

## Modelos basados en el individuo y la plataforma NetLogo

Marta Ginovart; Xavier Portell; Pol Ferrer-Closas; Mónica Blanco

### Resumen

El objetivo general de este trabajo se enmarca en el deseo de dar a conocer la metodología de la modelización basada en el individuo y sus posibilidades para ser utilizada, entre otras metodologías, en la adquisición de la competencia de modelización en docencia. Para ello se presenta: i) una descripción de los elementos que permiten identificar, describir, comunicar e implementar los modelos basados en el individuo (IBMs), y como obtener resultados de simulación; y ii) un ejemplo-muestra de un IBM de NetLogo que ilustra con detalle la manera de proceder con este tipo de modelo.

### Abstract

The overall objective of this work is the desire to make the methodology of individual-based modeling known and to show its potential for use, among other methods, in the acquisition of modeling competence in teaching. Here we present: i) a description of the elements that allow the identification, description, communication and implementation of Individual-based models (IBMs), and how to achieve simulation results; and ii) one example-sample of IBM in NetLogo illustrating how to proceed with this type of model in detail.

### Resumo

O objetivo geral deste trabalho seria parte do desejo de tornar conhecida a metodologia de modelagem baseada no indivíduo e na sua capacidade de ser usado, entre outros métodos, a aquisição de competências em modelagem de ensino. Aqui nós apresentamos: i) uma descrição dos elementos que identificar, descrever, comunicar e implementar modelos de base individual (IBMs), e como obter resultados de simulação, ii) um exemplo da IBM mostra que NetLogo ilustra detalhadamente como proceder com este tipo de modelo.

### 1. Introducción

La modelización matemática clásica fundamentada en funciones continuas, funciones derivables, ecuaciones diferenciales, métodos de optimización, ajuste de funciones, así como la modelización estadística, son las metodologías de modelización que habitualmente se estudian en el ámbito académico, en asignaturas de matemáticas tanto del bachillerato como de los primeros cursos universitarios de titulaciones científicas y tecnológicas. De hecho, la adquisición de la capacidad y habilidad para llevar a cabo de forma autónoma todos los aspectos relacionados con el proceso de modelización en un contexto determinado se incluye en la mayoría de los planes de estudios. Siguiendo las directrices del proyecto PISA (Programme for Indicators of Student Achievement) deberíamos prestar especial atención al desarrollo de competencias o habilidades tales como pensar matemáticamente, argumentar, representar y comunicar, resolver, o usar técnicas matemáticas, pero

también modelizar. De hecho, la enseñanza de las ciencias basada en la elaboración de modelos tiene gran aceptación entre la comunidad educativa (Justi, 2006; Aravena et al. 2008; Gomes-Neves y Duarte-Teodoro, 2010). La modelización científica tiene el propósito de encontrar, interpretar y validar las representaciones aproximadas de los sistemas, que se definen por conjuntos de elementos y conceptos, cuyas características y mecanismos de relación se describen con los objetos y operaciones matemáticas. En este contexto, los modelos computacionales juegan un papel clave en la expansión del horizonte de la ciencia cognitiva ya que permiten más cálculo y gran capacidad de visualización en la exploración de los sistemas que nos rodean (Gomes-Neves y Duarte-Teodoro, 2010). La llamada modelización basada en el individuo (Individual-based model, IBM) empezó a aparecer en la década de los 90 y representó un nuevo enfoque de modelización, con una filosofía y perspectiva diferente a la de la modelización clásica utilizada hasta entonces de forma habitual. Son modelos que describen a los individuos o partes que configuran un sistema como entidades autónomas y discretas, y focalizan toda su atención en caracterizar estas partes discretas mediante reglas de comportamiento, permitiendo que estas partes interactúen entre ellas y con el entorno en el que se hallan (Railsback, 2001). El concepto básico del IBM es simple y atractivo, diseñar un modelo de individuo y uno de medio ambiente, implementarlos en un código de computación, y dejar que el ordenador “cree” diversos organismos individuales y simule las interacciones de los individuos entre sí y con el medio que los rodea. El comportamiento de la población emerge de este conjunto de individuos evolucionando en el entorno simulado. Desde varias perspectivas, la construcción de un modelo de individuo (o agente) parece ser más fácil que la de un modelo de población (o sistema), ya que los individuos pueden ser controlados de maneras distintas y en un rango limitado de respuestas, mientras que la población no. Los IBMs no requieren de las mismas simplificaciones asumidas por los modelos continuos, aunque como todos los modelos, requieren asumir algunas otras simplificaciones (Grimm and Railsback, 2005). Se trata de una generación de modelos computacionales que a pesar de que no pueden ser tratados mediante resoluciones analíticas, son una buena alternativa de modelización para abordar el estudio de aspectos diversos y propios de los sistemas complejos (Grimm, 1999). Actualmente esta metodología está abriendo nuevas perspectivas para el estudio de biosistemas o conjuntos de entidades discretas con vida propia que evolucionan en un entorno cambiante. En muchos casos, los resultados de simulación que se pueden conseguir con un IBM complementan los resultados que un modelo continuo puede proporcionar. Por lo que de ninguna manera se deben percibir estas dos clases de modelización, la discreta y la continua, como excluyentes o antagónicas, sino todo lo contrario, deben ser consideradas como complementarias y necesarias para el estudio de sistemas complejos, aprovechando de cada una de ellas sus ventajas y compensando sus desventajas (Gómez-Moureló y Ginovart, 2009). Es en este contexto donde los modelos permitirán avanzar en la comprensión del funcionamiento de estos sistemas. El objetivo general de este trabajo se enmarca en el deseo de dar a conocer la metodología de la modelización basada en el individuo y sus posibilidades para que pueda ser utilizada, entre otras metodologías, en la adquisición de la competencia de modelización en docencia. Para ello se presenta: i) una descripción de los elementos que permiten identificar, describir, comunicar, e implementar IBMs, y como obtener resultados de simulación, ii) un ejemplo-muestra

de un IBM en NetLogo que ilustra con detalle la manera de proceder con este tipo de modelo.

## 2. Los modelos basados en el individuo

### 2.1. Modelización y docencia

La mayoría de los primeros ejemplos de utilización de IBMs se encuentran en el área de ecología, en campos específicos tan distintos como son el estudio de poblaciones de insectos, peces, mamíferos, aves, plantas, arboles, entre otras opciones. El libro de Grimm and Railsback (2005) recoge y analiza algunos de estos ejemplos, proporcionando referencias complementarias o adicionales. En los últimos años, un gran número de revistas científicas especializadas, y de perfiles distintos, están publicando cada vez más trabajos que tienen como denominador común el uso de un IBM, siendo las áreas de aplicación cada vez más diversas. En estos momentos podemos afirmar que la modelización basada en el individuo ha demostrado ser una herramienta válida para describir y tratar con sistemas complejos formados por entidades autónomas. Este tipo de modelo computacional facilita la experimentación y manipulación "virtuales" de los sistemas simulados. La ejecución reiterada y automática de los códigos de programación o simuladores ofrece muchas posibilidades para la replicación controlada de evoluciones temporales de los sistemas estudiados. Como si de un conjunto de "experimentos" se tratara, los resultados de simulación que se obtienen permiten el análisis y discusión de variables involucradas en los diferentes escenarios ensayados.

Teniendo en consideración que desde un punto de vista conceptual los modelos basados en el individuo son sencillos de comprender y no requieren de teorías matemáticas avanzadas, estamos convencidos que éstos pueden ser útiles desde una visión educativa para comprender y investigar cuestiones diversas relacionadas con los sistemas formados por seres vivos, así como para trabajar la competencia de la modelización en el ámbito académico desde diferentes niveles. Es decir, los IBMs pueden convertirse en un instrumento útil para exponer, profundizar y presentar, con una perspectiva distinta a la que la modelización continua tiene, conceptos y comportamientos relacionados con sistemas vivos. Su uso también conlleva una excelente oportunidad para desarrollar y practicar todas las etapas implicadas en un proceso de modelización. Algunos conceptos como competencia, cooperación, adaptación, diversidad, individualidad, variabilidad pueden ser más fáciles de exponer y apreciar mediante este tipo de modelización basada en el individuo, que utilizando por ejemplo ecuaciones diferenciales o modelos estadísticos. Las características particulares de los IBMs pueden ser exploradas y aprovechadas en actividades docentes, complementando los contenidos de modelización más tradicionales presentes en la mayoría de currículos académicos. Como los IBMs se construyen utilizando la matemática de los acontecimientos discretos y no mediante tasas o velocidades, constituyen una herramienta especialmente interesante para ser utilizada con imaginación en cuestiones ligadas a la realidad, la cual se caracteriza muy a menudo por la presencia de acontecimientos no continuos (Jacobson and Wilensky, 2006).

