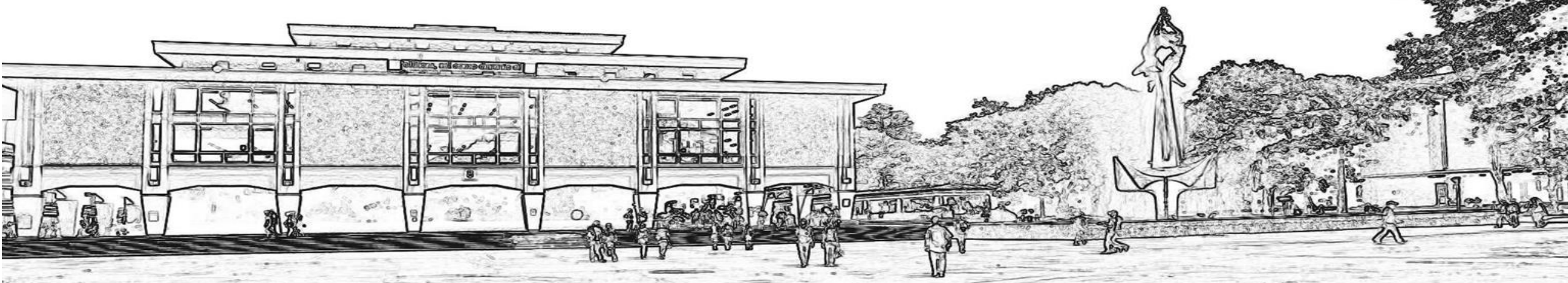




UNIVERSIDAD DE ANTIOQUIA

Facultad de Ingeniería



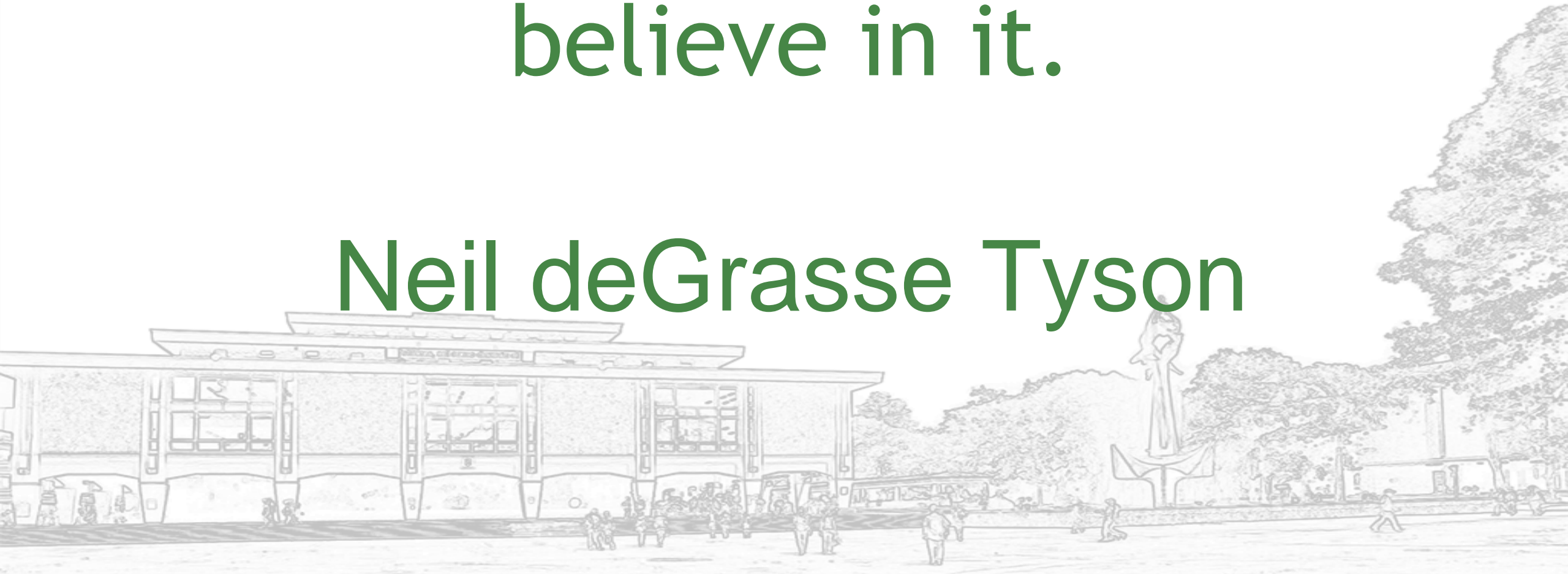
Simulación basada
en agentes

Julián Andrés Castillo Grisales
Ingeniero de Sistemas
Magister en Ingeniería
jandres.castillo@udea.edu.co

Docente de pregrado y posgrado UdeA
Docente Ocasional Tiempo Completo IUDigital de Antioquia

The good thing about science is
that it's true whether or not you
believe in it.

Neil deGrasse Tyson



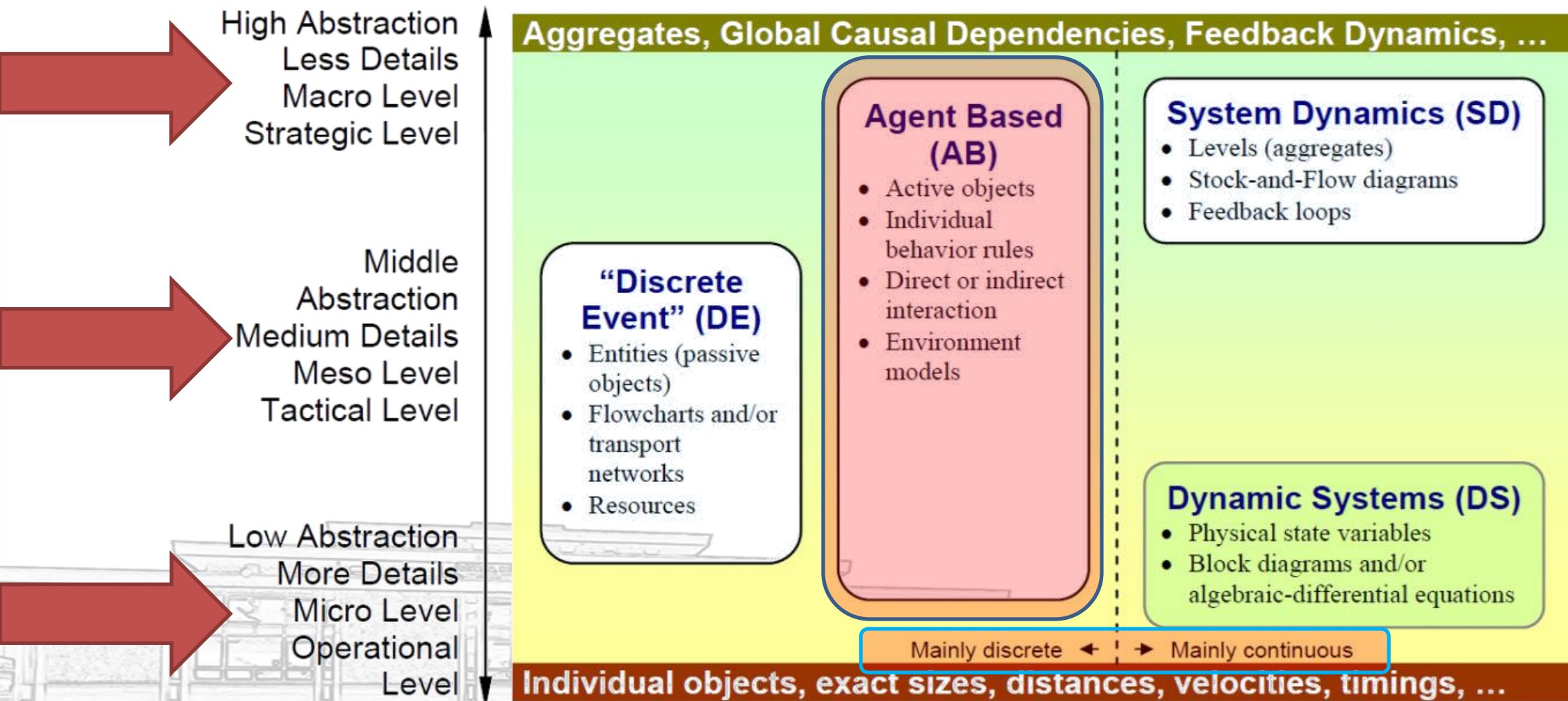
Palabras clave

1. ABS o ABM o ABMS o SBA: Agent Based Simulation / Agent Based Modelling / Agent Based Modeling / Agent Based Modeling and Simulation / Simulación Basada en Agentes.
2. EBM: Equation Based Modeling
3. CA: Cellular automata
 - Also called cellular spaces, tessellation automata, homogeneous structures, cellular structures, tessellation structures, and iterative arrays
4. ABSS: Agent Based Social Simulation
5. AI: Artificial Intelligence
6. API: Application Programming Interface
7. CAS: Complex Adaptive Systems
8. MAS: Multi Agent systems
9. SAAS: Social Aspects Of Agent Systems
10. ACA: Asynchronous Cellular Automaton
11. SCA: Stochastic Cellular Automaton
12. IBM: Individual Based Modeling

¿Por qué Simulación Basada en Agentes?

1. Un mundo no tan ideal.
 - Tratar de entender la naturaleza y los mundos sociales.
2. Un agente es un individuo/objeto computacional autónomo con un conjunto de propiedades y acciones.
 - Agentes en plural
3. La simulación basada en agentes es una forma de modelado/simulación computacional en donde un fenómeno es modelado en términos de agentes y sus interacciones.
 - Nuevos tópicos, nuevos mundos y nuevas simulaciones.
4. La simulación basada en agentes es una metodología que puede ser aplicada promiscuamente.
 - En los últimos 20 años ABM ha crecido casi exponencialmente.

