

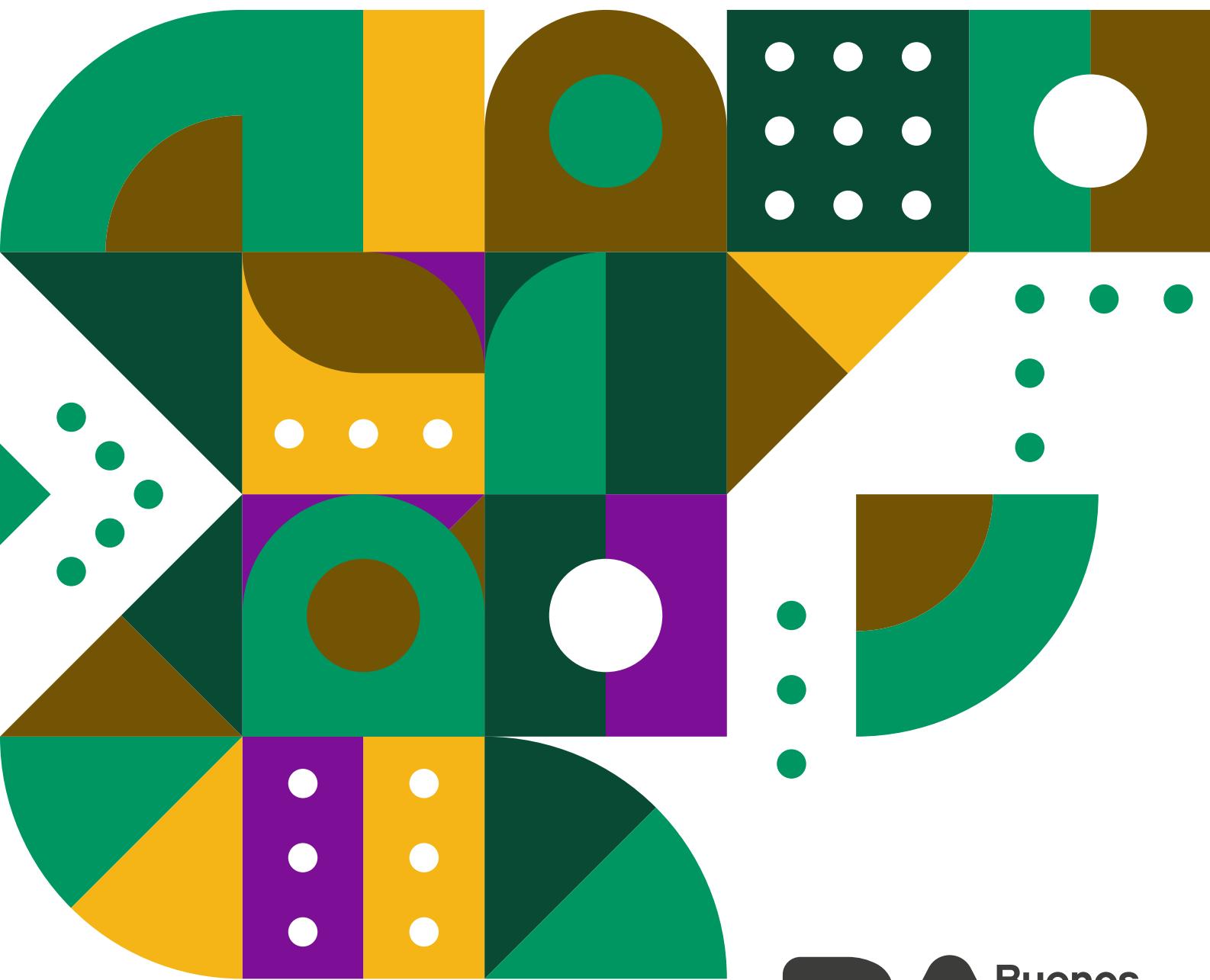


SERIE

Educación Digital, Programación y Robótica

# Agro y Ambiente

## Simulaciones computacionales: el agro y su entorno



**Jefe de Gobierno**

Horacio Rodríguez Larreta

**Ministra de Educación**

María Soledad Acuña

**Jefe de Gabinete**

Manuel Vidal

**Subsecretaría de Coordinación Pedagógica y Equidad Educativa**

María Lucía Feced Abal

**Subsecretario de Carrera Docente**

Oscar Mauricio Ghillione

**Subsecretario de Tecnología Educativa y Sustentabilidad**

Santiago Andrés

**Subsecretario de Gestión Económico Financiera y Administración de Recursos**

Sebastián Tomaghelli

**Subsecretaría de la Agencia de Aprendizaje a lo Largo de la Vida**

Eugenio Cortona

**Directora Ejecutiva de la Unidad de Evaluación Integral de la Calidad y Equidad Educativa**

Carolina Ruggero

**Director General de Planeamiento Educativo**

Javier Simón

**Directora General de Educación Digital**

Rocío Fontana

**Gerenta Operativa de Currículum**

Mariana Rodríguez

**Gerente Operativo Tecnología e Innovación Educativa**

Roberto Tassi

## Dirección General de Planeamiento Educativo (DGPLEDU)

### Gerencia Operativa de Curículum (GOC)

Mariana Rodríguez

**Asesora técnica pedagógica:** Carola Martínez.

### Equipo de especialistas en didáctica del Nivel Secundario:

Bettina Bregman (coordinación), Cecilia Bernardi, Ana Campelo, Mariana Gild, Marta Libedinsky, Adriana Vanin.

**Especialistas:** Marta Libedinsky (coordinación general), Hernán Miguel (GOC - Enlace Ciencias - Referente de la Orientación Agro y Ambiente), Andrea Clérici (Enlace Ciencias), Patricia Moreno (GOC), Cristián Rizzi Iribarren (Enlace Ciencias - Especialista en Educación Digital), María Mercedes Zambrana (Enlace Ciencias).

**Colaboradores:** Gabriela Fernanda Jiménez (GOC - Enlace Ciencias).

## Subsecretaría de Tecnología Educativa y Sustentabilidad (SSTES)

### Dirección General de Educación Digital (DGED)

### Gerencia Operativa Tecnología e Innovación Educativa (INTEC)

Roberto Tassi

