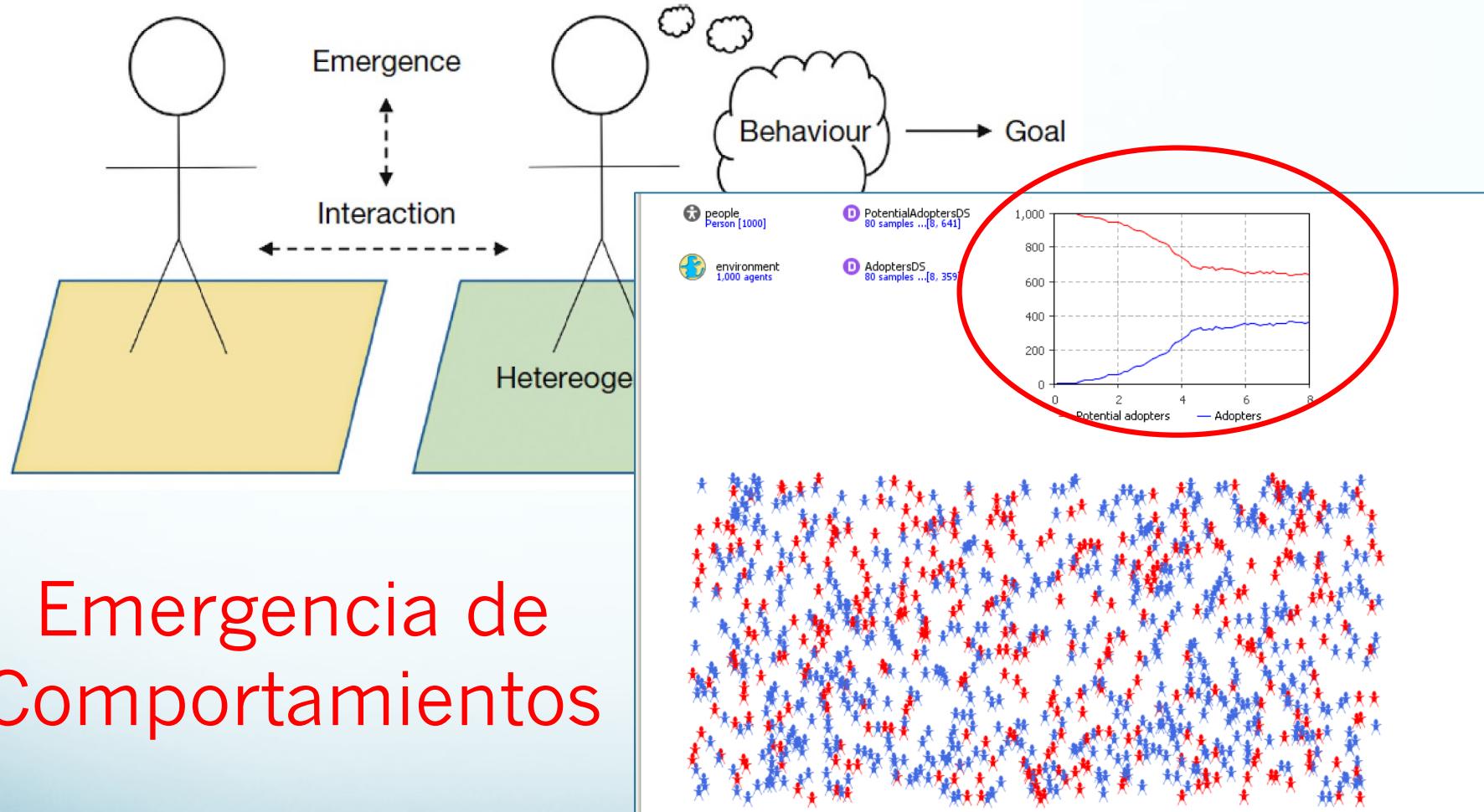
Simulación basada en agentes: ¿Qué es?



Simulación basada en agentes: ¿Qué es?

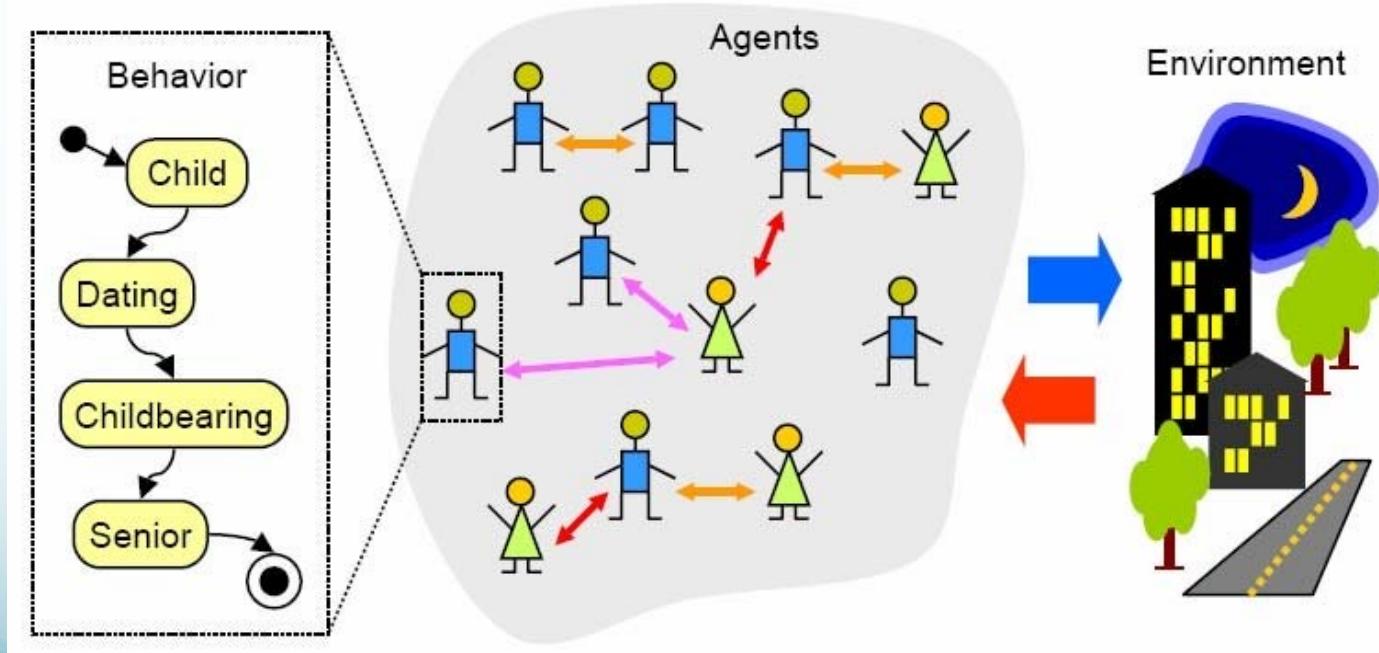


Simulación basada en agentes: ¿Qué es?



Simulación basada en agentes: ¿Qué es?