### 2.2. El protocolo ODD para la descripción de IBMs

Los modelos analíticos son fáciles de comunicar porque se formulan con el lenguaje matemático, y nos hemos acostumbrados a su reiterada presencia durante toda la formación académica. Su descripción suele ser completa, clara y accesible al lector. En cambio, la mayoría de las descripciones publicadas de IBMs que uno puede consultar, son generalmente difíciles de leer, incompletas, ambiguas, y por lo tanto menos accesibles (Grimm et al., 2006). Hay dos cuestiones principales y relacionadas entre sí que han condicionado hasta ahora la descripción de IBMs: a) No ha habido un protocolo estándar para su descripción hasta hace relativamente bien poco, por lo que cada IBM elaborado es explicado según los criterios y preferencias de sus autores, que generalmente pertenecen a grupos de investigación de áreas distintas (ecólogos, matemáticos, ingenieros, botánicos, físicos, químicos, microbiólogos, veterinarios,...); b) Se describen a menudo verbalmente sin una indicación clara de las ecuaciones, reglas y esquemas que el modelo utiliza, y el hecho de que la inclusión del código completo de computación no sea admisible en la mayoría de las publicaciones, dificulta el acceso a la información completa del IBM. Por ejemplo, cuando consultamos un modelo analítico esperamos encontrar un conjunto de ecuaciones y la definición de las variables que utilizan, junto con una tabla de valores de los parámetros. Cuando se consulta un IBM publicado en una revista científica, no se identifica ningún orden preestablecido o consensuado en la transmisión de la información que aglutina el modelo computacional. Muchas veces la lectura detallada y exhaustiva de todo el que el simulador puede controlar y reproducir es excesiva e ineficiente para conocer información relevante del modelo, su estructura general o sus rasgos fundamentales. A menudo se mezclan consideraciones generales, con descripciones de procesos, con justificaciones para la formulación empleada o hipótesis previas. Una solución, si bien parcial, a estos problemas pasaría por consensuar y unificar dentro de la comunidad científica la manera de presentar y comunicar los IBMs. Tomado en consideración este contexto, se formó un grupo de trabajo formado por 27 modelizadores que trabajaban y tenían experiencia con esta metodología, con el propósito de preparar y redactar un protocolo de descripción a seguir en la presentación de IBMs (Grimm et al. 2006). De esa experiencia surgió el protocolo llamado ODD, el cual recibe su nombre por los bloques que utiliza para agrupar la información: O de "Overview", D de "Design concepts", y D de "Details". Cada uno de estos tres bloques principales está constituido por una serie de elementos que se presentan en la Tabla 1.

|                      |  |
|----------------------|--|
| O:"Overview"         | Propósito  |
|                      | Variables de estado y escalas  |
|                      | Perspectiva del proceso y programación   |
| D: "Design concepts" | Conceptos de diseño o conceptos generales tratados en el campo de los sistemas adaptativos complejos (Emergencia, Adaptación, Finalidad, Predicción, Sensibilidad, Interacción, Estocasticidad, Colectivos, Observación) |
| D:"Details"          | Inicialización   |
|                      | Entrada  |
|                      | Sub-modelos  |

**Tabla. 1: Los distintos elementos que considera el protocolo ODD para la descripción de los IBMs. Adaptada de Grimm et al. (2006).**

### 2.3. La implementación de IBMs en códigos de computación

Después de trabajar el modelo conceptual y llegar a una formulación de éste dentro del marco de la metodología IBM, el diseño obtenido ha de ser transformado en un algoritmo numérico, en un código de computación. Con la codificación utilizada para su implementación se consigue un programa o simulador preparado para ser ejecutado. Las etapas que hay que cubrir para desarrollar un modelo-simulador de este tipo son las propias de todo proceso de modelización con, evidentemente, algún matiz propio del contexto de aplicación: i) lenguaje de programación, código de computación, y verificación, ii) parametrización, calibración, y análisis de sensibilidad, iii) valoración de la adecuación o idoneidad del modelo-simulador, y solución de problemas propuestos o respuestas a preguntas formuladas. Sin embargo, el desarrollo del código de computación puede ser un posible obstáculo para la utilización de IBMs en muchos proyectos docentes. El no poder disponer del simulador correspondiente al modelo formulado es, evidentemente, una limitación indiscutible para que estos IBMs puedan ser utilizados en actividades académicas, así como para que estén presentes en materiales didácticos para la clase. Una formación limitada en el ámbito de la informática avanzada de docentes (lo mismo sucede en el ámbito de la investigación) y la imposibilidad, del todo razonable, de no poder conocer los últimos avances en sistemas operativos y lenguajes de programación, son elementos que no se pueden ignorar u obviar. En cualquier caso, la formación informática-computacional no debe ser el factor limitante para el uso de los IBMs.

Para minimizar este inconveniente y, al mismo tiempo, unificar y simplificar el proceso de construcción de este tipo de modelos, se pueden utilizar entornos informáticos de modelización que se han ido desarrollando en los últimos años en centros de investigación y universidades (Railsback et al., 2006). Una alternativa a utilizar estas plataformas ya preparadas específicamente para tratar con la modelización discreta, sería la de programar los propios códigos de computación haciendo uso de lenguajes como Fortran, C, Java o C++, entre otros, lo cual conllevaría naturalmente tanto ventajas como desventajas. No obstante, estamos convencidos de que la utilización de plataformas ya preparadas para la implementación de IBMs facilita la tarea de programación y ejecución que requiere todo modelo computacional, a la vez que permite su divulgación a un público más amplio y heterogéneo. El crecimiento de trabajos con IBMs se debe primeramente a la habilidad de estos modelos para resolver problemas que los modelos clásicos no pueden resolver, gracias a la evolución de la teoría y las estrategias para interpretar la ciencia mediante IBMs, pero también al incremento de las plataformas o entornos informáticos específicos para realizar estas simulaciones (DeAngelis et al, 2005). Dentro de este software de modelización encontramos por ejemplo a Swarm (Minar et al.1996), Repast (North et al. 2006), MASON (Luke et al. 2005), StarLogo (Colella et al. 2001) y NetLogo (Wilensky, 1999). Creemos que para el ámbito docente, lo más adecuado sería alguna de las opciones que la familia de plataformas Logo ofrece. Tanto StarLogo como NetLogo tienen características que las hacen muy atractivas para poderlas utilizar en actividades académicas programadas. Son plataformas que han ido evolucionando de forma distinta, pero que inicialmente surgieron del lenguaje de programación Logo, creado con el objetivo de ofrecer a los estudiantes un entorno donde construir y aprender sobre modelos discretos simples.

En la web <http://education.mit.edu/starlogo/> se puede encontrar información sobre StarLogo y de cómo iniciarse en su uso. Las raíces de NetLogo también están en el ámbito educativo, y eso ha significado una manera especial de hacer en el proceso de extensión y avance desde su concepción hasta la actualidad. La última versión de NetLogo es muy potente e innovadora, lo que la está convirtiendo hoy en día en una plataforma cada vez más citada y utilizada en el tratamiento de IBMs. Refleja claramente su herencia como herramienta útil para el estudio y aprendizaje, ya que ese fue el primer objetivo por el cual se creó, pero su avance ha sido tan espectacular, que actualmente puede ser también utilizada para trabajar incluso con proyectos de investigación que no requieran de IBMs excesivamente sofisticados. El lenguaje contiene muchos controles y estructuras que reducen en gran medida el esfuerzo de la programación, pero no tiene todas las capacidades que un lenguaje de programación estándar posee.