**Especialistas de Educación Digital:** Julia Campos (coordinación), Uriel Frid, Ignacio Spina.

---

### Equipo Editorial de Materiales y Contenidos Digitales (DGPLEDU)

**Coordinación general:** Silvia Saucedo.

**Coordinación editorial:** Marcos Alfonzo.

**Asistencia editorial:** Leticia Lobato.

**Edición:** Ana Cecilia Forlani.

**Corrección de estilo:** Ana Premuzic.

**Diseño gráfico y desarrollo digital:** Alejandra Mosconi, Gabriela Ognio.

**Producción audiovisual:** Nicolás Bustamante, Joaquín Simón (edición), Vanina Barbeito (locución).

**Imágenes:** FreePik, Pxfuel.com, Wikimedia Commons.

---

ISBN en trámite.

Se autoriza la reproducción y difusión de este material para fines educativos u otros fines no comerciales, siempre que se especifique claramente la fuente. Se prohíbe la reproducción de este material para venta u otros fines comerciales.

Las denominaciones empleadas en este material y la forma en que aparecen presentados los datos que contiene no implican, de parte del Ministerio de Educación del Gobierno de la Ciudad Autónoma de Buenos Aires, juicio alguno sobre la condición jurídica o nivel de desarrollo de los países, territorios, ciudades o zonas, o de sus autoridades, ni respecto de la delimitación de sus fronteras o límites.

La mención de empresas o productos de fabricantes en particular, estén o no patentados, no implica que el Ministerio de Educación del Gobierno de la Ciudad Autónoma de Buenos Aires los apruebe o recomiende de preferencia a otros de naturaleza similar que no se mencionan.