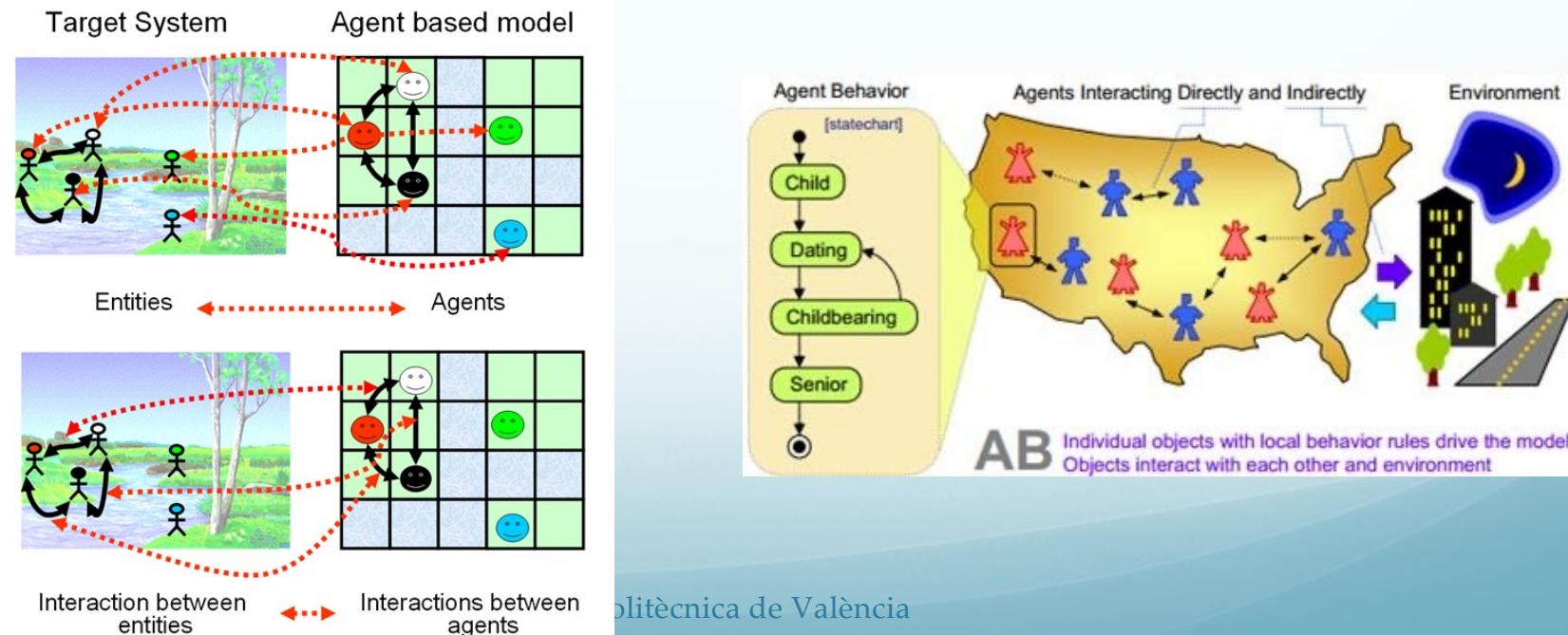
- Un sistema se modela como una colección de entidades autónomas que toman decisiones, llamadas agentes.
- Cada agente evalúa individualmente su situación y toma decisiones sobre la base de un conjunto de reglas.
- Los agentes pueden ejecutar varios comportamientos apropiados para el sistema que representan, por ejemplo, producir, consumir o vender.



Simulación basada en agentes: ¿Qué es?

Micromotivos vs Macroconducta

- El comportamiento colectivo emerge de la interacción entre los agentes
- Existe una relación de interdependencia y causalidad: las decisiones tomadas por unos agentes tienen efectos sobre las decisiones de otros.
- Permiten observar los fenómenos emergentes de conducta colectiva a partir de la interacción entre los agentes sociales.



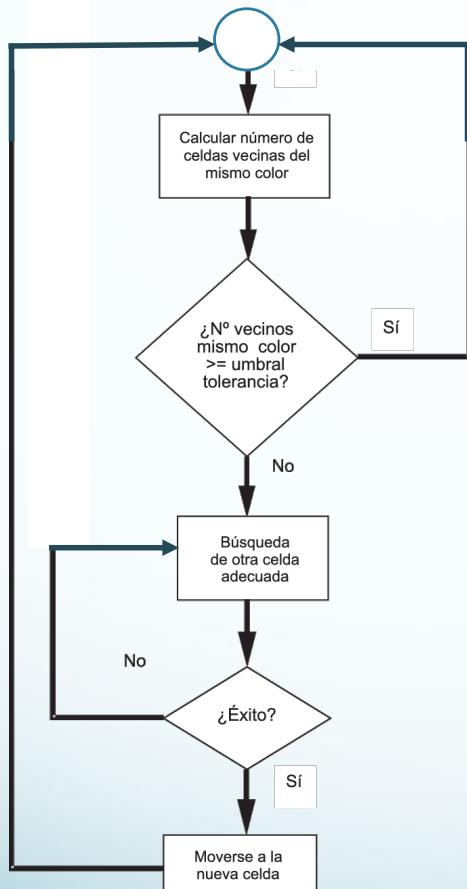
Modelo de Segregación

- Ejemplo: Modelo de Segregación <http://ncase.me/polygons-es/>
 - Rejilla de 500x550 celdas. 1.500 agentes : 1050 verdes, 450 rojos, 1000 celdas vacantes
 - Hipótesis: *si un individuo tolera la diversidad racional pero no tolera estar en minoría en su ambiente local, se produce y estabiliza la segregación*
- Reglas simples de los agentes
 - Un agente es “feliz” y no cambia de celda, si tiene una determinada cantidad de agentes del mismo color en su vecindario local
 - Si la cantidad de agentes es menor a la deseada, el agente “infeliz” se cambia de celda hacia una donde sea “feliz”
 - Si un agente “infeliz” se cambia de celda, puede “perturbar” a un agente “feliz” que pasa a ser “infeliz” y se cambiará de celda

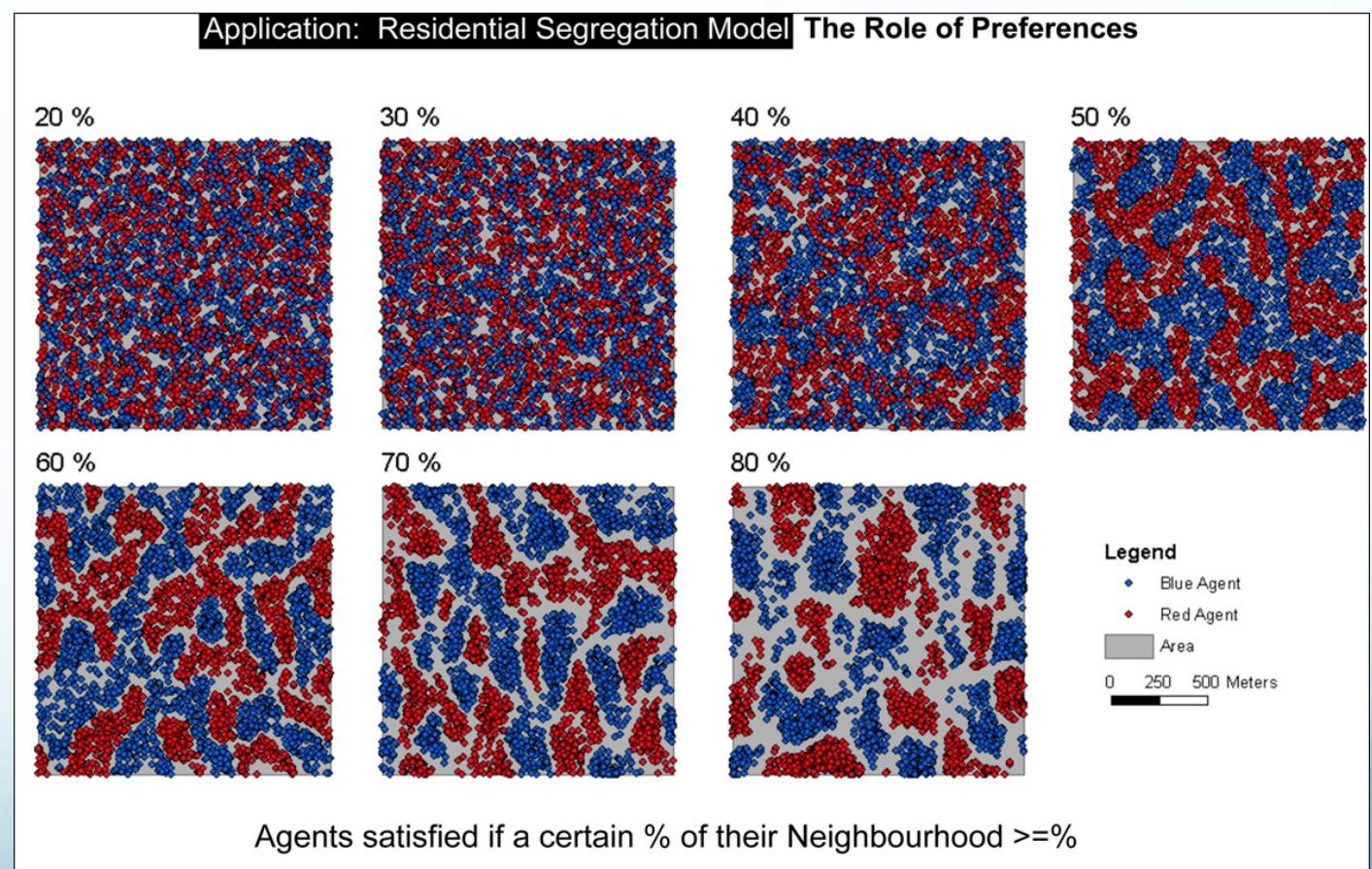
Thomas Schelling. Dynamics Models of Segregation (1971). Journal of Mathematical Sociology, 1: 143-186.