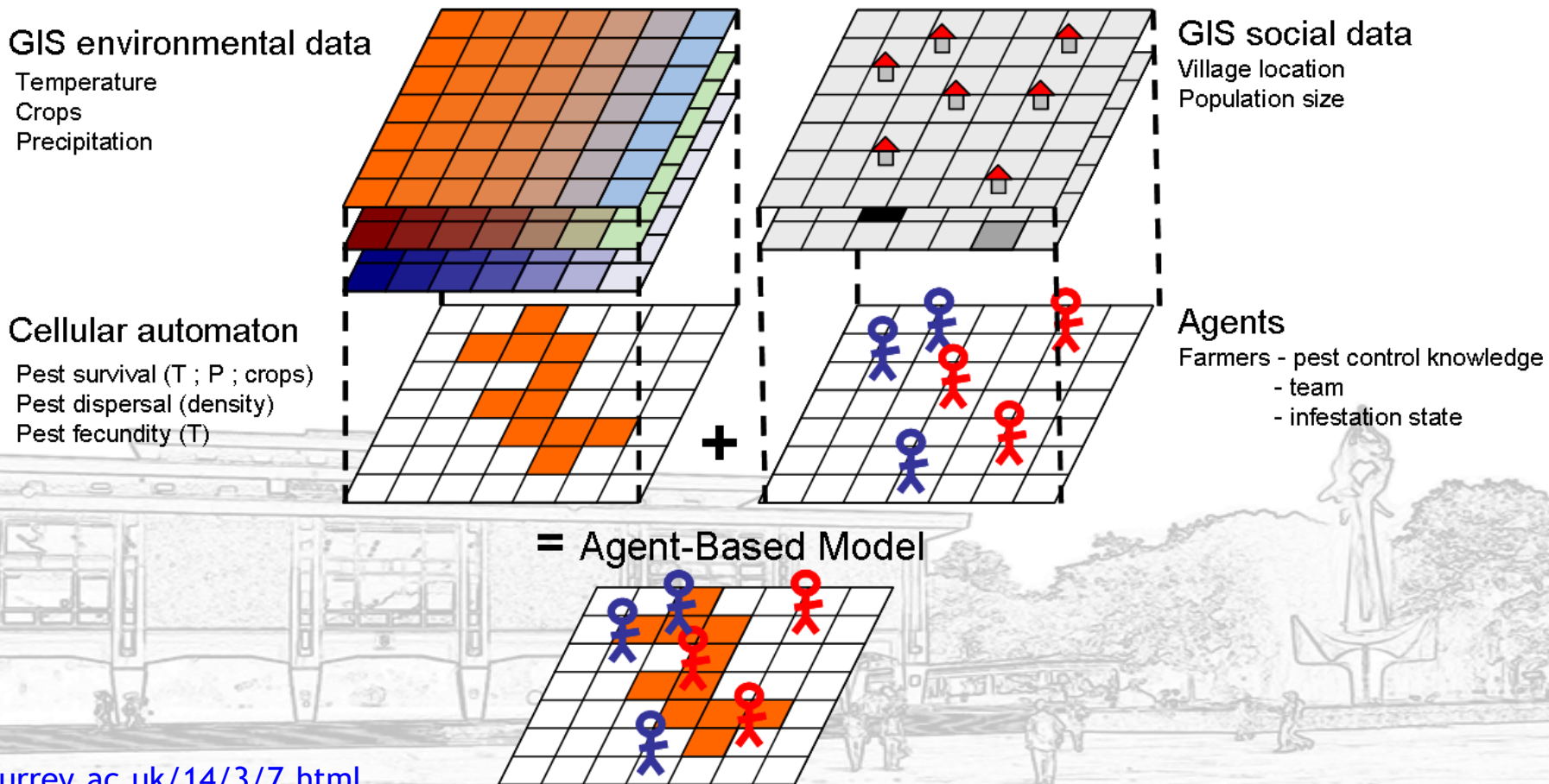
Simulación basada en agentes



Simulación basada en agentes

Ejemplo

An Example of ABM in Agriculture Pest Control Studies. Source: University of Surrey



Simulación basada en agentes

Percepción general*

La simulación basada en agentes permite a los individuos con su propio grupo de características y reglas poder ser creados. Estos individuos pueden ser puestos o ubicados en ambientes o recreaciones realísticas y sus interacciones (tanto con si mismo como con el entorno) pueden ser graficadas.

La capacidad para capturar y poder entender el comportamiento individual de actores heterogéneos a través de múltiples escalas espaciales y temporales nos brinda una riqueza única para comprender los procesos y los impulsores que dan forma a los sistemas sociales y geográficos.

Simulación basada en agentes

Características de los agentes

Cada grupo o área de conocimiento tiene su propia definición de SBA y tienen varias connotaciones acerca de las características de un agente, a continuación se realiza una lista de los atributos clave tomados de varias definiciones.

Autonomía: Los agentes son autónomos, esto significa que ellos actúan sin influencia directa de un control externo y centralizado. Los agentes toman sus propias decisiones gracias a un proceso independiente e interno, estas decisiones son basadas en la información que los agentes poseen y que pueden intercambiar con otros agentes o con el entorno.



Simulación basada en agentes

Características de los agentes

Heterogeneidad: Los agentes son heterogéneos, por ejemplo un agente que representa un humano puede tener atributos tales como: edad, genero, trabajo y otros. Grupos de agentes pueden ser incorporados y representar clases diferentes como por ejemplos deportistas, pueden incluir futbolistas, maratonistas, gimnastas, ciclistas y otros los cuales pueden tener atributos independientes y comportamientos independientes.



Simulación basada en agentes

Características de los agentes

Actividad o acciones: Los agentes realizan acciones independientes durante el curso de la simulación, las siguientes características suelen realizarse por los agentes y son comúnmente modeladas.

- **Basado en objetivos:** a los agentes se les asignan objetivos los cuales pretenden cumplir a lo largo de la simulación.
- **Reactivo o perceptivo:** los agentes se les asignan características para estar activos o reactivos a una situación en especial o la presencia de otro agente en la simulación. Los agentes pueden tener o no conocimiento previo de ciertas características.
- **Racionalidad:** en la mayoría de las ciencias sociales el agente suele programarse para tomar la decisión mas racional del paradigma de elección del agente en el cual debe cumplir una serie de parámetros descritos por Axelrod en 2007.
- **Interacción o comunicación:** los agentes tienen la capacidad de interactuar o comunicarse con otros agentes incluyendo el entorno en variedad de situaciones.
- **Movilidad:** agentes pueden moverse libremente alrededor del modelo permitiéndoles obtener información del entorno para tomar decisiones e interactuar libremente con otros agentes.

Simulación basada en agentes

Características de los agentes

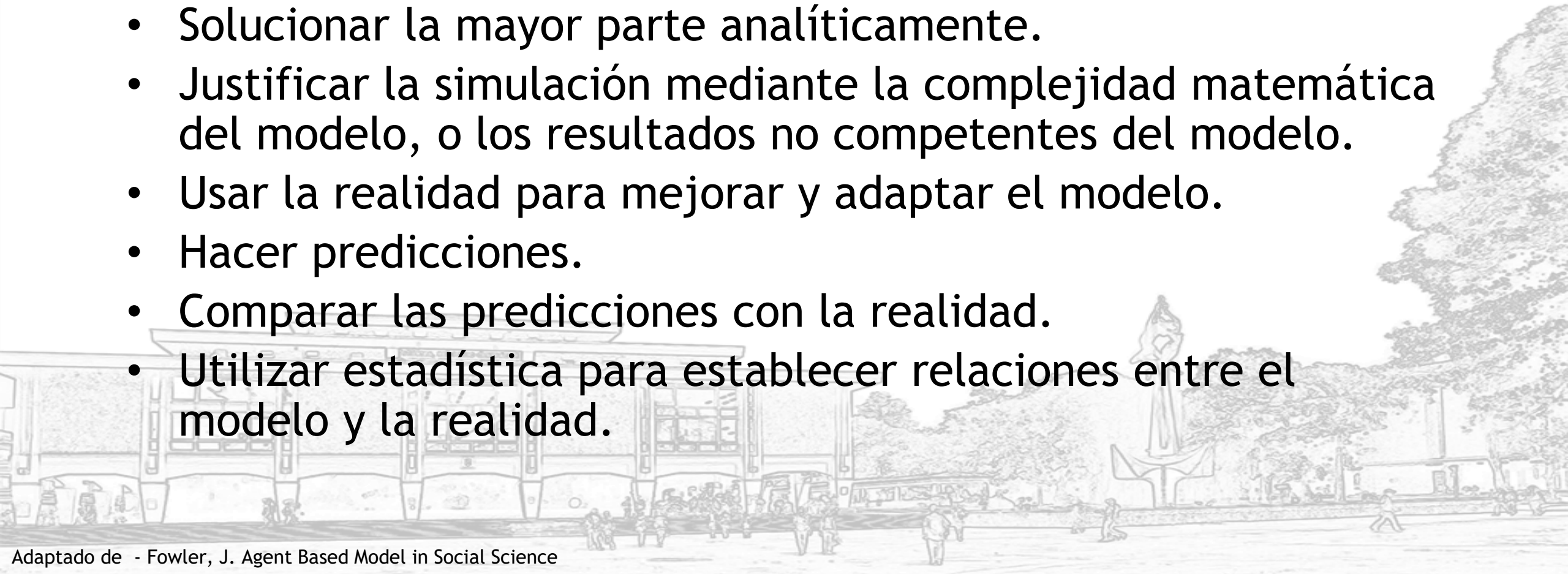
Aprendizaje y adaptación: los agentes pueden tener la habilidad de aprender y adaptarse. Se les puede modelar para tener la capacidad de modificar su estado actual con base en estados anteriores o uso de memoria.

La lista previa puede ser extensa y no significa que sea exhaustiva. Con la cantidad de productos con agentes realizados anualmente y un notable crecimiento en el uso de ABM el presente resumen de características solo puede crecer.