### 3. La plataforma NetLogo

NetLogo (<http://ccl.northwestern.edu/netlogo/>) fue creada por Uri Wilensky en 1999 y está en continuo desarrollo en el Center for Connected Learning and Computer-Based Modeling de Estados Unidos (<http://ccl.northwestern.edu/>). Es un entorno que permite la implementación de IBMs y la simulación del comportamiento de sistemas complejos a lo largo del tiempo y en el espacio (bidimensional o tridimensional). Fue diseñado para tratar con agentes o individuos móviles actuando al mismo tiempo en un espacio y con un comportamiento dominado por las interacciones locales durante un periodo de tiempo. El lenguaje de programación de NetLogo es más sencillo que los lenguajes utilizados por este tipo de programación (Objective-C o Java), con una animación automática ligada al modelo y controles gráficos (opcionales). No se ha establecido un número máximo de individuos o reglas que pueda tener el modelo, pero las simulaciones son más o menos rápidas en función del número de individuos del sistema, reglas fijadas para cada individuo y entorno, y características del ordenador con el que se trabaje (Gilbert y Bankes, 2002). Los implementadores de modelos pueden dar instrucciones a cientos o miles de individuos para que todos ellos operen de manera independiente entre sí y con el entorno. Esto hace posible explorar la relación entre el comportamiento a bajo nivel de los individuos y los patrones macroscópicos que surgen a partir de la interacción de muchos individuos entre sí.

El programa incluye una galería de modelos con una breve descripción de sus propósitos, que pueden ser ejecutados, y que permiten modificar valores de parámetros implicados en las simulaciones ("NetLogo Models Library", <http://ccl.northwestern.edu/netlogo/models/>). Son los modelos más cuidadosamente controlados de que dispone el servicio vía web, y están destinados a ser ejemplos de buenas prácticas de codificación y documentación. Los modelos marcados como "no verificado" aún están en proceso de ser revisados por el contenido, exactitud y calidad del código. De hecho existe una Comunidad de Usuarios de NetLogo importante, y estos modelos son aportados por ésta para poder ser compartidos con otros usuarios de NetLogo. Los modelos que se pueden encontrar en esta galería pertenecen a ámbitos muy diversos: arte, ciencias de la computación, biología, química, física, ciencias del suelo, juegos, matemáticas (fractales, probabilidad, sistemas dinámicos), ciencias sociales, economía, y psicología social. Algunos

modelos ilustrativos de IBMs del ámbito de la biología son: i) “SIDA”, la propagación del virus de inmunodeficiencia humana (VIH) a través de la transmisión sexual, en una pequeña población de humanos aislados para examinar los efectos emergentes de cuatro aspectos distintos de la conducta sexual; ii) “Ants Line”, el comportamiento de las hormigas siguiendo a un líder hacia una fuente de alimento, o “Ants”, una colonia de hormigas y su comportamiento cuando encuentran un resto de comida; iii) “Cooperation”, donde los agentes (vacas) compiten por los recursos naturales (hierba), y donde se incluye dos tipos de vacas en relación a como utilizan la hierba, etiquetadas como codiciosas y cooperantes, para mostrar cómo estas dos estrategias diferentes hace que el sistema evolucione de forma distinta a lo largo del tiempo; iv) “Fireflies”, una población de luciérnagas que sincronizan su parpadeo utilizando sólo las interacciones entre las luciérnagas individuales; v) “Rabbits Grass Weeds” que explora un ecosistema sencillo compuesto por conejos, hierba y maleza, la hierba y las malezas pueden crecer a ritmos diferentes y dar a los conejos diferentes cantidades de energía, para explorar las ventajas competitivas de estas variables, vi) “Algae”, un ecosistema acuático que consiste en una columna de agua que contiene algas, luz y nutrientes, con más luz en la parte superior, pero más comida en la parte inferior, por lo que las algas se mueven arriba y abajo para equilibrar sus necesidades.

Existe abundante documentación y tutoriales sobre NetLogo en sus páginas web. En <http://sites.google.com/site/manualnetlogo/> se puede encontrar un manual de NetLogo en español que permite familiarizarte con este entorno a través de pequeños programas-ejemplo. NetLogo permite a los usuarios abrir simulaciones y “jugar” con ellas, explorando su comportamiento bajo una serie de condiciones que se pueden manipular. Asimismo, permite al usuario la creación de sus propios modelos. NetLogo es lo suficientemente sencillo como para que estudiantes y profesores puedan ejecutar las simulaciones o incluso construir las suyas propias si el contexto académico lo permite. Existen dos maneras de interactuar con NetLogo, descargando e instalando el programa (permite simular y editar modelos, así como la creación de nuevos modelos), o ejecutando un applet desde una página web (únicamente permite la ejecución de modelos). Esto es especialmente deseable en el ámbito docente puesto que el estudiante puede experimentar con el simulador sin necesidad de instalar el programa, requiriendo únicamente conexión a internet. Otra característica atractiva de NetLogo es la existencia de una herramienta de simulación participativa llamada Hubnet. Hubnet permite, mediante dispositivos conectados en red, que cada estudiante controle un individuo o agente en una simulación. Finalmente, hay que destacar la posibilidad de crear nuevos procedimientos en Java (u en otros lenguajes) denominados extensiones, que permiten utilizar otros programas como por ejemplo los sistemas de información geográfica, el programa estadístico R o de resolución de sistemas de ecuaciones diferenciales, en las simulaciones de Netlogo.

Una prueba indiscutible de las posibilidades del uso de NetLogo para la creación, implementación y uso de IBMs orientados al aprendizaje de este tipo de modelización discreta en el ámbito docente es la existencia de la web <http://www.railsback-grimm-abm-book.com/index.html>. Desde esta web se puede acceder al material de un libro en preparación titulado “A Course in Individual- and Agent-based Modeling - Scientific Modeling with NetLogo”. El libro está pensado en

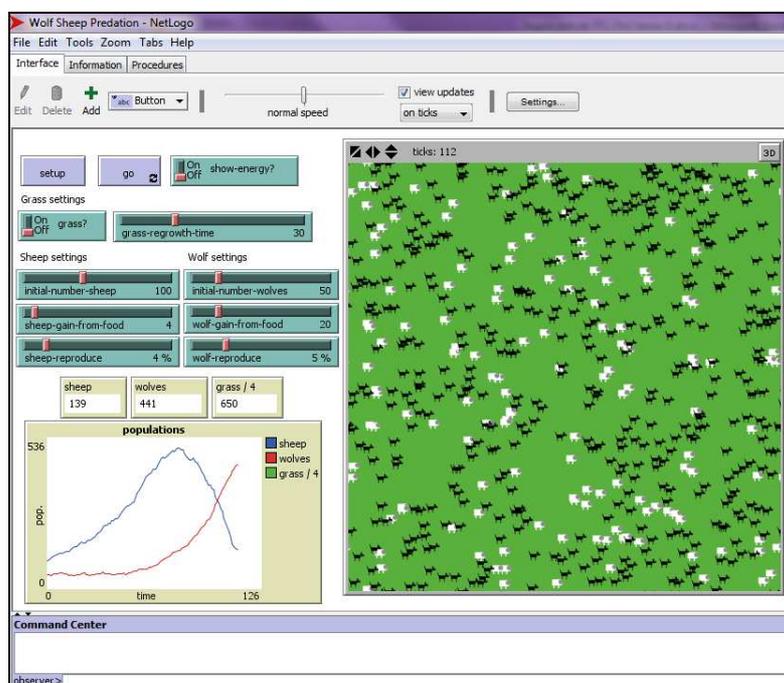
principio como un libro de texto de nivel intermedio sobre las aplicaciones científicas de modelización basada en agentes (utilizando los términos "de base individual" y "basado en agentes" como sinónimos). Este material docente utiliza el software NetLogo como la plataforma para construir y analizar modelos. Después de la publicación de este libro que se prevé durante el año 2011, los autores quieren convertir ese sitio web en un foro de apoyo a los instructores y otros usuarios. Al mismo tiempo, el creador Uri Wilensky del programa NetLogo y William Rand también han preparado un libro de texto sobre modelización basada en agentes o individuos con NetLogo (Wilensky y Rand, en prensa). Textos de este tipo permitirán llenar la necesidad actual de textos introductorios para académicos y científicos, que puedan ser utilizados en cursos universitarios o de nivel superior a la educación secundaria, así como para instruir a potenciales usuarios autodidactas. Un curso sobre IBMs debe principalmente centrarse en las habilidades prácticas necesarias tanto para diseñar modelos como para implementarlos y ejecutarlos en entornos computacionales, con el fin de poder analizar y discutir los resultados de simulación.