Fecha de consulta de imágenes, videos, textos y otros recursos digitales disponibles en Internet: 15 de enero de 2022.

© Gobierno de la Ciudad Autónoma de Buenos Aires / Ministerio de Educación. Dirección General de Planeamiento Educativo / Gerencia Operativa de Curículum, 2022. Carlos H. Perette y Calle 10. –C1063 – Barrio 31 – Retiro – Ciudad Autónoma de Buenos Aires.

© Copyright © 2022 Adobe Systems Software. Todos los derechos reservados. Adobe, el logo de Adobe, Acrobat y el logo de Acrobat son marcas registradas de Adobe Systems Incorporated.

# Presentación

La serie *Educación Digital, Programación y Robótica* contiene diversas propuestas de enseñanza para el desarrollo de los contenidos, conceptos, capacidades, prácticas, valores y actitudes definidos en el Diseño Curricular de la NES y en el *Anexo Curricular de Educación Digital, Programación y Robótica* (resolución en proceso).

La propuesta de esta serie se enmarca en las Resoluciones N° 321/MEGC/2015 y N° 1189/MEGC/2015 y sus modificatorias N° 1189/MEGC/2015 y 3510/MEGC/2015, en la Resolución CFE N.º 263/15 y en los Núcleos de Aprendizaje Prioritarios para Educación Digital, Programación y Robótica aprobados por el Consejo Federal de Educación mediante la Resolución N° 343/18.

Además, responde a las características y las modalidades de trabajo pedagógico señaladas en el documento *Orientaciones para la Organización Pedagógica e Institucional de la Educación Obligatoria*, aprobado por la Resolución CFE N.º 93/09, que establece el propósito de fortalecer la organización y la propuesta educativa de las escuelas de Nivel Secundario de todo el país. A esta norma actualmente vigente, se agrega el documento *MOA - Marco de Organización de los Aprendizajes para la Educación Obligatoria Argentina*, aprobado por la Resolución CFE N.º 330/17, que plantea la necesidad de instalar distintos modos de apropiación de los saberes que den lugar a nuevas formas de enseñanza, de organización del trabajo docente y del uso de los recursos y los ambientes de aprendizaje.

En todas las normas mencionadas se promueven diversas modalidades de organización institucional, un uso flexible de los espacios y de los tiempos y nuevas formas de agrupamiento de las y los estudiantes que se traduzcan en talleres, proyectos, articulación entre espacios curriculares, experiencias formativas y debates, entre otras actividades, en las que incluso participen estudiantes de diferentes años. En el ámbito de la Ciudad, el *Diseño Curricular de la Nueva Escuela Secundaria* incorpora temáticas emergentes y abre la puerta para el abordaje de problemáticas actuales de significatividad social y personal para la población joven.

La normativa vigente permite afirmar que existe acuerdo sobre la magnitud de los cambios que demanda el nivel secundario para lograr incluir al conjunto de estudiantes, y promover los aprendizajes necesarios para el ejercicio de una ciudadanía responsable y la participación activa en ámbitos laborales y de formación. En este sentido, si bien se ha recorrido un importante camino, es indispensable profundizar, extender e incorporar propuestas que ofrezcan reales oportunidades de aprendizaje

y hagan de la escuela un lugar convocante y un espacio privilegiado para despertar inquietudes y vocaciones.

Los materiales que componen la serie articulan contenidos propios de los espacios curriculares de la formación general y de la formación específica de los bachilleratos orientados con contenidos de Educación Digital, Pensamiento Computacional, Programación y Robótica. Ofrecen orientaciones y una guía de actividades que culminan con una producción que anticipa y plantea tres diferentes niveles de logro, de manera de contemplar los diversos contextos o entornos.