REGLAS: los agentes para cumplir el grupo de características descritas deben cumplir un conjunto de reglas para poder establecer los criterios descritos previamente. Las reglas son conjuntamente relacionadas con las cláusulas “if-then-else” y el detalle de aprendizaje y adaptación es relacionado al uso de vectores que almacenan el conocimiento adquirido.

Pasos para generar un modelo basado en agentes

- Dibujar el modelo
- Solucionar la mayor parte analíticamente.
- Justificar la simulación mediante la complejidad matemática del modelo, o los resultados no competentes del modelo.
- Usar la realidad para mejorar y adaptar el modelo.
- Hacer predicciones.
- Comparar las predicciones con la realidad.
- Utilizar estadística para establecer relaciones entre el modelo y la realidad.



El Mundo de ABM

Varios mundos: ABM tiene mundo 1D, 2D y 3D

1D: Autómatas celulares

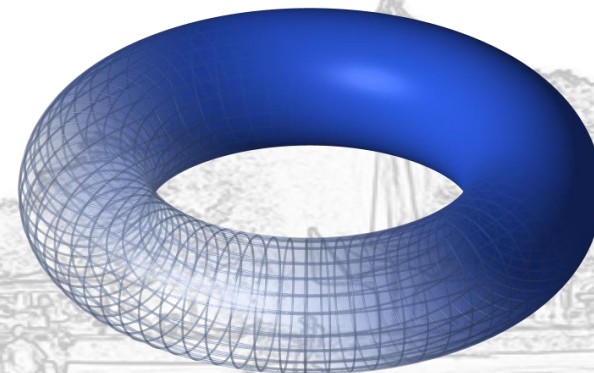
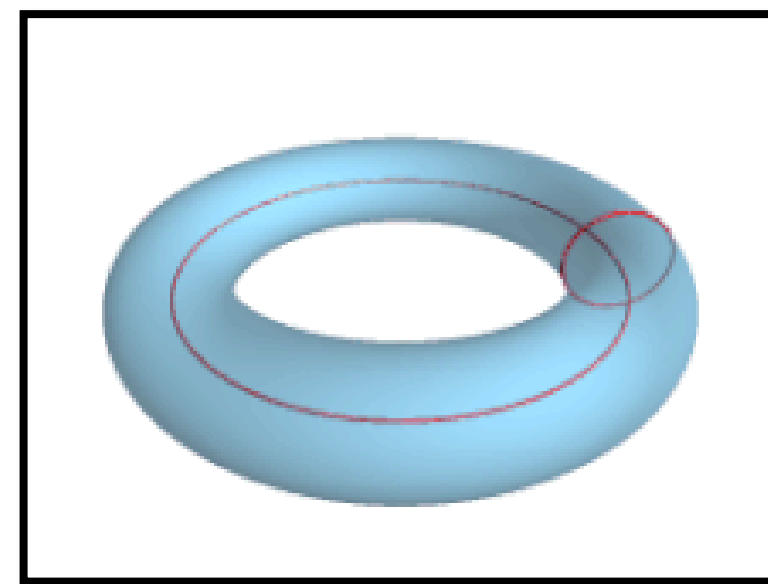
0	1	0	0	1	0	0	1	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---

2D: Autómatas celulares y Agentes

0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
0	0	0	0	1	1	1	0	1	1	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	1	1	1	0	0	0	0	0	0	1	1	1	0	0
0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1

3D: Autómatas celulares y Agentes

Un Torus es una superficie de revolución generada por la rotación de un círculo en el espacio tridimensional sobre un eje que es coplanar con el círculo y asemeja a un universo continuo muy utilizado en ABM.



El Mundo 2D

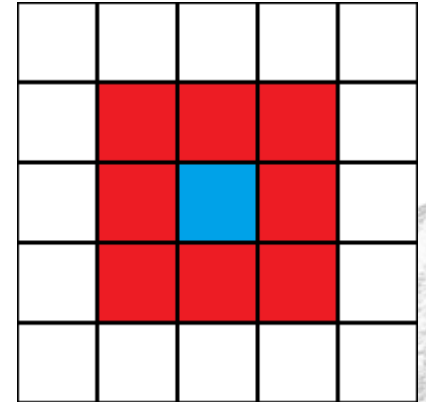
Vecindarios: ABM tiene mundo 1D, 2D y 3D

Todos se basan en la idea de las celdas ubicadas en una cuadrícula o malla, pero varían en las reglas utilizadas para actualizar los estados de las celdas y en su definición de qué celdas son vecinas. Los CA pueden utilizar varios conceptos de vecindario, un autómata 1D solo tiene izquierda o derecha pero en un mundo 2D existen varias opciones.

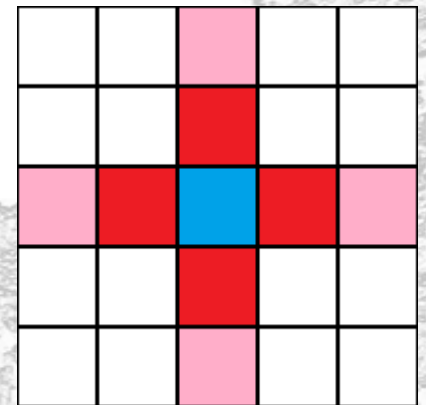
La vecindad se basa en celdas adyacentes. Los dos tipos más comunes de vecindarios son el vecindario de von Neumann y el vecindario de Moore.



1D



2D Moore



2D von Neuman

Simulación Basada en Agentes

Clasificación

1. Autómatas Celulares

- a) Unidimensionales (Elemental)
- b) Bidimensional
- c) Bidimensional cíclico
- d) Reversible
- e) Totalitario

2. Simulación basada en agentes

- a) Modelamiento individual
- b) Modelamiento basado en agentes



Autómatas celulares

Autómatas celulares o del ingles “Cellular Automaton (CA)” es un modelo basado en celdas objetos con las siguientes características:

- **CELDA:** Las celdas viven en una cuadrícula. (Existen unidimensionales, bidimensionales e incluso un numero finito de dimensiones)
- **ESTADO:** Cada celda tiene un estado. El número de posibilidades de estado suele ser finito. El ejemplo más simple tiene las dos posibilidades de 1 y 0 (de lo contrario se conoce como "on" y "off" o "alive" y "dead")
- **VECINDARIO:** Cada celda tiene un vecindario. Esto se puede definir de muchas maneras, pero normalmente es una lista de celdas adyacentes.



Autómata celular elemental o unidimensional

¿Cuál es el autómata mas simple que existe?

La respuesta es que con el modelo de CA más simple imaginable, veremos las propiedades de los sistemas complejos en funcionamiento.

Con base en la definición necesitamos tres elementos fundamentales.

1. Celdas: El mas sencillo es una línea unidimensional de celdas continuas.



2. Estado: El estado mas simple es 1 y 0, encendido y apagado.



3. Vecindario: El vecindario en una dimensión son sus vecinos de izquierda y derecha.



Autómata celular elemental o unidimensional

El detalle mas importante para el CA es el tiempo.

No es tiempo real o tiempo específico de humanos o GMT (Greenwich Mean Time), por el contrario CA tiene un tiempo específico al modelo limitado al periodo o tiempo específico que puede ser llamado “Generación” o también en ingles “frame count”.

Las figuras de color verde en la diapositiva anterior nos muestran CA en el tiempo igual a 0 o generación 0. Ahora

¿Cómo calculamos los estados de todas las celdas en la generación 1? ¿Y la generación 2? Y así sucesivamente.

Autómata celular elemental o unidimensional

Generación 0	1	0	1	0	1	1	1	0	0	1
Generación 1	?	?	?	?	?	?	?	?	?	?
Generación 2	?	?	?	?	?	?	?	?	?	?

Los criterios para establecer un cambio o un avance en los estados de las celdas se puede interpretar como las relaciones entre vecindario con relación a una celda llamada “*CELDA*” y las interacción es un tiempo “*t*” se detalla con el siguiente pseudocódigo.

$$ESTADO\ CELDA\ en\ t = f(CELDA\ Vecindario\ (t - 1))$$

En otras palabras el estado de una celda depende de una función que detalla el estado y su vecindario en un periodo de tiempo previo al actual.

Autómata celular elemental o unidimensional

Calculamos el presente estado de una CELDA de acuerdo a las reglas del estado de tiempo anterior.

Generación 0



Generación 1



Podemos mirar todas las configuraciones posibles de una CELDA y su vecino y definir el resultado del estado para cada configuración posible. Tenemos tres celdas, cada una con un estado de 0 o 1.

¿Cuántas formas posibles podemos configurar los estados?

Tres celdas definen un número de 3 bits, y con esos 3 bits puedes contar hasta 8.



Autómata celular elemental o unidimensional

Debemos definir una regla luego de tener todos los datos de configuración de estados. Para el presente ejercicio la regla será de la siguiente manera.