### 3.1. ¿Cómo empezar a utilizar NetLogo?

Los dos elementos o agentes principales que usa NetLogo son "turtles" y "patches". Los primeros son los agentes que tienen capacidad de movimiento en el "mundo" (espacio donde éstos se desarrollan y evolucionan), interactúan entre sí y con el medio, y están registrados por un identificador que es único para cada "turtle", (la variable numérica "who"). Los segundos son las unidades básicas que no tienen movimiento, son cada una de las porciones cuadradas en que se subdivide el "mundo", identificando cada "patch" por las coordenadas cartesianas de su punto central. Estos agentes ("turtles" y "patches") tienen una serie de variables que les son propias, aunque el programador puede definir variables específicas, las cuales se controlarán con las sentencias *turtles-own [nombre]* o bien *patches-own [nombre]*. Por ejemplo, *turtles-own [energía]*, indica que sobre cada individuo controlamos la variable *energía* que variará a lo largo de la simulación. Se pueden definir distintos tipos de "turtles" mediante la instancia *breed [nombre\_plural nombre\_singular]*. Por ejemplo, *breed [ovejas oveja]* indica que hay un tipo especial de "turtles" que nosotros identificamos como *oveja*, cuando nos referimos a un individuo específico, o *ovejas*, cuando nos referimos a más de una *oveja*. Otro elemento importante son las "ticks" o unidades temporales, útiles para controlar la duración de las simulaciones.

Con el fin de familiarizarse con el programa, y tener una visión general de su funcionamiento, lo más conveniente es realizar simulaciones con los ejemplos de IBMs disponibles en la galería de modelos de NetLogo. En abrir uno de estos modelos, encontramos la ventana "Interface" (Fig. 1). Las partes principales de la ventana "Interface" son: i) barra de control de la velocidad de simulación, ii) botón "setup", iii) botón "go", iv) "sliders". El botón "setup" se utiliza para iniciar o reiniciar el modelo, pero no para ejecutarlo sino que, únicamente muestra el estado inicial de la simulación. Por lo tanto, si se realiza cualquier modificación de los parámetros en pantalla, para que tenga validez, primero hará falta activarlo de nuevo. Seguidamente habrá que apretar el botón "go", el cual hará que empiece la simulación, la cual se puede seguir mediante los gráficos que contiene la ventana (Fig. 1). Es importante hacer notar que puede haber dos tipos de botones de "go",

uno que al apretarlo empieza a funcionar y no se detiene hasta que lo vuelves a apretar ("go forever"), y otro que se tiene que ir apretando conforme van pasando los pasos de programa ("go once"). También encontraremos una barra que muestra la velocidad de la simulación, que puede ser ajustada a voluntad por el usuario. Se podría decir que la ventana "Interface" es la parte más atractiva ya que en ella se muestra cómo es y qué sucede a lo largo de la simulación. Además de los elementos comentados anteriormente se pueden añadir algunas opciones más: i) el "Button", que permite ejecutar la orden relacionada con él (con dos tipos de botones, los que actúan una vez y otros que al apretarlos empiezan a ejecutar las órdenes internas y no se detienen hasta que no se vuelven a apretar, se trataría del "setup" y el "go" comentados en el anterior apartado); ii) el "Slider" permite la creación de rangos de valores para los parámetros o variables del modelo, con el fin de crear situaciones iniciales diferentes o sistemas iniciales con características distintas, con lo cual se podrán obtener diferentes evoluciones en diversas simulaciones (por ejemplo, variar el número inicial de individuos, o las probabilidades de ocurrencia de sucesos específicos como reproducción o muerte); iii) el "Switch" permite activar o desactivar una acción (por ejemplo sería si queremos conocer a lo largo de la simulación la cantidad de energía o reservas que tiene cada uno de los individuos); (iv) el "Chooser" hace posible la elección de un dato o variable para controlar (sería el caso que se presentara un modelo con diferentes variables, de las cuales, sin embargo, únicamente se quiere que actúen una en una, es decir, que habría que hacer una simulación para cada variable disponible; v) el "Monitor" que permite calcular el valor de cualquier expresión del modelo, se trate de "turtles" o de "patches", a lo largo de la simulación; vi) el "Plot" permite la creación y generación de gráficos con diversas variables, a lo largo del tiempo de la simulación (Fig. 1).



Fig

Figura 1: Vista de las partes básicas de la ventana "Interface" del IBM "Wolf Sheep Predation" de la galería de modelos implementados en la plataforma NetLogo.

Además de la ventana "Interface" de NetLogo, existen las ventanas "Information" y "Procedures". La ventana "Information" contiene la documentación correspondiente al modelo que se encuentra implementado desde qué es y cómo se utiliza, hasta las referencias que se pueden consultar en el caso de los ejemplos-muestra. En la creación de un nuevo modelo, el programa marca unos apartados a seguir para redactar el texto en esta ventana ("What is it?, How it Works?, How to use it?, Things to notice?, Things to try?, Extending the model?, NetLogo features, Related models, Credits and references), siendo recomendable seguir las indicaciones del programa con el fin de mantener un formato común y más fácil de gestionar. En la ventana "Procedures" encontramos el verdadero corazón del modelo, el conjunto de procedimientos que se ejecutan en la simulación (Fig. 2). En ella se encuentran implementadas las acciones sobre el entorno y las reglas de comportamiento de los individuos que han sido conceptualmente diseñadas en el modelo. Todos los elementos de la ventana "Interface" funcionarán gracias a su programación en la ventana de "Procedures", ya que por sí solos no podrían realizar ninguna acción.

```

globals [grass] ;; keep track of how much grass there is
;; Sheep and wolves are both breeds of turtle.
breed [sheep a-sheep] ;; sheep is its own plural, so we use "a-sheep" as the singular.
breed [wolves wolf]
turtles-own [energy] ;; both wolves and sheep have energy
patches-own [countdown]

to setup
  clear-all
  ask patches [ set pcolor green ]
  ;; check GRASS? switch.
  ;; if it is true, then grass grows and the sheep eat it
  ;; if it false, then the sheep don't need to eat
  if grass? [
    ask patches [
      set countdown random grass-regrowth-time ;; initialize grass grow clocks randomly
      set pcolor one-of [green brown]
    ]
  ]
  set-default-shape sheep "sheep"
  create-sheep initial-number-sheep ;; create the sheep, then initialize their variables
  [

```

Figura 2: Vista de la ventana "Procedures" del IBM "Wolf Sheep Predation" de la galería de modelos implementados en la plataforma NetLogo.

### 3.2. Algunos elementos y procedimientos básicos en NetLogo

En NetLogo encontramos dos tipos de agentes más, aparte de de los "turtles" y los "patches" expuestos anteriormente, los "links" y el "observer" que se utilizan en programación adelantada. Los "links" son agentes que conectan dos "turtles", y el "observer" no tiene una posición fija en el espacio, se encuentra fuera de él, viéndolo todo y pudiendo interactuar con él creando y destruyendo agentes, asignando

propiedades a los agentes, etc. El código de programación se genera construyendo diferentes procedimientos. Cada uno de los procedimientos tiene un nombre precedido de la palabra "to", seguido por una serie de sentencias ejecutables, y finalizando con la expresión "end". Dentro del fragmento ejecutable podemos encontrar órdenes e informadores. Las primeras son acciones que un individuo (agente) puede llevar a cabo, mientras que los segundos hacen los cálculos y después informan del resultado (Fig. 2). Así, pues, la mayoría de las órdenes son verbos ("create", "die", "jump", "inspect", "clear"), mientras que los informadores, normalmente, son nombres o frases ("random", "distance", "set-default-shape", "set-current-plot"). NetLogo lleva también por defecto una serie de órdenes y de informadores que se llaman primitivas, pero que no siempre pueden ser suficientes para implementar un IBM, por lo que se permite crear nuevos "procedimientos a realizar". En el apartado "Help" del NetLogo se encuentra disponible un diccionario que recoge todas las primitivas del lenguaje, además de una guía para el usuario. Una vez se ha descrito un procedimiento, ya se puede utilizar tantas veces como sea necesario. Esta fuera del alcance de este trabajo presentar más elementos y procedimientos necesarios para programar un IBM con NetLogo, no obstante, estas indicaciones básicas permiten vislumbrar la manera de trabajar en esta plataforma. En el caso de que el lector quisiera iniciarse en la programación de un nuevo IBM recomendamos consultar la documentación pertinente que se encuentra en la web de NetLogo y realizar los tutoriales propuestos para ello.