Modelo de Segregación

Cada agente ejecuta
(nivel micro)



Resultados a nivel macro



Modelo de Segregación

Reflexiones del modelo:

- A partir del comportamiento de cada agente, desde reglas simples, se genera, a nivel emergente, un estado final del sistema
 - Creación de guetos y la segregación racial
- Con valores de tolerancia de un 30% se producen guetos
- **Ninguno de los individuos tiene un conocimiento global ni puede anticipar las consecuencias de sus decisiones agregadas**
- <http://www.netlogoweb.org/launch#http://www.netlogoweb.org/assets/modelslib/Sample%20Models/Social%20Science/Segregation.nlogo>

Ejemplo: Sugarscape

- Ejemplo muy conocido por ilustrar “ciencia social generativa”
- Define fenómenos sociales y los patrones basados en el comportamiento individual y las interacciones
- El objetivo de esta perspectiva consiste en hacer emerger a partir de la interacción de los agentes en un ambiente determinado las propiedades de un sistema social.
- Búsqueda de los procesos mínimos necesarios: agentes simples generan nuevas estructuras sociales: Bottom-Up
- Plataforma de implementación computacional de un modelo basado en agentes orientado hacia las sociedades artificiales:
 - *objetivo principal:* poner a prueba diferentes escenarios relacionados con fenómenos sociales
- <http://en.wikipedia.org/wiki/Sugarscape>
- <http://ccl.northwestern.edu/netlogo/models/community/Sugarscape>

Ejemplo: Sugarscape

- Elementos: entorno, agentes y reglas determinadas
- Entorno bidimensional en el cual se mueven los agentes
- Normalmente es una rejilla de 50x50, donde cada espacio puede ser ocupado por un agente a la vez.
- Dos componentes: azúcar y especias
- Los agentes nacen con un metabolismo determinado:
 - con un nivel de consumo de azúcar por unidad de tiempo.
- Existen dos regiones con alta disponibilidad de azúcar (Noreste y Suroeste) y dos regiones en la cual es escasa (Noroeste y Sureste),

Ejemplo: Sugarscape

- Los agentes tienen también un cierto nivel de visión, que les permite ver o no ver las reservas de azúcar.
- El modelo se puede ir complicando, agregando reglas de comportamiento y también del entorno:
 - Ejemplo: incorporar vidas finitas en los agentes por lo que mueren pasado un tiempo. En ese momento, son reemplazados por un nuevo agente cuyos parámetros son aleatorios.
 - Otras reglas: polución, dispersión de la polución, reproducción, transmisión cultural, comercio (se comercia con las especies)
 - Sólo se comercia si el beneficio de ambos agentes aumenta, por lo que se puede observar fenómenos como distribución de la riqueza y fijación de precios.

Ejemplo: Sugarscape

- Los agentes se mueven de casilla según su estrategia
- Posibles estrategias:
 - Un agente se mueve a la celda que prefiere, que esté dentro de su rango de visión para reponer las existencias de azúcar y especias
 - Pero también pueden intercambiar (trueque) con otros agentes vecinos
- Observación:
 - los agentes rápidamente tienden hacia las montañas de azúcar, especialmente aquellos con una visión más desarrollada
 - algunos individuos tienden a acumular grandes cantidades de azúcar, generando riqueza personal.
 - pocos agentes con visión corta y baja tasa metabólica logran subsistir en las zonas periféricas pero sin lograr acumular riqueza (se producen desigualdades)

Ejemplo: Sugarscape

Para probarlo:

- Descargar e instalar NetLogo
- Abrir “biblioteca de modelos”
- Seleccionar “Sugarscape 2 Constant Growback”

Simulación basada en agentes: ¿Se usa?

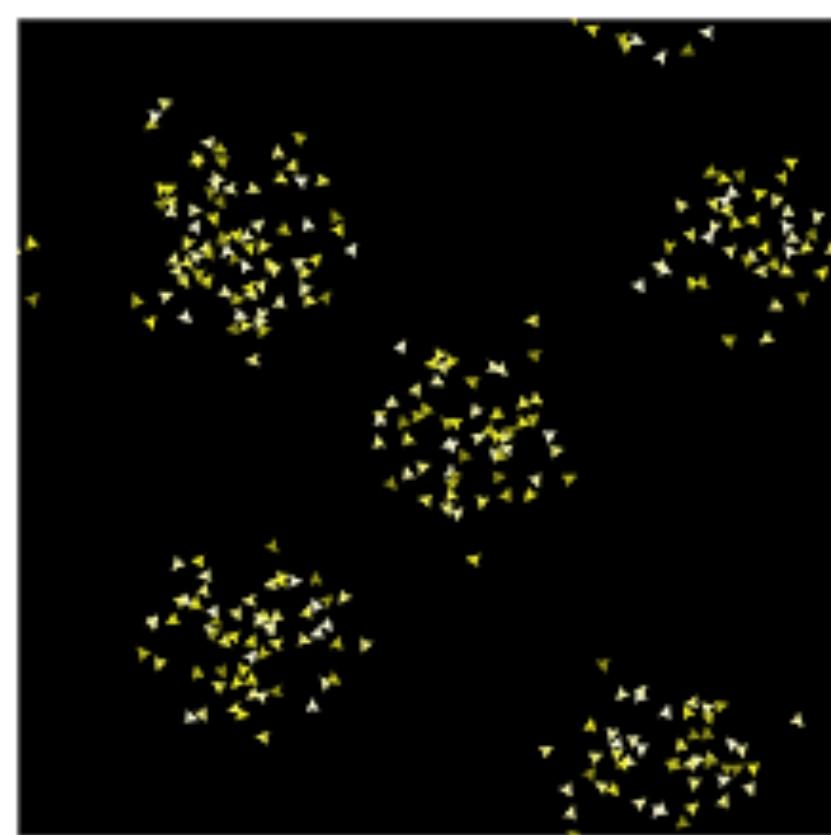


Ejemplos de modelos

- Vida artificial. Inteligencia colectiva
 - Modelo de Flocking. REYNOLDS, Craig W. Flocks, herds and schools: A distributed behavioral model.
- Simulación de multitudes
 - Modelo de Fuerza Social. Dirk Helbing y Peter Molnar en 1995 con varios avances en las últimas décadas.
 - Posteriormente, Illes Farkas y Tamas Vicsek contribuyeron a su desarrollo.
- Infecciones
 - Modelos epidemiológicos basados en SIR, SEIR, SEAIR
- Modelos económicos
 - Hamill, L., & Gilbert, G. N. (2016). Agent-based modelling in economics. Chichester: Wiley.
 - Fundamentos teóricos y metodológicos de la utilización de ABS en economía: modelos de demanda, de mercado, mercado laboral, mercado internacional, banca, ...
- Red eléctrica
 - Ringler, P., Keles, D., & Fichtner, W. (2016). Agent-based modelling and simulation of smart electricity grids and markets—a literature review. Renewable and Sustainable Energy Reviews, 57, 205-215.
- Transporte. Movilidad
 - Bazzan, A. L., & Klügl, F. (2014). A review on agent-based technology for traffic and transportation. The Knowledge Engineering Review, 29(3), 375.
- Salud
 - Tracy, M., Cerdá, M., & Keyes, K. M. (2018). Agent-based modeling in public health: current applications and future directions. Annual review of public health, 39, 77-94.