000	001	010	011	100	101	110	111
↓	↓	↓	↓	↓	↓	↓	↓
0	1	0	1	1	0	1	0

Según Wolfram el estándar para iniciar es tener todas las celdas en cero excepto la celda del centro que inicia en 1. (Se recomienda una configuración impar para tener un centro específico)

0	0	0	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---

Autómata celular elemental o unidimensional

Aplicamos el siguiente set de reglas descritas previamente como se muestra a continuación a un estado inicial

000	001	010	011	100	101	110	111
↓	↓	↓	↓	↓	↓	↓	↓
0	1	0	1	1	0	1	0

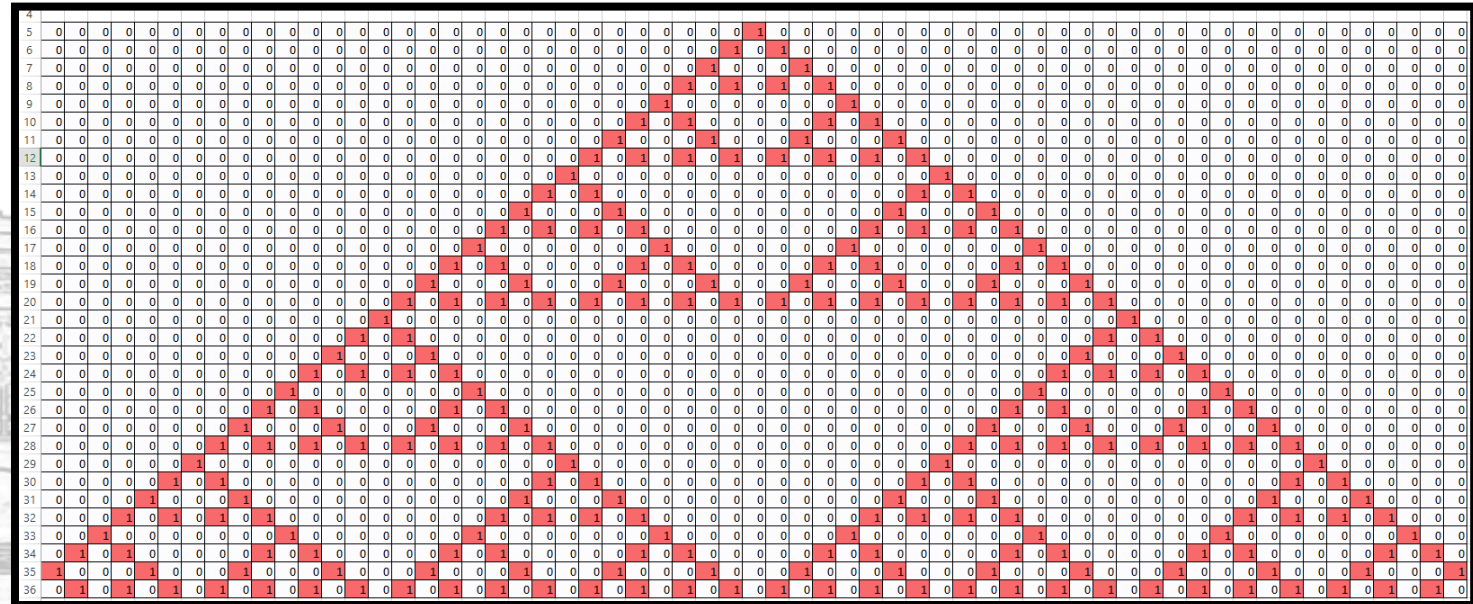
Se define el estado inicial

0	0	0	0	1	0	0	0	0
↓	↓	↓	↓	↓	↓	↓	↓	↓
0	0	0	1	0	1	0	0	0

Realizamos las operaciones

El triángulo de Sierpiński

La forma de baja resolución (pocas celdas) que realizamos en la diapositiva anterior es el "triángulo de Sierpiński". Nombrado en honor al matemático polaco Wacław Sierpiński, es un patrón fractal y un sistema increíblemente simple de 0 y 1 con pequeños vecindarios de tres celdas que pueden generar una forma tan sofisticada y detallada como el triángulo Sierpiński.



Autómatas Bidimensionales

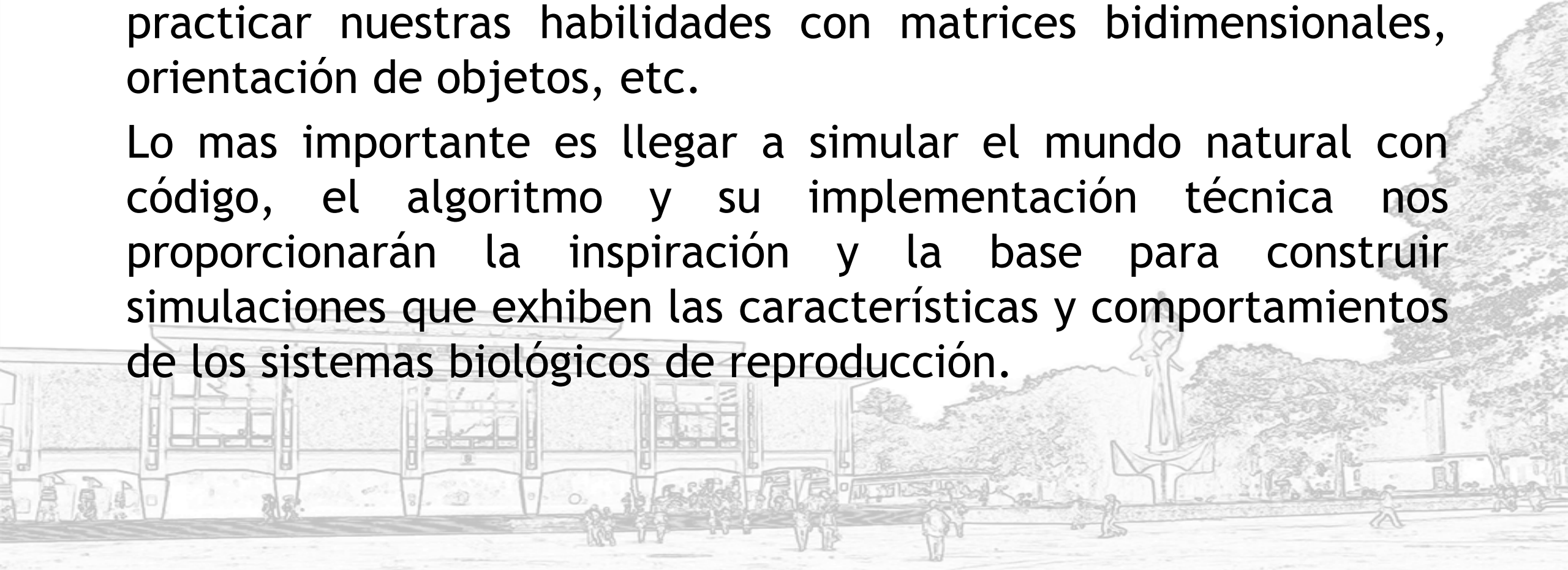
Al pasar de una a dos dimensiones agregamos mayor complejidad al sistema, cada celda tendrá un vecindario mayor y como se describió previamente podemos utilizar el vecindario von Neuman, von Neuman extendido, Moore o incluso un vecindario específico.

Para comprender mejor el modelo bidimensional utilizaremos el famoso “**Game of Life**” detallado en 1970 en el artículo “Scientific American” que documentó el matemático John Conway describiéndolo como “recreacional” y que el lector podría sacar un juego de ajedrez y simplemente “jugar”.

Autómatas Bidimensionales

“The Game of Life” proporciona una buena oportunidad para practicar nuestras habilidades con matrices bidimensionales, orientación de objetos, etc.

Lo mas importante es llegar a simular el mundo natural con código, el algoritmo y su implementación técnica nos proporcionarán la inspiración y la base para construir simulaciones que exhiben las características y comportamientos de los sistemas biológicos de reproducción.



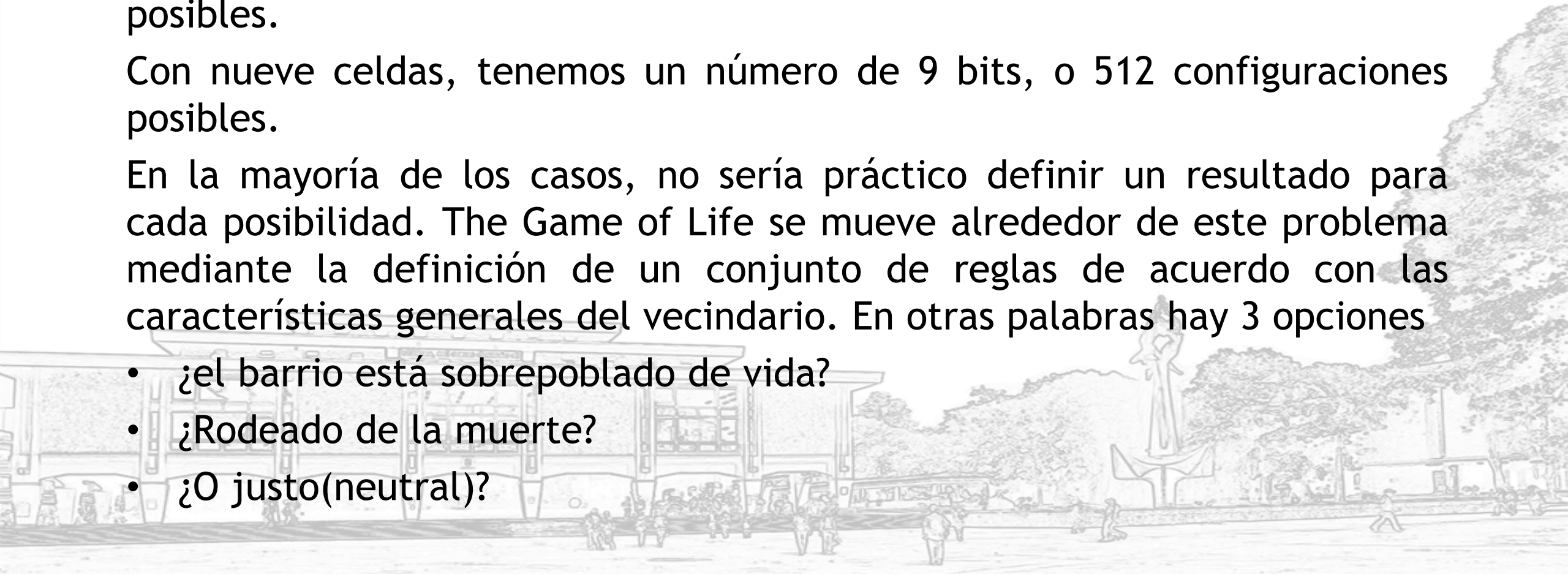
The Game of Life

Con tres celdas, tenemos un número de 3 bits con ocho configuraciones posibles.