#### 4. Un ejemplo de IBM en NetLogo: "Cooperation"

El modelo "Cooperation" (Wilensky, 1997) es uno de los ejemplos de IBM que contiene la galería "NetLogo Models Library" en el ámbito de la biología (<http://ccl.northwestern.edu/netlogo/models/Cooperation>). En él, de forma muy esquemática, los individuos compiten por los recursos naturales, los más eficientes en conseguir estos recursos se reproducirán más frecuentemente, y por lo tanto, tendrán un éxito evolutivo superior. El modelo incluye dos tipos de individuos, los "greedy" o "egoístas" y los "cooperative" o "solidarios", cada uno de los cuales muestra un comportamiento diferente a la hora de conseguir recursos naturales. Con este modelo se puede analizar cada una de las dos estrategias alimenticias y comprender sus consecuencias cuándo compiten entre ellas dentro de una población, en un determinado espacio, y a lo largo del tiempo. La información proporcionada por la web respecto al modelo implementado responde de forma muy concisa a las cuestiones ¿qué es?, ¿cómo trabaja?, ¿cómo utilizarlo?, aspectos a considerar, cosas para hacer, extensión del modelo, características de NetLogo, modelos relacionados, y referencias. No obstante, además de conocer cómo manipular NetLogo y manejar los simuladores implementados en la web, para trabajar con este tipo de modelización en el marco educativo y poder preparar actividades docentes para la consecución de la competencia en modelización, es importante enmarcar conceptualmente y de manera adecuada el modelo.

Es conveniente que su descripción y comunicación se ciña al protocolo ODD recientemente aceptado por la comunidad científica (Grimm et al. 2006). Hay que tener en cuenta que las referencias que se obtienen directamente de la web para este modelo datan de años anteriores. Fue creado como parte del proyecto titulado "Connected mathematics: making sense of complex phenomena through building

object-based parallel models”, y fue convertido a NetLogo como parte de los proyectos “Participatory simulations: network-based design for systems learning in classrooms” y “Integrated simulation and modeling environment” en el año 2001.

#### 4.1. Descripción del IBM “Cooperation” según el protocolo ODD

En este apartado presentaremos el modelo “Cooperation” siguiendo el protocolo ODD de la Tabla 1, de acuerdo con las indicaciones establecidas por Grimm et al. (2006) para la comunicación de los IBMs.

##### 4.1.1. O: “Overview”

**Propósito.** El propósito de este modelo es investigar a nivel cualitativo cómo afecta a la supervivencia de una especie animal su comportamiento con el entorno medioambiental o hábitat; y más concretamente, la competencia intraespecífica referida a la utilización que se hace de los recursos alimenticios.

En el sistema modelizado encontraremos dos tipos diferentes de actitudes individuales en la población versus el recurso alimenticio, una correspondiente a lo que sería un individuo “egoísta”, que utilizara todo el alimento disponible, y otra a la de un individuo “solidario”, que no acabará con todo lo que encuentre a su alcance. Es un modelo esencialmente teórico y abstracto para ensayar, evaluar y analizar algunos aspectos de esta competencia intraespecífica.

**Descripción de variables y escalas.** El modelo está compuesto por dos entidades: los individuos o animales (vacas en este caso) y la hierba sobre la que estos individuos viven. La agrupación de individuos con la misma estrategia para utilizar los recursos disponibles da lugar a una comunidad, y la suma de las dos comunidades a la población total. Para el ambiente, la agrupación de todas las unidades de hierba presentes en un espacio concreto, da lugar a una unidad de ambiente, y la totalidad de unidades de ambiente, al espacio.

Para cada individuo se controlaran las siguientes variables: (1) estado energético, con esta energía será capaz de moverse para conseguir alimento y reproducirse, pero sin ella morirá; (2) localización o unidad de ambiente que está ocupando en cada paso de tiempo (varios individuos pueden ocupar una misma unidad de ambiente, sin limitación en su capacidad máxima); (3) estrategia o tipo de comportamiento en relación al uso del recurso energético que se identifica con la ingesta de hierba, pudiendo ser “egoísta” o “solidario”. Es un modelo explícitamente espacial, la heterogeneidad del dominio se debe a la distribución de hierba que se consume y que rebrota.

El espacio será la totalidad de unidades de ambiente, en este caso serán 441 parcelas, distribuidas uniformemente dentro de un marco de 21 x 21. Cada unidad ambiente o parcela viene caracterizada por la cantidad de hierba que contiene (altura de ésta). El modelo no especifica las unidades para los valores de los parámetros implicados, se utilizan valores adimensionales sin correspondencia con unidades reales. Cada unidad temporal equivaldrá a un “tick” o paso de programa. La distribución de la hierba en el dominio se visualiza representando las parcelas del dominio con una gama de colores verdosos, donde el verde vivo significa que se encuentra al 100% de capacidad, y el negro la ausencia total de hierba (Fig 3). La Tabla 2 muestra los valores que pueden tomar los parámetros y variables implicados en el modelo. Con este diseño únicamente se aborda el aspecto cualitativo de los

resultados de las simulaciones, que están representadas por las evoluciones temporales de los tamaños de las dos comunidades (número de individuos “solidarios” y de “egoístas”) así como su ocupación o distribución en el dominio cuadrado, y por la cantidad de hierba en el espacio.

**Perspectiva general y programación:** En el modelo se fijan una serie de reglas de comportamiento para los animales y unas acciones sobre el entorno. La consideración de tiempo es discreta, con los “ticks o pasos de programa. Las reglas de comportamiento para los individuos que se realizan en cada paso de programa son: i) movimiento aleatorio en un radio determinado con un gasto energético que se puede asociar al metabolismo; ii) ingesta de hierba en su parcela que le permitirá adquirir energía; los animales “solidarios” comprobarán antes de alimentarse que exista un nivel mínimo de unidades de hierba disponibles  $i$ , si no es así, no se alimentarán, mientras que los animales “egoístas” se alimentarán siempre que haya hierba disponible; iii) reproducción si su energía es suficiente, con la aparición de un descendiente en su misma posición y que asumirá su misma estrategia alimentaria, también con un coste energético; iv) muerte si su energía es nula. Las acciones sobre la hierba son el crecimiento de ésta después de que haya sido ingerida por los individuos. En cada paso de tiempo, y aleatoriamente, la hierba puede o no crecer dependiendo de su altura, mayor probabilidad para una altura de hierba no inferior a un valor específico, y menor probabilidad por debajo de esa altura. El modelo no considera la edad, por lo que no contempla ni la madurez reproductiva ni el envejecimiento de los individuos. Tampoco requiere apareamientos o coincidencias en el espacio con otros individuos para la reproducción.

#### 4.1.2. D: “Design concepts”

En relación a la emergencia podemos indicar que la dinámica de la población no ha estado preconcebida por ninguna función ni definida previamente, sino que surge, aparece como resultado de la interacción del comportamiento de los dos tipos de individuos que la configuran, los cuales deben adaptarse a la cantidad de hierba disponible que encuentran en su entorno. No hay ni tasa de crecimiento, ni de reproducción, ni de mortalidad constante para los individuos, la evolución temporal del número de individuos “egoístas” y de “solidarios” es la consecuencia de la energía que pueden conseguir en su entorno, de las condiciones del medio en el que se desarrollan. No hay adaptación individual ya que la estrategia alimentaria del individuo no cambia ni se modifica, se mantiene durante toda su existencia. En el ámbito de la biología, la finalidad es el éxito de un individuo de pasar sus genes a las siguientes generaciones. En este caso, la finalidad de los individuos es sobrevivir como comunidad, es decir, que su comportamiento a la hora de alimentarse les permita existir como colectivo, ya que a nivel individual el modelo no les permite variar su estrategia de alimentación.

Lo que está por decidir es su supervivencia como comunidad. En relación a la interacción, esta se encuentra de forma indirecta ya que los individuos compiten por el mismo recurso energético, la hierba. Los individuos no pueden ni predecir las consecuencias futuras de sus actuaciones ni tienen sensibilidad para conocer o percibir el entorno y utilizar esa información para tomar decisiones con el fin de adaptarse mejor, la estrategia que utilizan para alimentarse está fijada para cada

uno de ellos y su movimiento es aleatorio. Ellos no pueden optar por modificar su conducta o su manera de proceder.