Ejemplos de modelos

- Vida artificial. Inteligencia colectiva
 - Modelo de Flocking. REYNOLDS, Craig W. Flocks, herds and schools: A distributed behavioral model.
- Simulación de multitudes
 - Modelo de Fuerza Social. Dirk Helbing y Peter M
 - Posteriormente, Illes Farkas y Tamas Vicsek con
- Infecciones
 - Modelos epidemiológicos basados en SIR, SEIR,
- Modelos económicos
 - Hamill, L., & Gilbert, G. N. (2016). Agent-based m
 - Fundamentos teóricos y metodológicos de la mercado, mercado laboral, mercado internaci
- Red eléctrica
 - Ringler, P., Keles, D., & Fichtner, W. (2016). Agent markets—a literature review. Renewable and Sustai
- Transporte. Movilidad
 - Bazzan, A. L., & Klügl, F. (2014). A review on age Knowledge Engineering Review, 29(3), 375.
- Salud
 - Tracy, M., Cerdá, M., & Keyes, K. M. (2018). Agent-based modeling in public health: current applications and future directions. Annual review of public health, 39, 77-94.



and

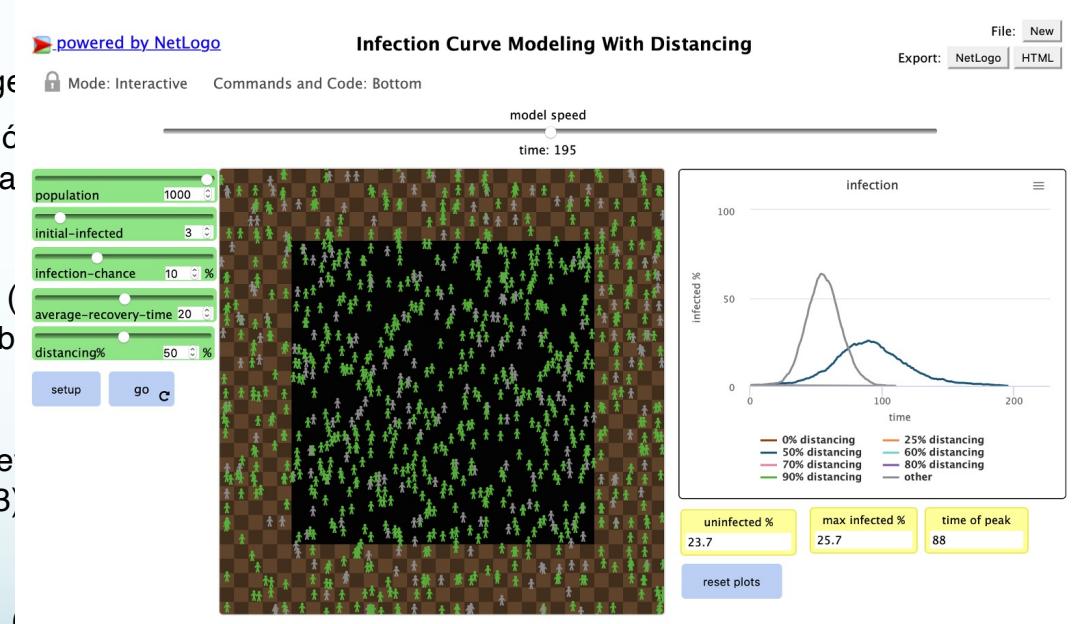
Ejemplos de modelos

- Vida artificial. Inteligencia colectiva
 - Modelo de Flocking. REYNOLDS, Craig W. Flocks, herds and schools: A distributed behavioral model.
- Simulación de multitudes
 - Modelo de Fuerza Social. Dirk Helbing y Peter Molnar en 1995 con varios avances en las últimas décadas.
 - Posteriormente, Illes Farkas y Tamas Vicsek contribuyeron a su desarrollo.
- Infecciones
 - Modelos epidemiológicos basados en agentes
- Modelos económicos
 - Hamill, L., & Gilbert, G. N. (2016). Agent-based modeling in economics: A review. *Journal of Economic Surveys*, 30(1), 1–39.
 - Fundamentos teóricos y metodológicos. Agent-based modeling in economics: A review. Mercado, mercado laboral, mercad
- Red eléctrica
 - Ringler, P., Keles, D., & Fichtner, W. (2012). Agent-based modeling of power systems: A review. *Renewable and Sustainable Energy Reviews*, 16(1), 102–114.
- Transporte. Movilidad
 - Bazzan, A. L., & Klügl, F. (2014). Agent-based modeling in transportation. *Journal of Knowledge Engineering Review*, 29(1), 1–20.
- Salud
 - Tracy, M., Cerdá, M., & Keyes, K. M. (2018). Agent-based modeling in public health: current applications and future directions. *Annual review of public health*, 39, 77–94.