Con nueve celdas, tenemos un número de 9 bits, o 512 configuraciones posibles.

En la mayoría de los casos, no sería práctico definir un resultado para cada posibilidad. The Game of Life se mueve alrededor de este problema mediante la definición de un conjunto de reglas de acuerdo con las características generales del vecindario. En otras palabras hay 3 opciones

- ¿el barrio está sobrepoblado de vida?
- ¿Rodeado de la muerte?
- ¿O justo(neutral)?



The Game of Life

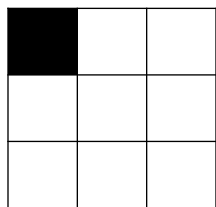
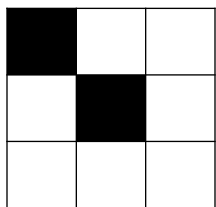
Las reglas son simples y son las siguientes:

- Muerte: una celda viva *estado* = 1 muere y pasa a *estado* = 0 bajo las siguientes circunstancias:
 - Sobrepoblación: si la celda tiene cuatro o mas vecinos vivos muere.
 - Soledad: si la celda tiene uno o cero vecinos muere.
- Nacimiento: una celda muerta *estado* = 0 vive o nace y pasa a *estado* = 1 si tiene exactamente tres vecinos vivos, ni mas ni menos que tres.
- Neutral: En los demás casos la celda no cambia, los casos son los siguientes:
 - Mantenerse con vida: una celda con dos o tres vecinos permanece viva.
 - Permanece muerta: una celda muerta con menos de tres vecinos vivos permanece muerta.

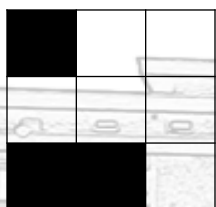
The Game of Life

Ejemplos de las reglas

Muerte, solo un vecindario vivo



Vida, tres vecindarios vivos



El Juego trata cada generación como un solo fotograma en una animación. Así que en lugar de ver todas las generaciones a la vez, solo vemos una a la vez, y el resultado se asemeja a bacterias de rápido crecimiento.

Puedes encontrar los ejemplos creados en el presente taller en:

https://github.com/juliancastillo-udea/JornadaConferencias_IyS_2022



Simulación basada en agentes

Sistemas complejos

El estudio de sistemas complejos se ha convertido en una importante característica del estudio científico. Por el término sistema complejo nos referimos a un sistema compuesto por muchas partes que interactúan.

Los sistemas complejos ven un crecimiento acelerado a mitad de los 80's y nace de las múltiples uniones entre la economía, la física y la ecología.



Simulación basada en agentes

Emergencia

Emergencia es una propiedad clásicamente exhibida por los modelos de agentes, y ocurre cuando un atributo que puede ser descrito en un nivel del sistema no es específicamente codificado en el nivel individual.

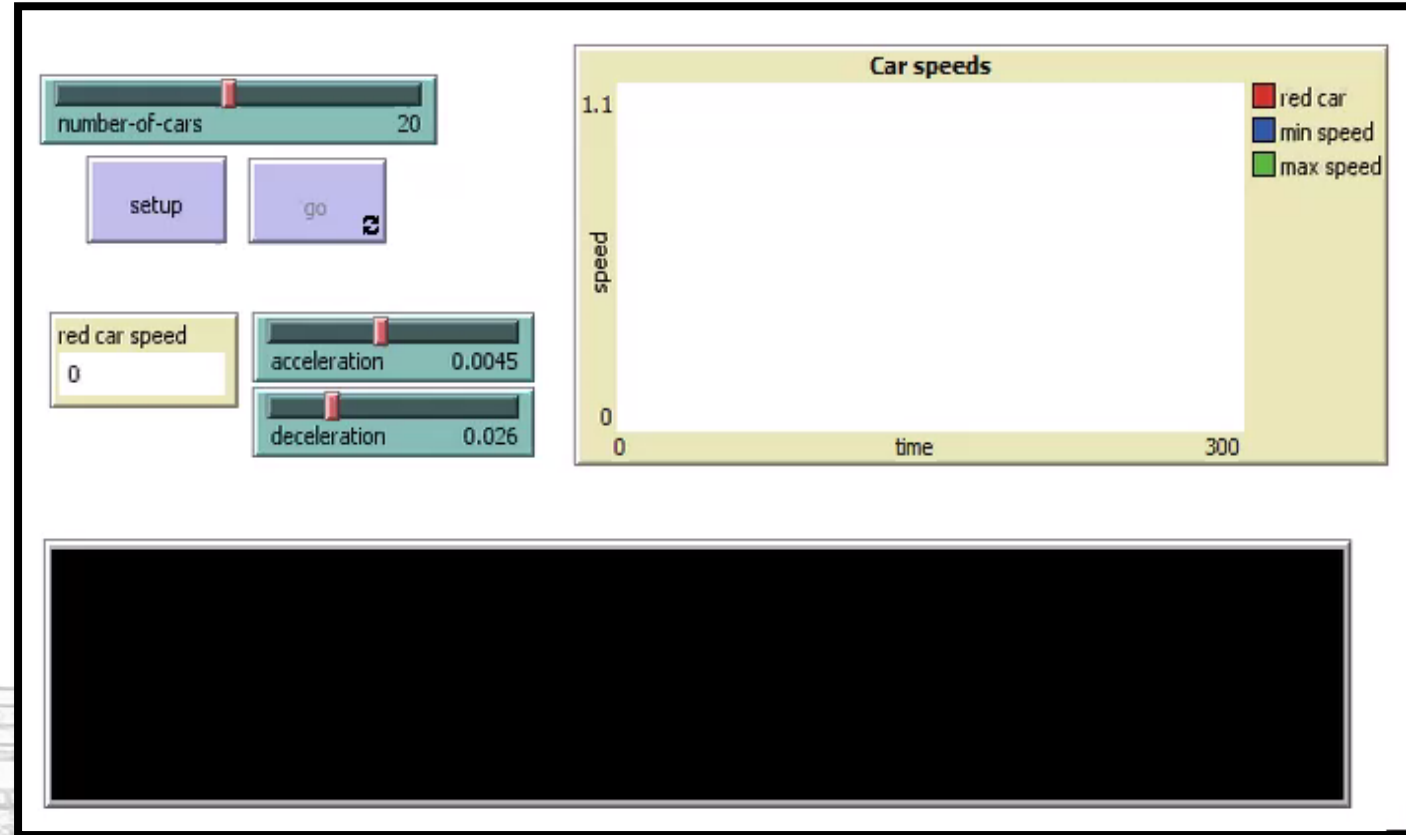
Esto puede llevar a “sorpresas” cuando las codificaciones a nivel individual resulta en un inesperado comportamiento del sistema a nivel macro.

La mayoría de las personas cuando encuentran con un comportamiento emergente, tienen a no identificarlo y responsabilizar el comportamiento a modelos o resultados determinísticos de control centralizado.

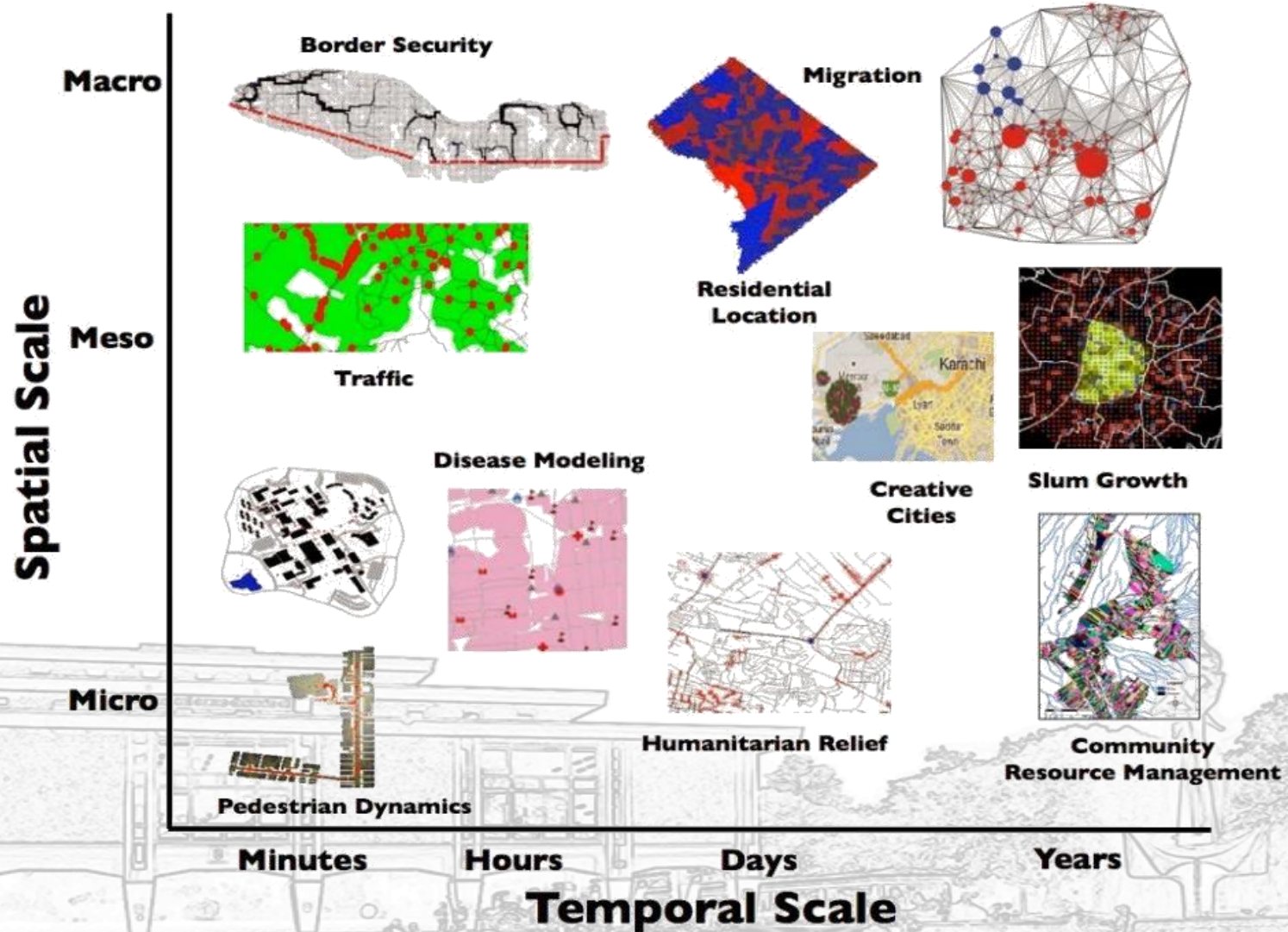
Sistemas complejos son caracterizados por fenómenos o patrones emergentes que suelen parecer complejos resultan de reglas simples.