La estocasticidad se encuentra presente en el movimiento de los individuos por el dominio (razonable si los individuos no pueden percibir las propiedades de su entorno y decidir), y en el crecimiento de la hierba donde múltiples factores son los que interviene en el hecho de que rebrote o mejore el estado de la hierba en el suelo, siendo algunos casos más favorables que otros para poder regenerar la hierba. Con un número reducido de individuos, aparece una interacción local, pero a medida que la población aumenta, sus individuos tiene la posibilidad de explorar un mayor espacio y esta interacción se convierte en global. La interacción pasa por la competencia en la consecución de la hierba (fuente de energía). La interacción global también se podría dar en caso de que el tiempo de evolución del sistema fuera suficientemente dilatado para que los individuos llegaran a pasar por todo el dominio. En relación a colectivos, como se ha mencionado existe un colectivo “egoísta” y otro “solidario”. En relación a la observación del sistema simulado, obtenemos tanto las evoluciones temporales de los tamaños de estos dos colectivos de individuos como la distribución espacial de éstos en la superficie del dominio donde evolucionan, junto con el nivel de hierba presente en cada parcela (Fig. 3).

#### 4.1.3. D: “Details”

**Inicialización:** Los individuos se disponen aleatoriamente en el dominio con la misma cantidad de energía inicial (50 unidades de energía inicial, lo que significa que un individuo podría mantenerse un número de pasos de programa sin ingerir hierba antes de morir, y este periodo de supervivencia depende de la energía asignada al metabolismo). La composición de la población se determina de forma aleatoria fijando la probabilidad de tener individuos pertenecientes a la comunidad “solidaria”. El nivel de hierba inicial es el máximo y uniforme en todo el dominio. La Fig. 3 muestra las condiciones de partida de la simulación. En la Tabla 2 se encuentran los valores de los parámetros utilizados para iniciar las simulaciones.

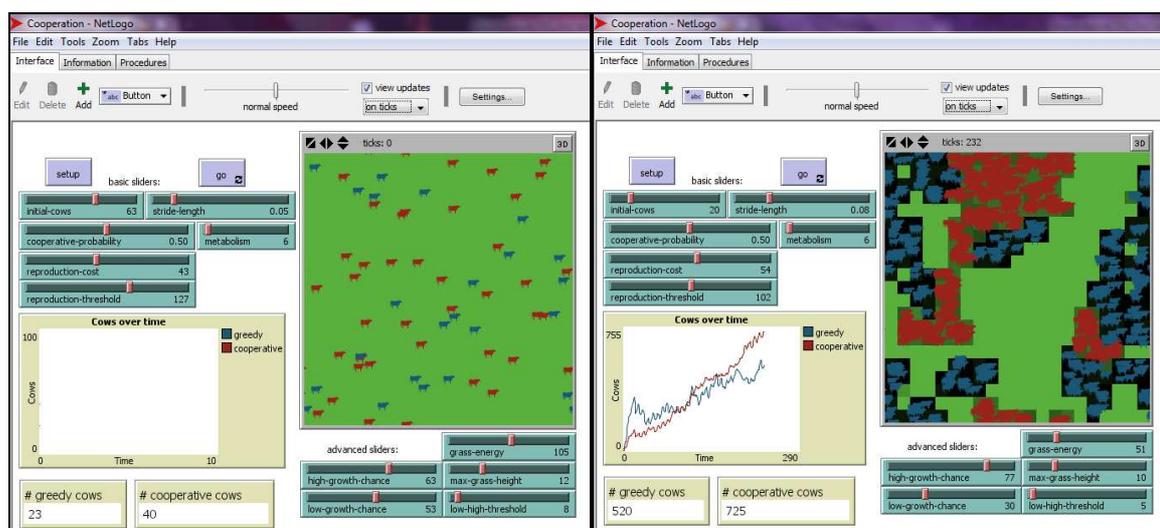


Figura 3: Capturas de pantalla que muestran el estado inicial de una simulación (izquierda) y el estado del sistema en otro instante de tiempo (derecha) del IBM "Cooperation" de la galería de NetLogo.

| Variable   | Valor por defecto | Valor mínimo | Valor máximo |
|--|-------------------|--------------|--------------|
| Número de individuos iniciales   | 20                | 0            | 100          |
| Probabilidad de individuos "solidarios"                                | 0.5               | 0            | 1            |
| Gasto energético del movimiento/unidad temporal                        | 6                 | 0            | 99           |
| Distancia a la que se puede desplazar en el movimiento/unidad temporal | 0.08              | 0            | 0.3          |
| Gasto energético en la reproducción individual                         | 54                | 0            | 99           |
| Nivel de energía mínimo para poder reproducirse                        | 102               | 0            | 200          |
| Energía que proporciona la hierba                                      | 51                | 0            | 200          |
| Cantidad máxima de hierba/unidad ambiente                              | 10                | 1            | 40           |
| Probabilidad alta para el crecimiento de la hierba                     | 77                | 0            | 99           |
| Probabilidad baja para el crecimiento de la hierba                     | 30                | 0            | 99           |
| Umbral o nivel de hierba que condiciona su crecimiento                 | 5                 | 0            | 99           |

**Tabla 2: Rango de valores de los parámetros y variables de entrada para la ejecución de simulaciones correspondientes al IBM "Cooperation" de la galería NetLogo.**

**Entrada.** No se consideran ni entradas ni salidas de individuos o recursos energéticos del sistema durante la evolución, ni tampoco se contemplan condiciones medioambientales externas (lluvias, ciclos de temperaturas...) que modifiquen la regeneración de hierba.

**Sub-modelos.** Los diferentes sub-modelos que integran el IBM "Cooperation" son: movimiento, alimentación, reproducción y muerte individual, y rebrote de hierba. Considerando que el IBM está implementado en la web de NetLogo, a continuación se muestra cada uno de los procedimientos del código de programación con la correspondiente descripción verbal de lo que representan.

|   |   |
|---|---|
| Movimiento  |   |
| <i>rt random 360</i><br><i>fd stride-length</i><br><br><i>set energy energy - metabolism</i>  | Aleatoriamente se escoge una dirección y se avanza la distancia determinada para situarse en una nueva posición.<br>Disminución de la energía individual (este coste se puede asimilar a la energía de metabolismo que todo individuo requiere para mantenerse).  |
| Reproducción  |   |
| <i>if energy &gt; reproduction-threshold [</i><br><i>set energy energy - reproduction-</i><br><i>cost</i><br><i>hatch 1</i><br><i>]</i> | Cuando la energía acumulada es mayor que la que se ha fijado para la reproducción, ésta tiene lugar, con un coste energético (disminuye la energía que se tenía). Aparecerá un nuevo individuo con las mismas características que el ascendiente (posición, energía, estrategia alimentaria).   |
| Alimentación: Ingesta "solidaria"   |   |
| <i>if grass &gt; low-high-threshold [</i><br><i>set grass grass - 1</i><br><i>set energy energy + grass-energy</i><br><i>]</i>          | Ingerirá hierba únicamente cuando el nivel presente en la unidad de ambiente sea superior al umbral fijado, ya que por debajo de ese valor la probabilidad de rebrote de la hierba es pequeña.<br>Actualización del nivel de hierba después de la ingesta.<br>Incremento de energía individual dependiendo del valor energético que se le haya asignado a la hierba |

|  |  |
|--|--|
| Alimentación: Ingesta “egoísta”  |  |
| <pre> if grass &gt; 0 [   set grass grass - 1   set energy energy + grass-energy ] </pre>  | <p>Ingerirá hierba siempre que haya disponibilidad en su unidad ambiente.</p> <p>Actualización del nivel de hierba después de la ingesta.</p> <p>Incremento de energía individual dependiendo del valor energético que se le haya asignado a la hierba</p>   |
| Muerte   |  |
| <pre> if energy &lt; 0 ( die ) </pre>  | <p>Si ha gastado toda su energía de reserva y no ha podido conseguir más, el individuo morirá.</p>   |
| Crecimiento de la hierba   |  |
| <pre> ifelse ( grass &gt;= low-high-threshold) [   if high-growth-chance &gt;= random- float 100   [ set grass grass + 1 ] ] if low-growth-chance &gt;= random- float 100   [ set grass grass + 1 ] ] if grass &gt; max-grass-height   [ set grass max-grass-height ] </pre> | <p>Por sorteo aleatorio se determinara si hay o no crecimiento de hierba. Si la cantidad de hierba en una unidad ambiente es mayor que el valor umbral fijado que condiciona su rebrote, entonces la probabilidad de que suceda será elevada, mientras que si la cantidad de hierba es menor que ese umbral, la probabilidad de que rebrote es menor.</p> <p>Si la hierba en unidad de ambiente es ya la máxima que se ha fijado no hay incremento, y se mantiene este valor máximo.</p> |