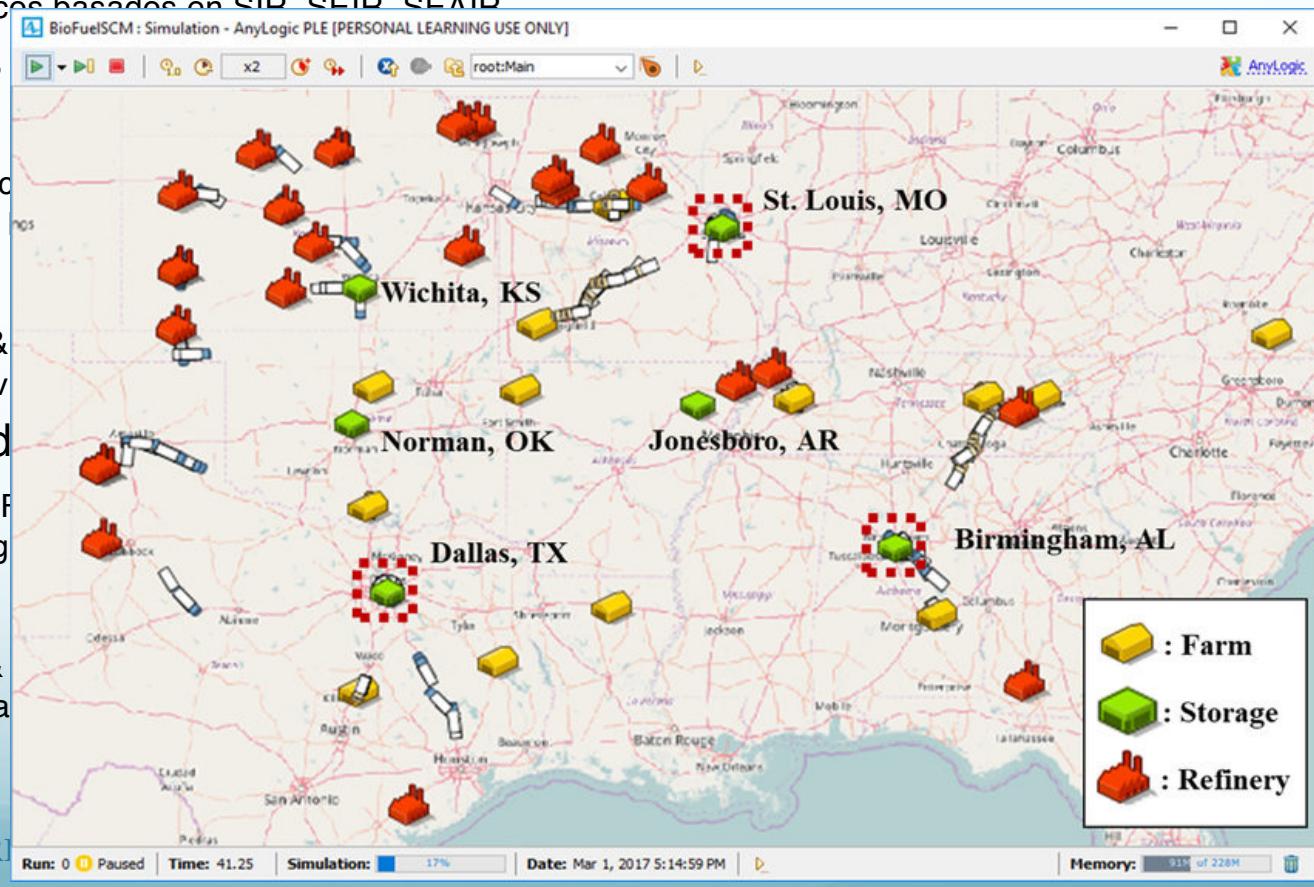
Ejemplos de modelos

- Vida artificial. Inteligencia colectiva
 - Modelo de Flocking. REYNOLDS, Craig W. Flocks, herds and schools: A distributed behavioral model.
- Simulación de multitudes
 - Modelo de Fuerza Social. Dirk Helbing y Peter Molnar en 1995 con varios avances en las últimas décadas.
 - Posteriormente, Illes Farkas y Tamas Vicsek contribuyeron a su desarrollo.
- Infecciones
 - Modelos epidemiológicos basados en SIR, SEIR, SEAIR
- Modelos económicos
 - Hamill, L., & Gilbert, G. N. (2016). Agent-based modeling of the market for energy efficiency. *Energy Markets—A Literature Review*. Renewables and Energy Efficiency, 1(1), 1–15.
 - Fundamentos teóricos y metodológicos: mercado, mercado laboral, mercados de bienes y servicios, etc.
- Red eléctrica
 - Ringler, P., Keles, D., & Fichtner, W. (2014). The role of energy efficiency in power markets—a literature review. *Renewable and Sustainable Energy Reviews*, 38, 10–20.
- Transporte. Movilidad
 - Bazzan, A. L., & Klügl, F. (2014). A review of agent-based models in traffic and transport. *Journal of Knowledge Engineering Review*, 29(3), 219–245.
- Salud
 - Tracy, M., Cerdá, M., & Keyes, K. M. (2018). The future of mental health: challenges and future directions. *Annual review of public health*, 39, 77–94.



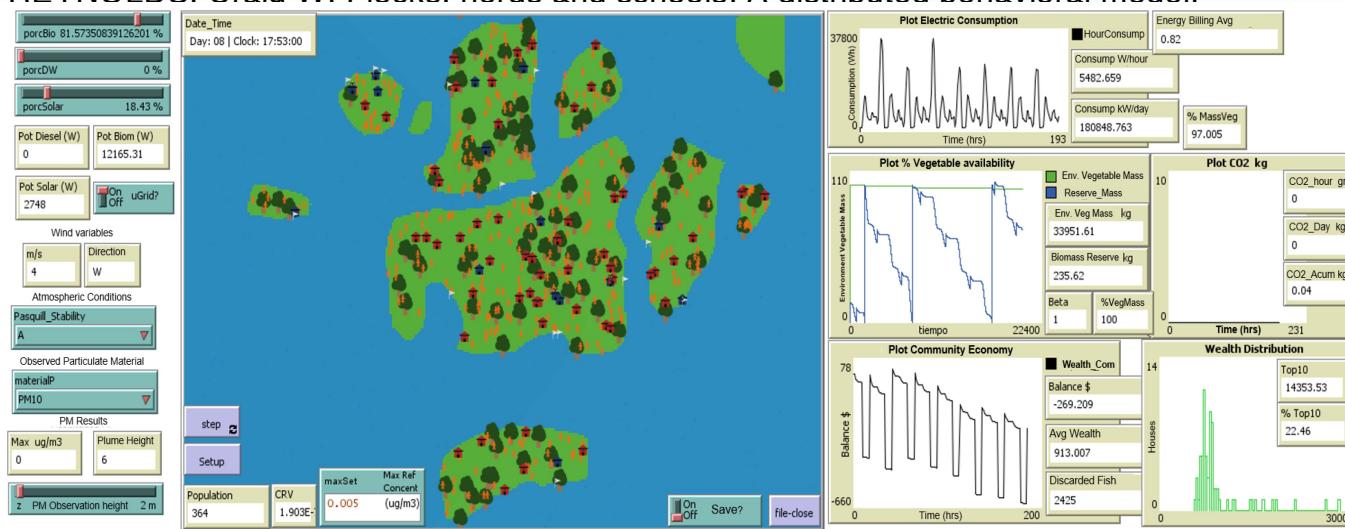
Ejemplos de modelos

- Vida artificial. Inteligencia colectiva
 - Modelo de Flocking. REYNOLDS, Craig W. Flocks, herds and schools: A distributed behavioral model.
- Simulación de multitudes
 - Modelo de Fuerza Social. Dirk Helbing y Peter Molnar en 1995 con varios avances en las últimas décadas.
 - Posteriormente, Illes Farkas y Tamas Vicsek contribuyeron a su desarrollo.
- Infecciones
 - Modelos epidemiológicos basados en SIR, SEIR, SEAIR
- Modelos económicos
 - Hamill, L., & Gilbert, G.
 - Fundamentos teóricos
 - mercado, mercado
- Red eléctrica
 - Ringler, P., Keles, D., & markets—a literature review
- Transporte. Movilidad
 - Bazzan, A. L., & Klügl, F. Knowledge Engineering
- Salud
 - Tracy, M., Cerdá, M., & future directions. Annual