Comportamiento emergente

- Se explica como el resultado de la simulación de una población de agentes “suelos” con reglas simples en un espacio o ambiente.
- Los fenómenos emergentes son patrones que surgen a partir de las interacciones descentralizadas de componentes individuales más simples.



Escala temporal y espacial de ABM



Software de Uso Especifico

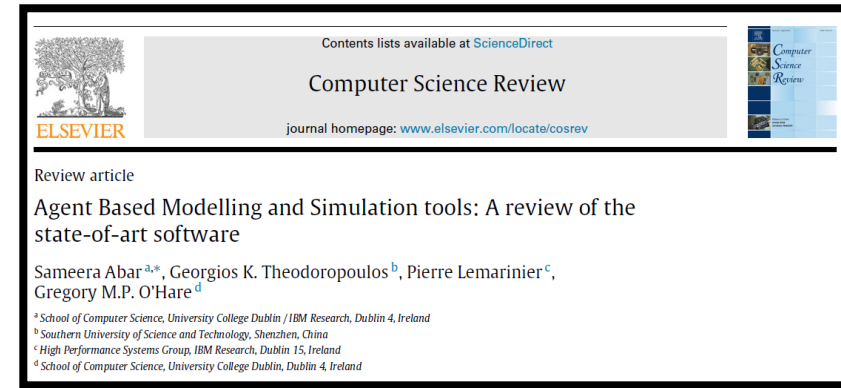
Simulación basada en agentes

Existen varios, incluso muchos programas de uso específico para el desarrollo y elaboraciones de simulaciones basadas en agentes, el autor Sameera Abar y su equipo de trabajo en el año 2017, realiza un documento en el cual detalla cada Software disponibles para desarrollo de ABS llamado: Agent Based Modelling and Simulation tools: A review of the state-of-the-art software.

Los Software de uso específico más utilizados en ABM son (SCOPUS 2022):

- NetLogo (6638)
 - CloudSim (5990)
 - AnyLogic (2242)
- 14.870

<https://www.sciencedirect.com/science/article/abs/pii/S1574013716301198>





NetLogo fue diseñado por Uri Wilensky, con base en el lenguaje de programación Logo.

Enseña conceptos de programación utilizando agentes en forma de tortugas, parches, enlaces y el observador. NetLogo fue diseñado para múltiples audiencias en mente sin una necesidad alta de programación para los fenómenos relacionados con el modelo.

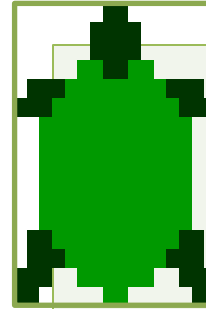
<https://en.wikipedia.org/wiki/NetLogo>

NetLogo graphical user interface	
Paradigms	multi-paradigm : educational , procedural , agent-based , simulation
Family	Lisp
Designed by	Uri Wilensky
Developer	Northwestern University Center for Connected Learning and Computer-Based Modeling
First appeared	1999; 21 years ago
Stable release	6.1.1 / September 26, 2019; 12 months ago
Typing discipline	Dynamic , strong
Scope	Lexical
Implementation language	Scala , Java
Platform	IA-32 , x86-64
OS	Cross-platform : JVM
License	GPL
Filename extensions	.nlogo, .nlogo3d, .nls
Website	ccl.northwestern.edu/netlogo



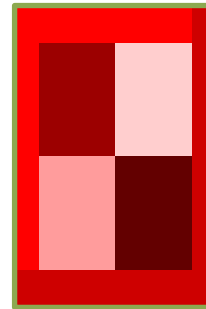
Los agentes en NetLogo tienen características detalladas y datos encapsulados (atributos y propiedades) que detallan métodos de comportamiento.

Los agentes se dividen de la siguiente manera:



Turtles

- Agentes
- Movimiento



Patches

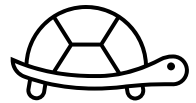
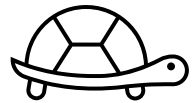
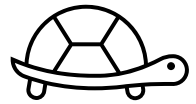
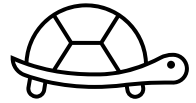
- Espacio cuadrado
- Tortugas interactúan con el



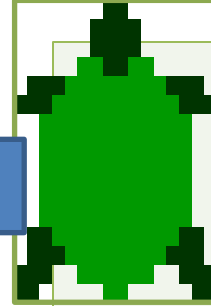
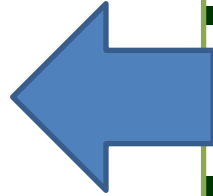
Observer

- Observador externo
- Control

NetLogo

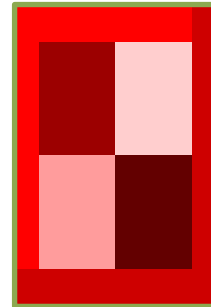


pen down



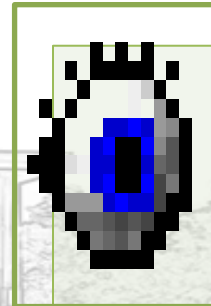
Turtles

- Agentes
- Movimiento



Patches

- Espacio cuadrado
- Tortugas interactúan con el

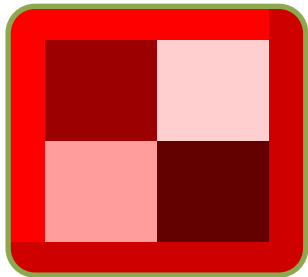
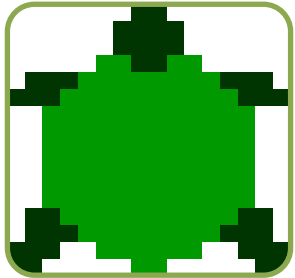


Observer

- Observador externo
- Control



Agentes



- Data & Variables
 - Humano
 - Genero, Edad, Raza, Estatura, etc.
- Procedimientos
 - Agentes – Condiciones
 - Mover si es infeliz, envejecer, cambiar color, cambiar y acumular energía, perder energía.

Agentes Patches

Los parches o del ingles (patches) es una malla o mundo en donde se encuentran distribuidos los parches de forma discreta.

Cada parche tiene atributos, tanto programables como integrados.

- `pxcor`, `pycor` relacionan a las coordenadas de localización de un parche.
- Los atributos programados pueden ser incorporados con la clausula `patches-own`.

Agentes Turtles

Las tortugas o del ingles (turtles) son los agentes con movimiento encima de los parches, múltiples tortugas pueden estar en el mismo parche.

Cada tortuga tiene atributos, tanto programables como integrados.

- `xcor`, `ycor` relacionan a las coordenadas de localización de una tortuga.
- Los atributos programados pueden ser incorporados con la clausula `turtles-own`.

Agentes Agentsets

Un “agentset” es un conjunto de agentes.

Existen configuraciones definidas por NetLogo para llamar a los conjuntos de agentes.

- turtles: todas las tortugas en el mundo.
- patches: todos los parches en el mundo.

Los agentsets pueden recibir instrucciones y selecciones.

A faint, grayscale illustration of a city scene serves as a background for the lower half of the slide. It depicts a large building with a flat roof, a fountain with a tall central column, and several people walking in the foreground.

Agentes

Variables globales

Las variables globales pueden ser accedidas por todos los agentes y tienen un solo valor.

Las variables globales pueden ser asignadas a objetos los cuales el modelo puede modificar en ejecución y afectar el estado del modelo en pre-ejecución, ejecución y post-ejecución.





Parches en color negro y verde.