## 4.2. Simulaciones con el IBM “Cooperation” de NetLogo

Avanzar en la comprensión del funcionamiento de un modelo computacional requiere de la ejecución reiterada de éste. La componente aleatoria presente en este tipo de modelo es la responsable de que las repeticiones de ejecuciones con el mismo juego de parámetros no produzcan exactamente los mismos resultados (como las réplicas de un experimento). Por este motivo, la exploración completa y exhaustiva de este tipo de modelo requiere de la realización de series de simulaciones, y de un análisis estadístico de los resultados conseguidos. En este caso, por tener este IBM algunos sub-modelos estocásticos como son el movimiento al azar de los individuos o el rebrote (crecimiento) de la hierba, las sucesivas simulaciones que se ejecutan (aún con el mismo conjunto de parámetros) reproducen distintos itinerarios para el sistema. Además, la posibilidad de generar simulaciones modificando valores iniciales de parámetros o variables del modelo ayuda a entender cómo trabaja el sistema simulado, obliga a razonar el porqué de los comportamientos observados, y a inferir respuestas a cuestiones planteadas.

A modo de ejemplo, y para ilustrar este tipo de estudio, para este trabajo se han realizado simulaciones variando el parámetro que fija la distancia que los individuos recorren en el espacio en cada paso de programa. Disminuyendo o aumentado el valor de este parámetro, los individuos tardaran más o menos tiempo en cambiar de entorno y explorar nuevas parcelas. Se presentan tres resultados de simulación con el fin de analizar el efecto que tiene este parámetro sobre los dos colectivos de individuos (“egoístas” y “solidarios”). La primera simulación utilizará para esta distancia el valor por defecto que tiene asignado el IBM en la galería NetLogo, mientras que para la segunda y la tercera simulación se disminuirá y aumentará este valor. Los valores son 0.08, 0.04 (0.08-0.04) y 0.12 (0.8+0.04)

respectivamente. La Fig. 4 presenta las evoluciones obtenidas para cada uno de estos tres casos. En la primera simulación el comportamiento del colectivo "egoísta" es el que se impone con éxito, ya que aunque corto plazo se constata un crecimiento mayor de individuos "solidarios", cuando éste alcanza su máximo, empieza a decrecer de forma súbita hasta su extinción. La evolución del colectivo "egoísta" es diferente, con una velocidad menor pero sostenida que le permite un tamaño de colectivo aproximadamente estable, con picos de máximos y mínimos condicionado por la capacidad energética que proporciona la superficie de hierba (consumo y rebrote sucesivamente). Con una disminución de 0.04 unidades de distancia/unidad temporal para el movimiento respecto al valor defecto, se observa en la segunda simulación un crecimiento exponencial inicial como en el caso anterior. No obstante esta vez se observa que los individuos "solidarios" ni decrecen ni se extinguen, sino que se mantienen en un tamaño de colectivo aproximadamente estable y mucho mayor que el que pueden alcanzar los "egoístas". En esta segunda simulación el colectivo "egoísta" tiene un crecimiento mucho más limitado, manteniéndose el número de individuos de esta comunidad muy por debajo de la de los "solidarios". En la tercera simulación, con un aumento en 0.04 unidades de distancia/unidad temporal en el movimiento individual respecto al valor defecto se observa un patrón a largo plazo similar al de la primera simulación realizada. A corto plazo, en la primera etapa, el crecimiento de estos dos colectivos no es igual ya que las interacciones locales en ese momento son muy importantes y significativas para el arranque de los dos crecimientos (Fig. 4).

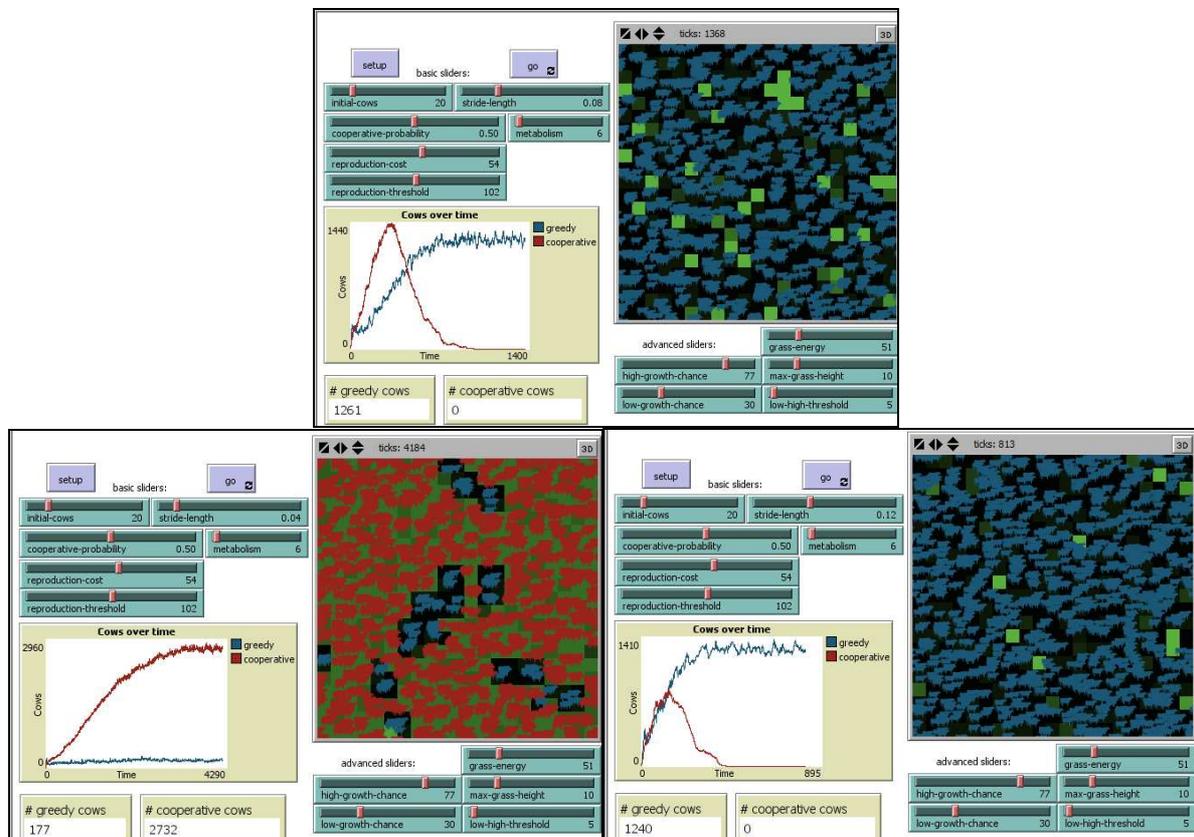


Figura 4: Tres simulaciones correspondiente al modelo "Cooperation" de la galería de NetLogo variando el parámetro que condiciona el alcance de movimiento de los individuos.

La distancia que los individuos pueden recorrer por unidad temporal se encuentra íntimamente relacionada con el triunfo final de uno de los colectivos. Como más pequeña sea esta distancia a explorar por parte de los individuos, los "solidarios" tendrán más éxito que los "egoístas", ya que su estrategia alimenticia permite regenerar de forma más eficiente la hierba y posibilita el mantenimiento del entorno. Cuando mayor sea la distancia que puedan recorrer los individuos, más exitosa será la estrategia "egoísta" ya que podrán desplazarse más lejos para conseguir más hierba. La máxima capacidad de carga del sistema planteado es diferente si el éxito lo consigue un colectivo u otro. En el primer y tercer caso, donde el comportamiento "egoísta" es el más viable a largo plazo, la capacidad oscila entre 1200 y 1400 individuos, mientras que en el segundo caso, donde la dinámica "solidaria" es mayoritaria, el número de individuos oscila al entorno de 2900 individuos. Este hecho es consecuencia de que no hay un número máximo de animales por parcela y, por lo tanto, en dos parcelas diferentes, con el mismo alimento disponible, una compuesta únicamente por animales "solidarios" y el otro por "egoístas", se alimentarían más individuos con comportamiento "solidario". La distribución del nivel o cantidad de hierba también corrobora esta discusión. Finalmente mencionar que en la distribución espacial de los individuos se observa la formación de conglomerados, agrupaciones de individuos del mismo tipo, debido a la manera de reproducirse, y a que hay difusión de individuos por movimiento y no redistribución (Fig. 4). Así, pues, en este modelo "Cooperation" el control del espacio ha resultado ser muy significativo y decisivo para su propósito. Otras muchas consideraciones, comentarios y sugerencias se podrían elaborar con series de simulaciones modificando los valores de otros parámetros del IBM.