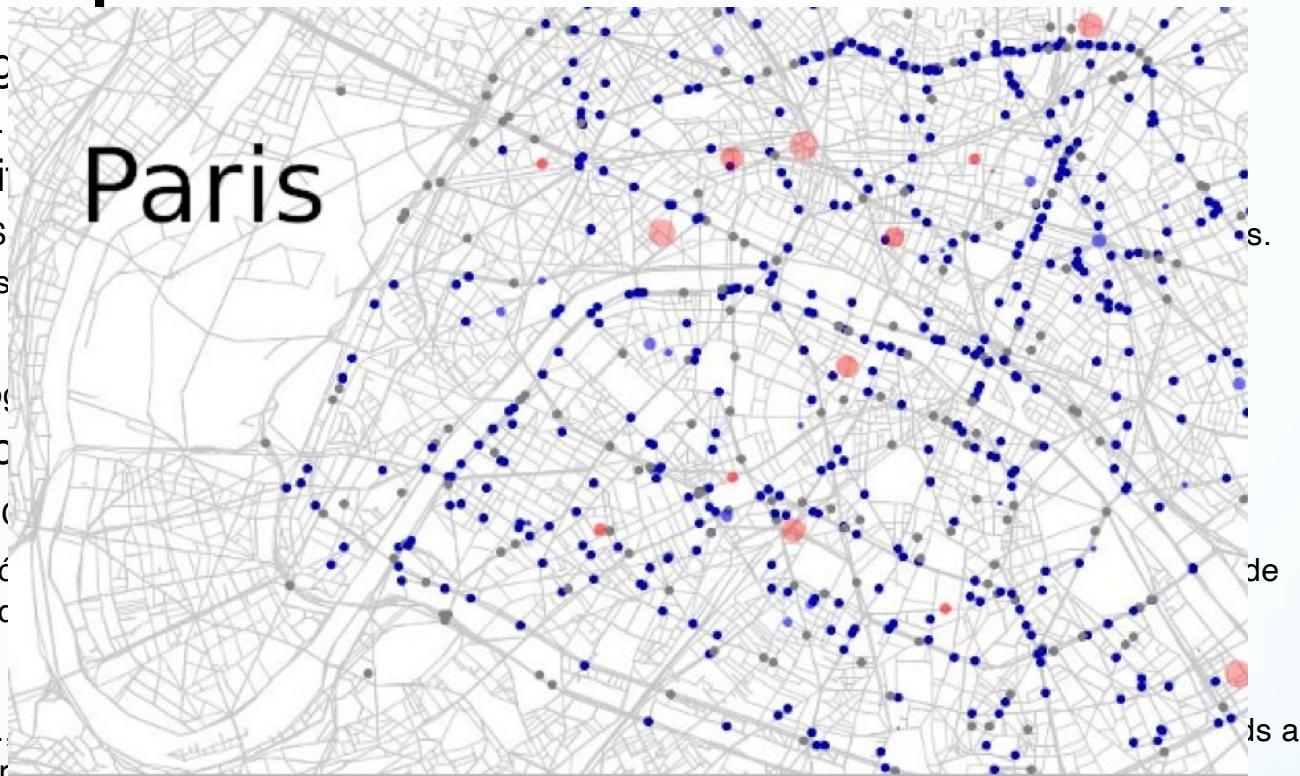
Ejemplos de modelos

- Vida artificial. Inteligencia colectiva
 - Modelo de Flocking. REYNOLDS. Craig W. Flocks, herds and schools: A distributed behavioral model.
- Simulación de multi
 - Modelo de Fuerza S
 - Posteriormente, Illes
- Infecciones
 - Modelos epidemiológicos
- Modelos económicos
 - Hamill, L., & Gilbert, C. (2014). Agent-based modeling of environmental health risks. *Journal of Health Politics, Policy and Law*, 39(3), 77-94.
 - Fundamentos tecnológicos: mercado, mercado, mercados
- Red eléctrica
 - Ringler, P., Keles, D., & Fichtner, W. (2016). Agent-based modelling and simulation of smart electricity grids and markets—a literature review. *Renewable and Sustainable Energy Reviews*, 57, 205-215.
- Transporte. Movilidad
 - Bazzan, A. L., & Klügl, F. (2014). A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, 29(3), 375.
- Salud
 - Tracy, M., Cerdá, M., & Keyes, K. M. (2018). Agent-based modeling in public health: current applications and future directions. *Annual review of public health*, 39, 77-94.



Ejemplos de modelos

- Vida artificial. Inteligencia Artificial
 - Modelo de Flocking.
- Simulación de multitudes
 - Modelo de Fuerza Social
 - Posteriormente, Illescas et al. (2018)
- Infecciones
 - Modelos epidemiológicos
- Modelos económicos
 - Hamill, L., & Gilbert, C. (2018). Agent-based modeling in economics: Fundamentos teóricos y aplicaciones. In *Agent-based modeling in economics: Fundamentos teóricos y aplicaciones* (pp. 1-20). Springer.
 - Fundamentos teóricos y aplicaciones en el mercado, mercados y sistemas de producción.
- Red eléctrica
 - Ringler, P., Keles, D., & Sener, M. (2018). Agent-based modeling in electricity markets—a literature review. *Renewable and Sustainable Energy Reviews*, 87, 260-270.
- Transporte. Movilidad
 - Bazzan, A. L., & Klügl, F. (2014). A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, 29(3), 375.
- Salud
 - Tracy, M., Cerdá, M., & Keyes, K. M. (2018). Agent-based modeling in public health: current applications and future directions. *Annual review of public health*, 39, 77-94.



Ejemplos de modelos

- Vida artificial. Inteligencia colectiva
 - Modelo de Flocking. REYNOLDS, Craig W. Flocks, herds and schools: A distributed behavioral model.
- Simulación de multitudes
 - Modelo de Fuerza Social. Dirk Helbing y Peter Molnar en 1995 con varios avances en las últimas décadas.
 - Posteriormente, Illes Farkas y Tamas Vicsek contribuyeron a su desarrollo.
- Infecciones
 - Modelos epidemiológicos basados en SIR, SEIR, SEAIR
- Modelos
 - Hamill, D., & Gitterman, S. (2005). Epidemic dynamics and agent-based modeling. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(07), 07015.
 - Fu, H., & Lee, C. (2014). Agent-based modeling for emergency medical services. *Journal of Health Politics, Policy and Law*, 39(2), 375-402.
- Red eléctrica
 - Ringle, M. A., & Winkler, J. (2014). Agent-based modeling in marketing. *Journal of Marketing Research*, 51(2), 231-245.
- Transporte
 - Bazzani, A. L., & Krugl, P. (2014). A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, 29(3), 375.
- Salud
 - Tracy, M., Cerdá, M., & Keyes, K. M. (2018). Agent-based modeling in public health: current applications and future directions. *Annual review of public health*, 39, 77-94.



Entornos de Simulación

- **NetLogo.** Entorno de programación de modelos para la simulación natural y social de fenómenos (Tisue y Wilensky, 2004). <https://ccl.northwestern.edu/netlogo/>
- **MESA. Entorno de simulación basado en NetLogo en Python**
<https://mesa.readthedocs.io/en/stable/>
- **REPAST.** Entorno para el modelado basado en agentes que utiliza un enfoque orientado a objetos. Desarrollado en Java. <https://repast.github.io/index.html>
- **MASON.** Entorno de simulación multiagente de eventos discretos desarrollado en java (Luke et al., 2004). <http://cs.gmu.edu/~eclab/projects/mason/>
- **GAMA.** Desarrollado en Java, incorpora su propio lenguaje de modelado GAML.
<http://gama-platform.org>
- **MATSim.** Especializado en simulación de transporte. <http://www.matsim.org>
- **AnyLogic.** Simulación multiparadigma (eventos discretos, dinámica de sistemas y ABM). <http://www.anylogic.com>
 - [Versión educación](#)
- **Simio.** Comercial y multiparadigma <https://www.simio.com/software/agent-based-modeling.php>
- **FlexSIM.** Añadido módulo de ABS. <https://www.flexsim.com/es/>
 - <https://www.flexsim.com/news/flexsim-2020-update-2-agent-based-features/>

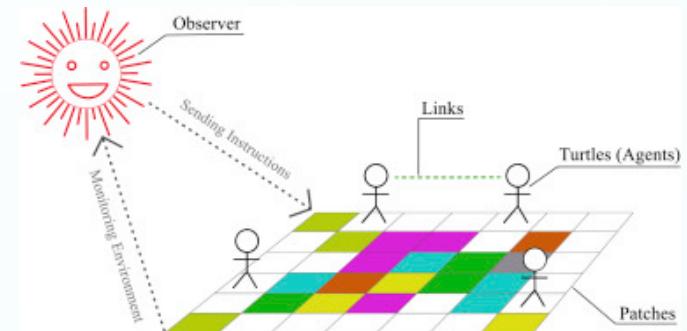
Sistemas Multiagente

Simulación basada en agentes
NetLogo (parte 2)

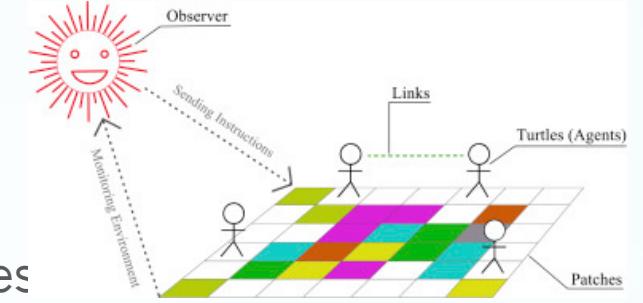
Authors: Vicent Botti, Vicente Julián

Netlogo

- Libre y open source
- Funciona sobre Windows, Mac OS X, Linux
- Lenguaje de programación propio
 - Diseñado para aprendizaje en la escuela
 - Descendiente lejano de Logo
- Buena documentación y tutoriales
- Lenguaje de alto nivel
 - Ideas complicadas pueden expresarse en una o dos líneas de código
 - Soporte gráfico
- Foro muy activo