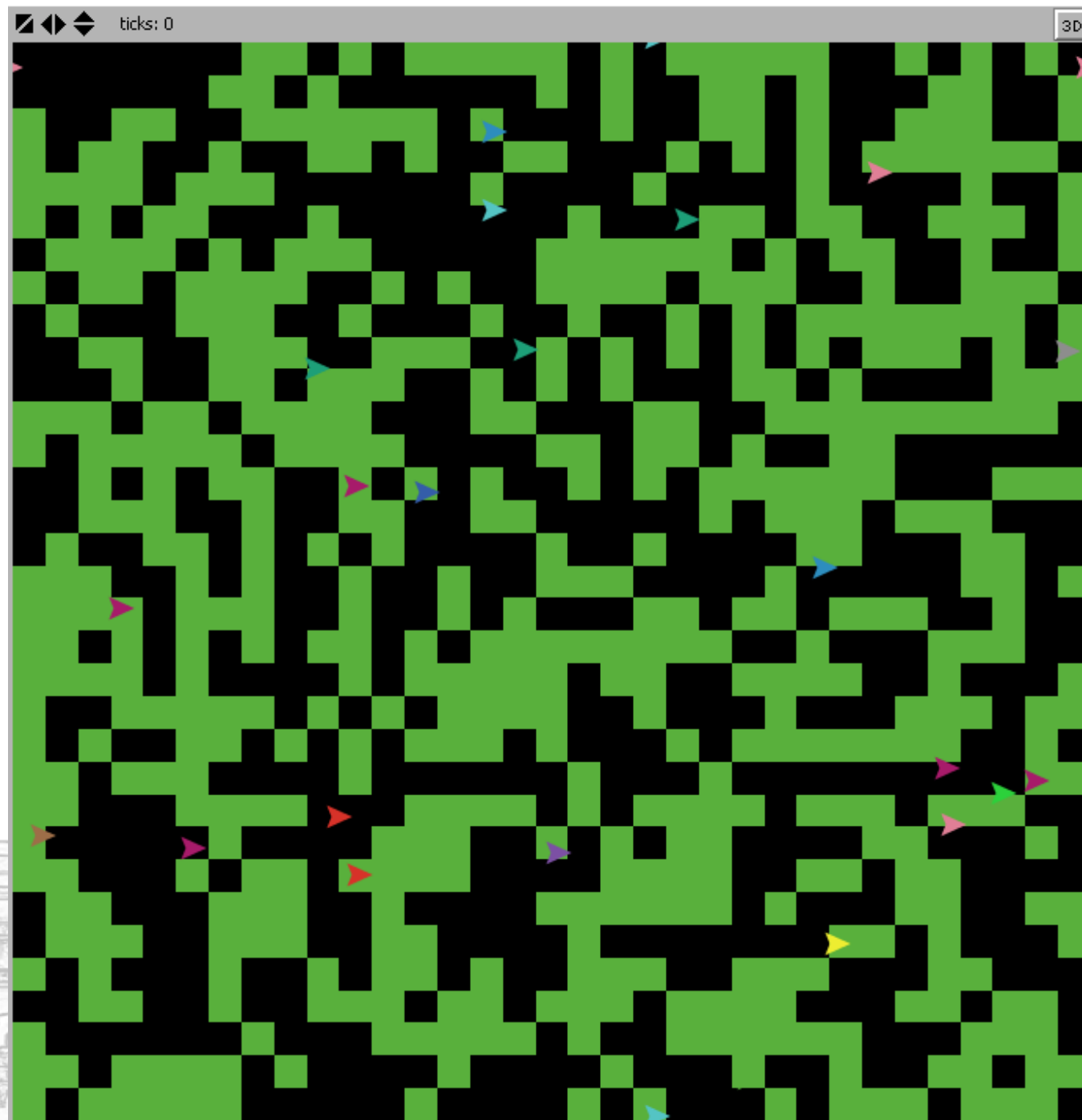
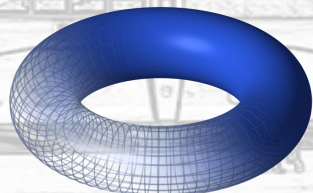
Parches son cuadrados generados aleatoriamente por el modelador.



Las tortugas son en varios colores y se pueden configurar, por defecto son flechas y pueden moverse “libremente”.



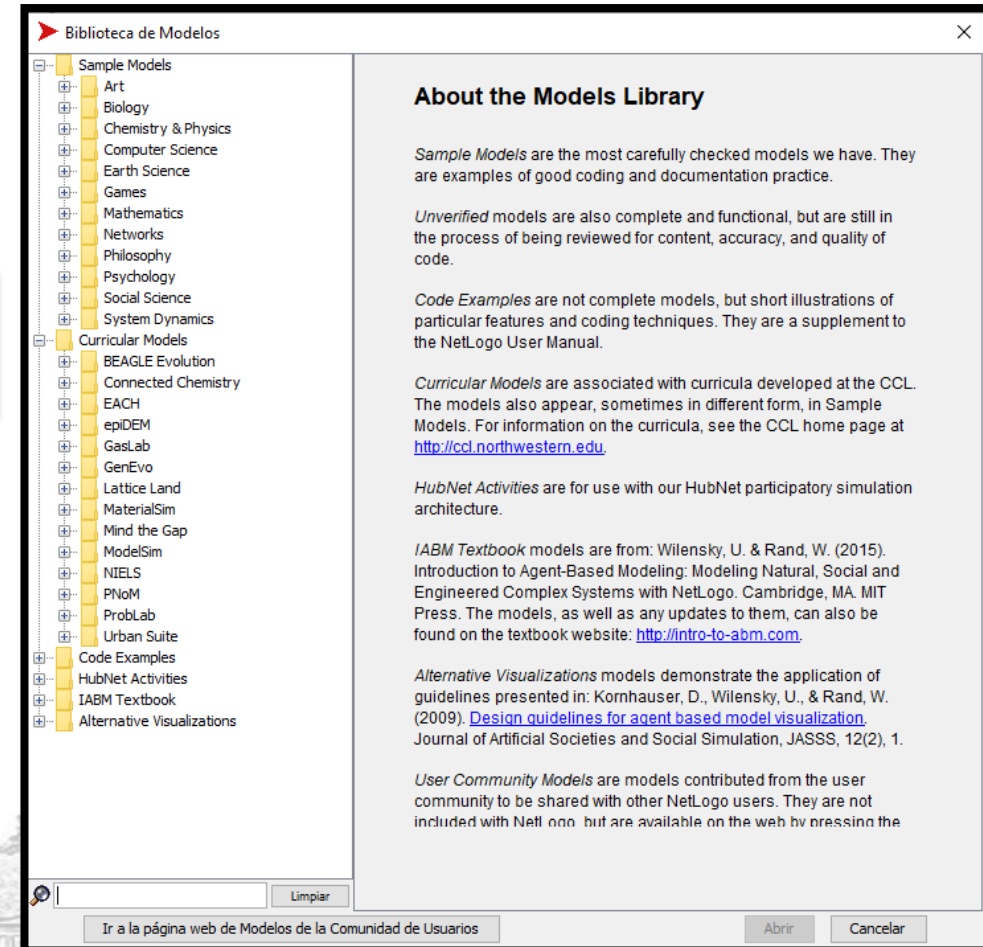
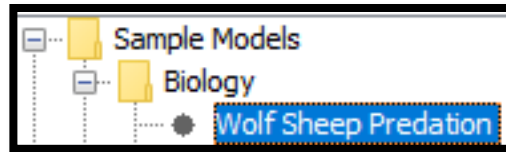
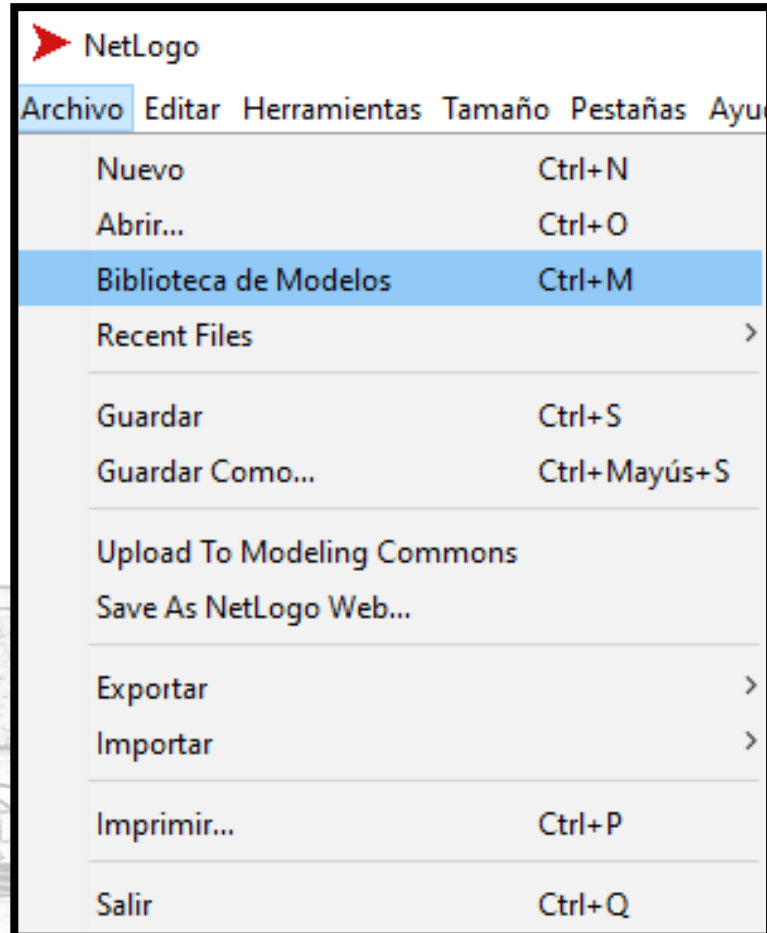
El mundo puede configurarse como un torus o un espacio limitado.