## 5. Conclusiones

Los IBMs se están convirtiendo en una herramienta válida para describir y tratar con sistemas complejos formados por entidades autónomas. Estos modelos consisten típicamente en un cierto número de individuos definidos a partir de sus comportamientos específicos y parámetros característicos en un ambiente o marco cambiante dentro del cual las interacciones se suceden. Conceptualmente son atractivos y ofrecen múltiples alternativas para ser aplicados en contextos muy distintos. No obstante, por ser modelos computacionales requieren de un entorno adecuado para que puedan ser implementados y ejecutados con éxito. Sin la posibilidad de obtener resultados de simulación de forma razonablemente eficiente, no podrían ser considerados nunca como una opción viable para actividades académicas en el aula. La posibilidad de tratar cómodamente con IBMs en entornos informáticos apropiados sugiere que la preparación de material docente sería viable, y se podrían convertir en una opción para trabajar la competencia de la modelización presente en la mayoría de los planes de estudios de ámbito científico o tecnológico. La plataforma NetLogo, de acceso libre vía web, es una buena elección para incorporar los IBMs en la docencia. El uso de la galería de modelos que contiene su web favorece y facilita la divulgación de los IBMs en el ámbito educativo. La competencia para la modelización incluye ser capaz de: a) estructurar el campo o situación que va a modelizarse; b) traducir la realidad a una estructura matemática; c) interpretar los modelos matemáticos en términos reales; d) reflexionar, analizar y criticar un modelo y sus resultados; e) comunicar las características y los resultados de un modelo, incluyendo sus limitaciones; y f) dirigir y controlar el proceso de

modelización. Todos estos aspectos podrían ser abordados con el uso de los IBMs. El ejemplo de IBM desplegado en detalle en este trabajo ha ilustrado algunas de estas acciones. El poder trabajar la competencia en modelización con contextos de aplicación atractivos, como puede ser el de la biología, es extremadamente favorable para su correcta adquisición. Al mismo tiempo, este tipo de modelo computacional facilita la experimentación y manipulación “virtual” sobre los sistemas modelizados, generando escenarios para debatir y argumentar, estimulando una perspectiva de modelización muy intuitiva. Estamos seguros que trabajos de este tipo ayudaran a divulgar las características propias de los IBMs, a fin de que esta metodología de modelización pueda ser incorporada en el ámbito académico de forma progresiva, complementando otras propuestas de modelización ya existentes.

### Bibliografía

- Aravena M., Caamaño C., Gimenez J. (2008). Modelos matemáticos a través de proyectos. *Revista Latinoamericana de Investigación en Matemática Educativa* 11, 49-92.
- Colella V.S., Klopfer E., Resnick M. (2001). *Adventures in Modeling: Exploring Complex Dynamic Systems with StarLogo*. Teachers College Press, New York.
- DeAngelis D.L., Mooij W.M. (2005). Individual-based modelling of ecological and evolutionary processes. *Annual Review of Ecology, Evolution, and Systematics* 36, 147-68.
- Gilbert, J.K. (2004). Models and Modelling: Routes to More Authentic Science Education. *International Journal of Science and Mathematics Education* 2, 115-130.
- Gomes-Neves R., Duarte-Teodoro V. (2010). Enhancing Science and Mathematics Education with Computational Modelling. *Journal of Mathematical Modelling and Application* 1, 2-15.
- Gómez-Moureló P., Ginovart M. (2009) The differential equation counterpart of an individual-based model for yeast population growth. *Computers and Mathematics with Applications* 58, 1360-1369.
- Grimm V. (1999). Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the future? *Ecological Modelling* 115, 129-148.
- Grimm V., Railsback S.F. (2005). *Individual-based modeling and ecology*. Princeton University Press, Princeton and Oxford.
- Grimm V., Berger U., Bastianen F., Sigrunn E., Ginot V., et al. (2006). A standard protocol for describing individual-based and agent-based models. *Ecological Modelling* 198, 115-126.
- Jacobson M., Wilensky U. (2006). Complex systems in education: Scientific and educational importance and implications for the learning sciences. *Journal of the Learning Sciences* 15(1), 11-34.
- Justi R. (2006). La enseñanza de ciencias basada en la elaboración de model;; os. *Enseñanza de las Ciencias* 24, 173-184.
- Luke S., Cioffi-Revilla C., Panait L., Sullivan K., Balan G. (2005). MASON: A Multiagent Simulation Environment. *Simulation: Transactions of the society for Modeling and Simulation International* 82(7), 517-527.
- Minar N., Burkhart R., Langton C., Askenazi M. (1996). The Swarm simulation system: A toolkit for building multi-agent simulations. <http://santafe.ed/projects/swarm>. Extraído el día 15 de octubre de 2010.

- North M.J., Collier N.T., Vos J.R., (2006). Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit. *ACM Transactions on Modeling and Computer Simulation* 16, 1-25.
- Railsback S.F. (2001). Concepts from complex adaptive systems as a framework for individual-based modelling. *Ecological Modelling* 139, 47-62.
- Railsback S.F., Lytinen S.L., Jackson S.K. (2006). Agent-based simulation platforms: Review and development recommendations. *Simulation* 82, 609-623.
- Wilensky U. (1997). NetLogo Cooperation model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/models/cooperation>. Extraído el día 15 de octubre de 2010.
- Wilensky U. (1999). NetLogo. Center for Connected Learning and Computer-Based Modelling, Northwestern University. Evanston, IL. Extraído el día 15 de octubre de 2010. <http://ccl.northwestern.edu/netlogo/>.
- Wilensky U., Rand W. (en prensa). *An introduction to agent-based modeling: Modeling natural, social and engineered complex systems with NetLogo*. MA: MIT Press, Cambridge.

**Marta Ginovart.** Licenciada en Ciencias-Sección de Matemáticas por la Universidad Autónoma de Barcelona y Doctora en Ciencias-Matemáticas por la Universidad Politécnica de Cataluña, donde actualmente es profesora en el área de Matemática Aplicada (Barcelona, España). Su actividad investigadora se centra en la modelización y simulación discreta de sistemas biológicos, así como en el análisis estadístico de datos. Ha participado en diversos proyectos de innovación docente para la mejora del aprendizaje de las matemáticas. [marta.ginovart@upc.edu](mailto:marta.ginovart@upc.edu)

**Xavier Portell** Ingeniero agrónomo por la Universidad de Lérida. Actualmente realiza su doctorado en el grupo de Modelización y Simulación Discreta de Sistemas Biológicos de la Universidad Politécnica de Cataluña (Barcelona, España). Su actividad investigadora se centra en el uso de la modelización basada en el individuo para el estudio de levaduras en procesos de interés biotecnológico y agroalimentario. [xavier.portell@upc.edu](mailto:xavier.portell@upc.edu)

**Pol Ferrer-Closas.** Ingeniero Técnico Agrícola por la Escuela Superior de Agricultura de Barcelona de la Universidad Politécnica de Cataluña. Actualmente finaliza esta finalizando los estudios de Ingeniería de Montes en la Escuela Superior de Ingeniería Agraria de Lérida de la Universidad de Lérida (España). [Ferrer.closas.pol@gmail.com](mailto:Ferrer.closas.pol@gmail.com)

**Mónica Blanco.** Licenciada en Matemáticas por la Universidad de Barcelona y Doctora en Matemáticas por la Universidad Autónoma de Barcelona. Actualmente es profesora en el área de Matemática Aplicada de la Universidad Politécnica de Cataluña (Barcelona, España). Su actividad investigadora se centra en la Historia de las Matemáticas, así como en el análisis estadístico de datos. Ha participado en diversos proyectos de innovación docente para la mejora del aprendizaje de las matemáticas. [monica.blanco@upc.edu](mailto:monica.blanco@upc.edu)