NETLOGO



- Netlogo es un mundo 2D ó 3D con 4 tipos de agentes
 - patches* – parcelas del entorno (agentes estacionarios)
 - Son las subdivisiones del mundo, donde se ubican los Agentes
 - Se acceden mediante sus coordenadas.
 - Pueden tener estado y evolucionar
 - turtles* – tortugas, se mueven por los patches (agentes móviles)
 - Son los agentes en el modelo de simulación
 - Tienen un estado
 - Pueden moverse por el entorno
 - links* – enlaces entre tortugas
 - Se utilizan para expresar relaciones entre tortugas
 - Un observer* – observa y controla lo que hacen los agentes.
 - Es el diseñador que realiza modificaciones al entorno de vida de los agentes

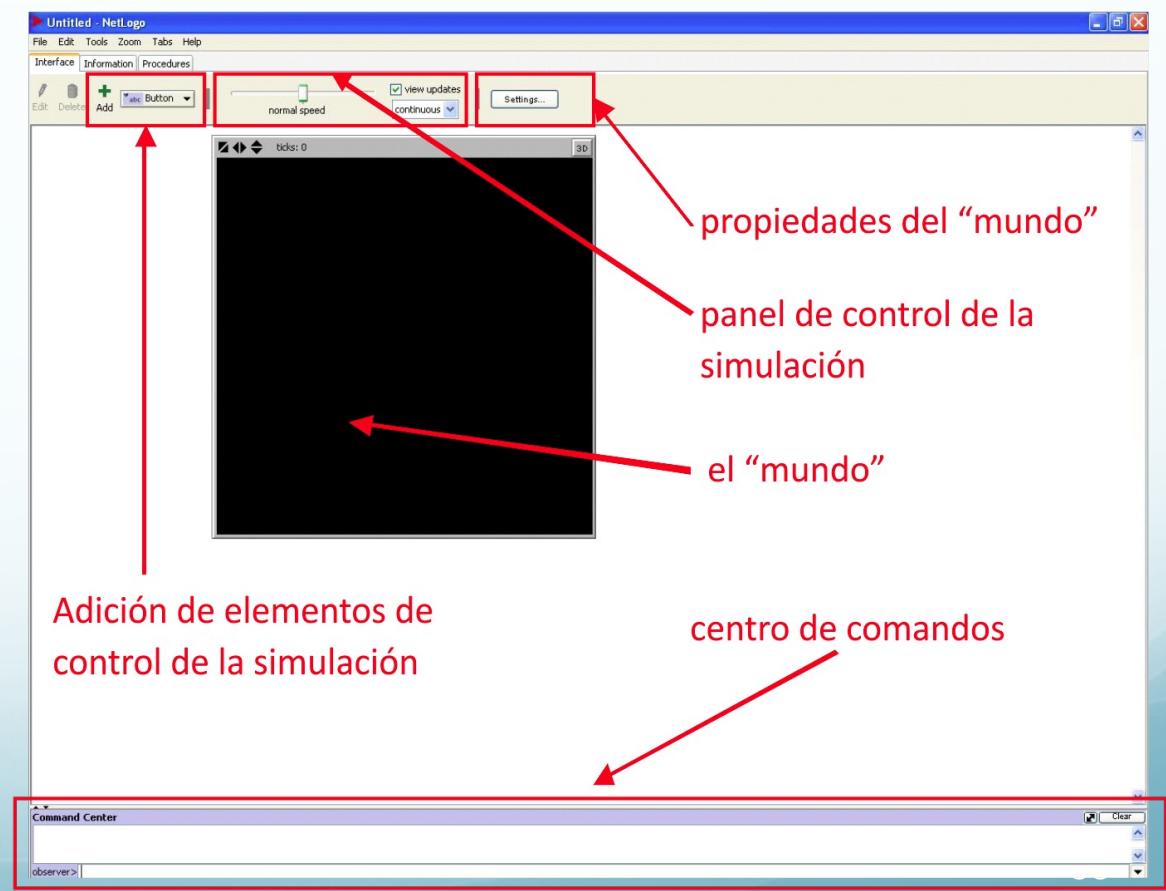
NETLOGO

- Pasos a seguir:
 - 1: Configurar el modelo: agentes, comportamientos, ...
 - 2: Configurar la visualización: controles, botones, gráficas,
...
 - 3: Ejecutar el modelo
 - 4: Analizar resultados
- Manual en castellano:
 - <https://ccl.northwestern.edu/netlogo/resources/Guia%20de%20programacion%20de%20NetLogo.pdf>

NETLOGO

■ Interfaz

- El mundo → rejilla configurable
- Elementos de control
- Centro de comandos



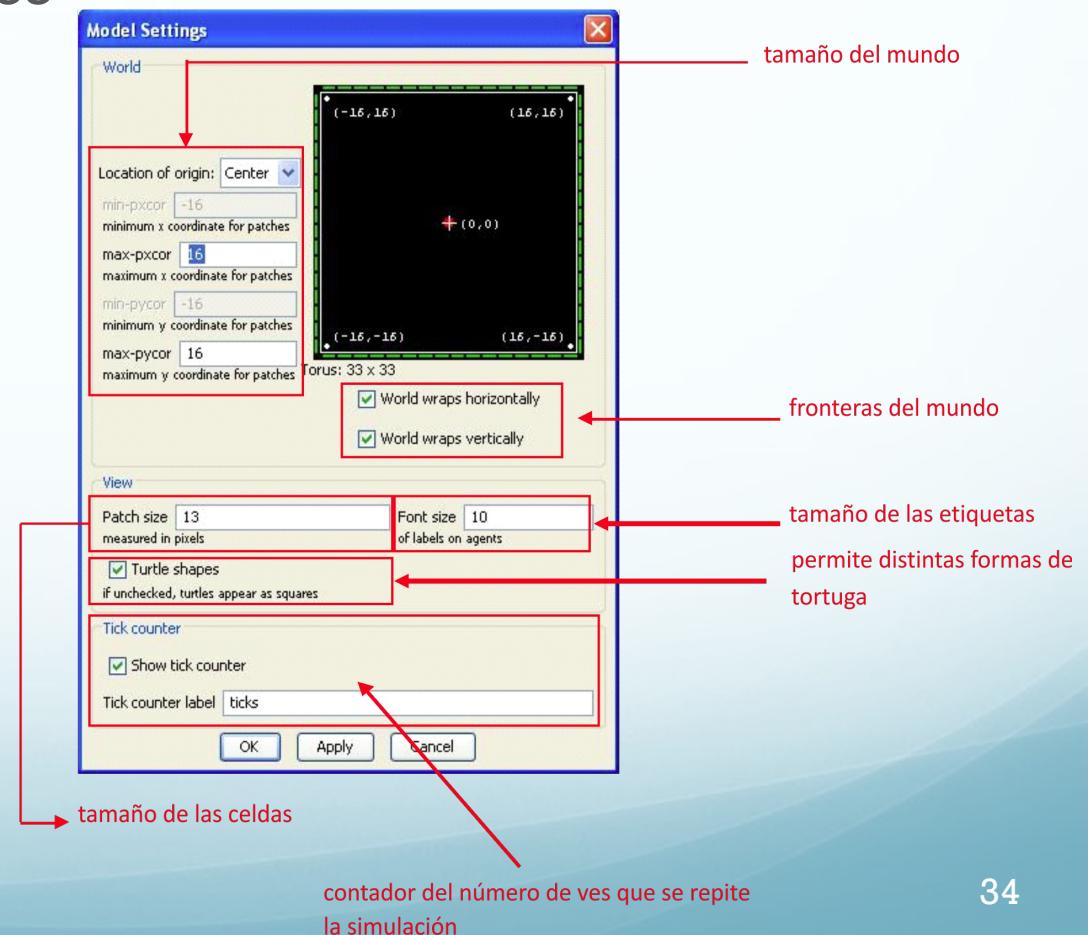
NETLOGO

■ Ajustes del modelo:

- Tamaño: max-pxcor y max-pycor → por defecto 16
- Equivale a rejilla de 33x33

Comandos:

- show world-width → anchura del mundo (por defecto 33 celdas)
- show world-height → altura del mundo (por defecto 33 celdas)
- show count patches → número de patches (celdas) , por defecto 1.089



NETLOGO

- Elementos de control y visualización:
 - BUTTON: Al pulsarlo, se ejecutan una serie de comandos.
 - SLIDER: Controla el valor de una variable global. Acota los valores de esta variable dentro de un rango, así como asignarle un valor por defecto.
 - SWITCH: Controla el valor de una variable global booleana.
 - CHOOSEN: Controla el valor de una variable global. Los valores que podrá tomar esta variable serán discretos.
 - INPUT: Controla el valor de una variable global. El usuario teclea dicho valor.
 - MONITOR: Muestra el valor que toma una variable global durante la ejecución del programa.
 - PLOT: Representación gráfica.
 - OUTPUT: Se trata de una pantalla en la que podemos mostrar mensajes a lo largo de la ejecución. Sólo se permite añadir un output por modelo.
 - NOTE: Texto.

NETLOGO. Primeros Pasos

- Primitivas básicas
 - **clear-all (ca)** limpia la pantalla
 - **create-turtles (crt)** para crear tortugas → crt 2 ;; crea dos tortugas
 - **ask**: pedir a los agentes que realicen acciones
 - **forward** ;; (fd) avanzar
 - **back** ;; (bk) retroceder
 - **left** ;; (lt) giro a la izquierda
 - **right** ;; (rt) giro a la derecha
 - **repeat** ;; repetir un conjunto de primitivas
 - **pen-up** ;;(pu) subir el lápiz (los agentes no deja huella al moverse)
 - **pen-down** ;;(pd) bajar el lápiz (los agentes dejan trazos al moverse)

ask turtles [fd 10]

ask turtles [fd 5 rt 90 fd 5 rt 90 fd 5 rt 90 fd 5]

NETLOGO. Primeros Pasos

- Procedimientos

- Es un comando que engloba comandos NetLogo

```
to comenzar ;; definimos el nombre del procedimiento
  ca ;; clear-all
  crt 1 ;; creamos una tortuga
  ask turtles [pd repeat 4 [fd 5 rt 90] pu] ;; dibujamos el cuadrado
end ;; fin del procedimiento
```

- Este procedimiento creará una tortuga, y ésta dibujará un cuadrado.
- Se pueden crear botones asociados a estos procedimientos

NETLOGO. Primeros Pasos

- **Procedimientos con parámetros**

```
to dibujar_poligono [num-sides len]
  pen-down
  repeat num-sides [ fd len rt 360 / num-sides ]
end

ask turtles [dibujar_poligono 8 who ]
```

- **Funciones**

```
to-report valor_absoluto [number]
  ifelse number >= 0 [ report number ]
    [ report (- number) ]
end

show valor_absoluto 15
```

NETLOGO. Primeros Pasos

- Cada uno de los agentes que creamos, ya sean tortugas o patches tienen una serie de propiedades.
- las tortugas tienen las siguientes propiedades predefinidas:
 - who ;; identificador (no se puede modificar)
 - color ;; color
 - heading ;; orientación
 - xcor ;; coordenada x
 - ycor ;; coordenada y
 - shape ;; forma
 - label ;; etiqueta
 - label-color ;; color de la etiqueta
 - breed ;; raza
 - hidden? ;; visible o no visible
 - size ;; tamaño
 - pen-size ;; tamaño del trazo al desplazarse (cuando pen-mode=down)
 - pen-mode ;; dejar trazo al desplazarse o no

NETLOGO. Primeros Pasos

- Los patches tienen como propiedades:
 - pxcor ;; coordenada x del patch
 - pycor ;; coordenada y del patch
 - pcolor ;; color del patch
 - plabel ;; etiqueta del patch
 - plabel-color ;; color de la etiqueta del patch
- Ejemplo
 - to comenzar
 - ca
 - ask patches [set pcolor blue]
 - crt 1
 - ask turtles [set color orange set heading 0]
 - ask turtles [pd repeat 4 [fd 5 rt 90] pu]
 - end

NETLOGO

- Se pueden añadir propiedades adicionales
 - Uso de la primitiva **own**

turtles-own [edad] ;;añadimos la propiedad "edad" a las tortugas

to comenzar

```
ca                      ;; clear-all
crt 1 [set edad 7]      ;; crea una tortuga y le asigna una edad de 7
crt 1 [set edad 10]     ;; crea otra tortuga y le asigna una edad de 10
ask turtles [show edad] ;; pide a las tortugas que muestren su edad
end
```

NETLOGO

- ¿Cómo seleccionamos agentes que tienen una propiedad concreta?

ask "agentes" with ["propiedad"] ["acciones"]

- Ejemplos

ask turtles with [color = red] [fd 5]

ask turtles with [color = blue] [set size 4]

ask turtles with [color = white] [set label "soy blanca" show label]

ask turtles with [ycor > 0] [fd 3]

ask patches with [pxcor < 0] [set pcolor yellow]

NETLOGO

- ¿Cómo crear diferentes tipos de agentes?
 - Uso de la propiedad **breed** (raza)
 - Ej: breed [coches coche] → se indica el nombre en plural y en singular
 - Se le pueden añadir nuevas propiedades (uso de own)
 - A partir de aquí se pueden configurar y enviar ordenes:
 - create-coches 20
 - ask coches [fd 10]

NETLOGO

- ¿Cómo crear diferentes tipos de agentes?
 - Uso de la propiedad **breed** (raza)

breed [coches coche]

breed [aviones avion]

to comenzar

ca

create-coches 20

create-aviones 10

ask coches [set shape "car" fd 10]

ask aviones [set shape "airplane" fd 2]

end

NETLOGO

Otros aspectos:

- Definir variables globales

```
globals [  
    variable_global_1  
    variable_global_2  
    variable_global_n ]
```

...

```
set variable_global_1 5
```

- Definir variables locales

```
let variable_local 5
```

...

```
set variable_local 7
```

NETLOGO

Otros aspectos:

- Sentencias de control

ifelse condición

[;; comandos que se ejecutan si se cumple la condición]

[;; comandos que se ejecutan si no se cumple la condición]

while condición

[;; comandos]

repeat n

[;; comandos]

NETLOGO

Evaluación de experimentos (Behavior Space)

- BehaviorSpace es una herramienta integrada que permite diseñar y ejecutar experimentos.
- BehaviorSpace ejecuta un modelo repetidas veces variando los valores de ciertas variables y almacenando los resultados.
- Permite explorar el espacio del modelo para analizar que combinación causa comportamientos de interés.

NETLOGO

Extensiones

- NetLogo permite crear nuevos comandos en Java (u otros lenguajes) y usarlos en sus modelos
- Las extensiones existentes pueden consultarse en <https://github.com/NetLogo/NetLogo/wiki/Extensions>

Declaración:

```
extensions [ext1 ext2 ...]
```

Uso:

```
ext1:comando
```

NETLOGO

Más documentación en:

<http://ccl.northwestern.edu/netlogo/docs/>

Muy útil:

<http://ccl.northwestern.edu/netlogo/docs/dictionary.html>

Veamos algún ejemplo