NetLogo

Datos y comandos

Abrimos la biblioteca de modelos.



wolf

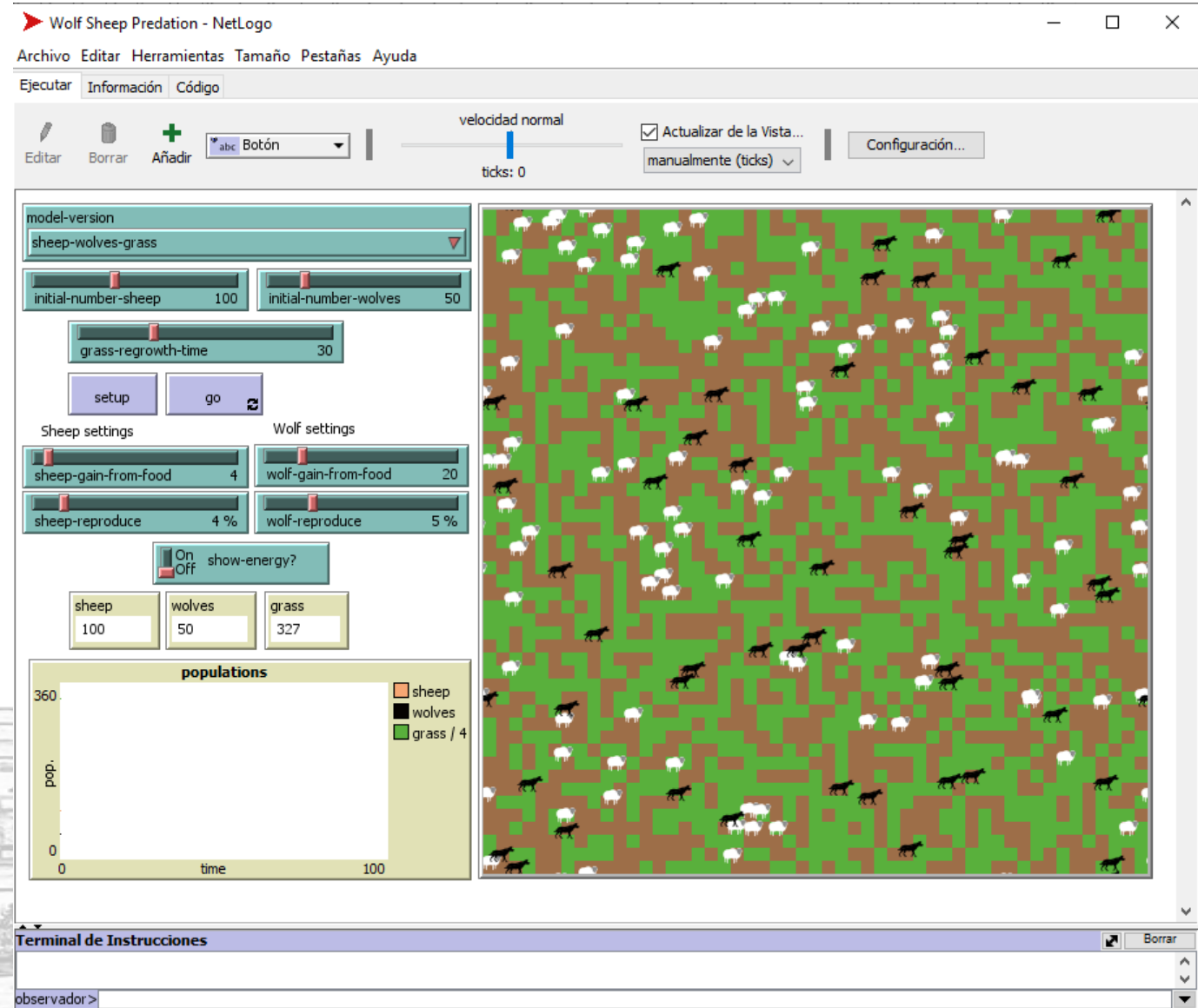
Limpiar

Un modelo sencillo de ovejas y lobos

Clic en setup

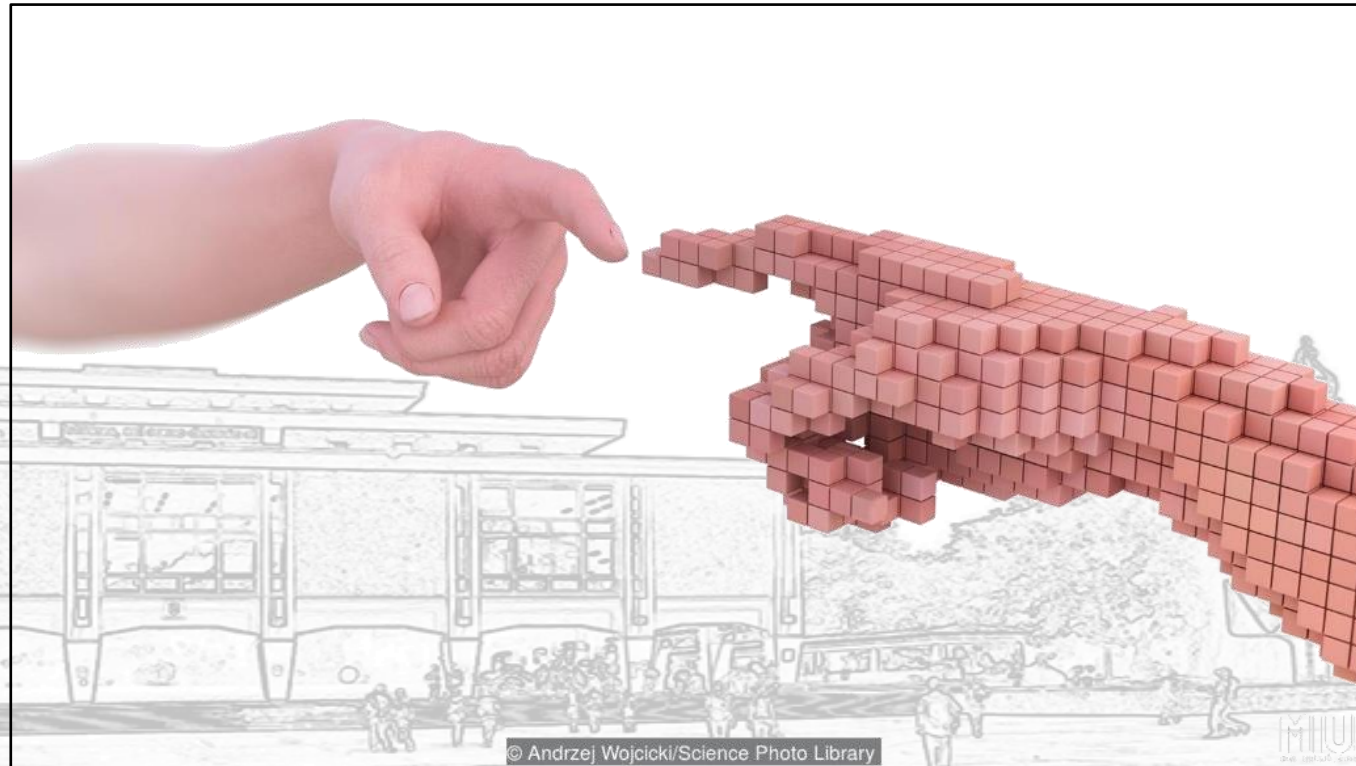
Y luego el entorno cambia.

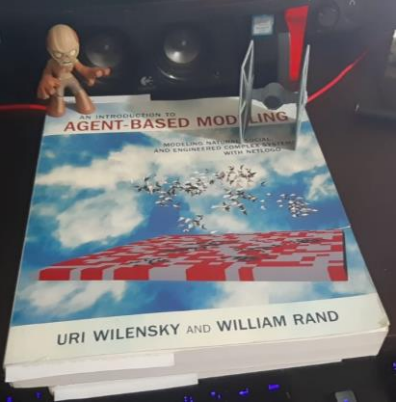
Cambiamos el modelo por su versión con pasto.



Agentes en el mundo

<https://www.washingtonpost.com/graphics/2020/world/corona-simulator/?fbclid=IwAR3Ua4dCOP8voM7k92A8eqgwK1Nt5pCYwlrTOgew-4fvleIA09uMykL2SOc>





Recomendaciones de lectura

- Wilensky, U. and Rand, W. (2015), *An Introduction to Agent-based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo*, MIT Press, Cambridge, MA.
- Railsback, S.F. and Grimm, V. (2011), *Agent-Based and Individual-Based Modeling: A Practical Introduction*, Princeton University Press, Princeton, NJ.
- Axtell, R. and Epstein, J.M. (1994), 'Agent-based Modelling: Understanding Our Creations', *The Bulletin of the Santa Fe Institute: Winter*, 28-32.
- Axtell, R. (2000), *Why Agents? On the Varied Motivations for Agent Computing in the Social Sciences*, Center on Social and Economic Dynamics (The Brookings Institute): Working Paper 17, Washington DC.
- Windrum, P., Fagiolo, G. and Moneta, A. (2007), 'Empirical Validation of Agent-based Models: Alternatives and Prospects', *Journal of Artificial Societies and Social Simulation*, 10(2): 8, Available at <http://jasss.soc.surrey.ac.uk/10/2/8.html>.
- Moss, S. (2008), 'Alternative Approaches to the Empirical Validation of Agent-based Models', *Journal of Artificial Societies and Social Simulation*, 11(1): 5, Available at <http://jasss.soc.surrey.ac.uk/11/1/5.html>.

GRACIAS

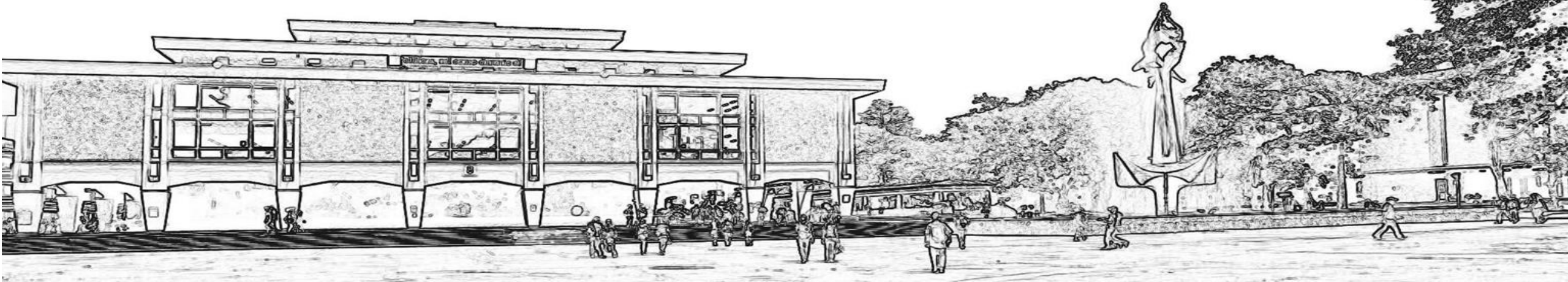
Espacio para preguntas y comentarios.





UNIVERSIDAD DE ANTIOQUIA

Facultad de Ingeniería



Muchas gracias por su atención

Julián Andrés Castillo G.
Jandres.castillo@udea.edu.co
Julian.castillo@iudigital.edu.co