El común denominador de los materiales es proponer problemas y temáticas que resultan desafiantes e interesantes para los y las jóvenes que cursan la escuela secundaria y ofrecer oportunidades y estrategias para que “aprendan haciendo”, diseñen, creen y recreen de manera sencilla y accesible productos y/o artefactos en forma individual o grupal con la guía del o de la docente. Al mismo tiempo, contribuyen al desarrollo gradual de capacidades para la exploración y el trabajo autónomo, a partir de las orientaciones precisas y claras sobre los procedimientos adecuados para el manejo de aplicaciones y de los entornos virtuales. Se espera que, a partir de estas experiencias, los y las estudiantes puedan apasionarse y continuar en forma individual o con sus compañeros y compañeras la indagación de otros problemas que conectan tecnología, ciencia, filosofía, sociedad, política y cultura.

Cabe aclarar que, en algunos casos, se podrá adoptar la propuesta completa y, en otros, seleccionar las partes que se consideren más convenientes. Asimismo, se podrá plantear un trabajo de mayor articulación o exigencia de acuerdos entre docentes, puesto que serán los equipos de profesores y profesoras quienes podrán tomar las decisiones didácticas en las que el uso de estos materiales cobre sentido.

Confiamos en que estos recursos didácticos constituirán un gran aporte para el trabajo cotidiano en las instituciones educativas de nivel secundario y, como toda serie en construcción, seguirá incorporando y poniendo a disposición de las escuelas de la Ciudad nuevas propuestas, que darán lugar a nuevas experiencias y nuevos aprendizajes.



**Javier Simón**  
Director General  
de Planeamiento Educativo



**Mariana Rodríguez**  
Gerenta Operativa  
de Currículum

# ¿Cómo se navegan los textos de esta serie?

Los materiales de la serie Educación Digital, Programación y Robótica cuentan con elementos interactivos que permiten la lectura hipertextual y optimizan la navegación.

## Itinerario de actividades



### Actividad 1

Organizador interactivo que presenta la secuencia completa de actividades.



Adobe Reader Copyright © 2022. Todos los derechos reservados.

Para visualizar correctamente la interactividad se sugiere bajar el programa [Adobe Acrobat Reader](#) que constituye el estándar gratuito para ver e imprimir documentos PDF.

## Pie de página



Folio, con flechas interactivas que llevan a la página anterior y a la página posterior.



Volver a vista anterior

Al cliquear regresa a la última página visitada.

## Índice interactivo

Introducción Contenidos Escenario Actividades Evaluación Anexo

Al pie de cada página se encuentra el índice interactivo, que lleva a todas las secciones del documento.

## Señalizadores gráficos

Estos íconos facilitan la localización de información relevante para el/la usuario/a, desde la columna lateral de la página.

### ETIQUETAS

Palabras clave en el planteamiento del escenario y de las actividades.



### Importante

Conceptos, recomendaciones, o reflexiones.



### Archivos

Documentos para descargar.

**Tutorial**  
**Información**  
Tutoriales o instructivos.



**Presentación**  
**Entrevista**  
**Tutorial**  
Contenido audiovisual.



**Tarjeta**  
Uso de tarjetas.



**Glosario**  
Búsqueda de palabras en la sección de glosario.

# Introducción

En esta secuencia didáctica se introduce el trabajo con modelos de simulación computacional para representar fenómenos que ocurren en los sistemas agrarios de nuestro país y de otras partes del mundo. Los/as estudiantes se desempeñarán como modelizadores, al usar, modificar y crear modelos de simulación computacionales que utilicen lenguajes y entornos desarrollados específicamente para el ámbito educativo.

Emplearán estos modelos para diseñar escenarios, elaborar hipótesis, analizar datos y comprender fenómenos propios de los sistemas agrarios. Además, integrarán sensores para realizar un tipo especial de modelización llamada *bifocal*, caracterizada por la correspondencia en tiempo real entre lo que ocurre en el mundo físico y lo que se observa en la pantalla de una computadora.

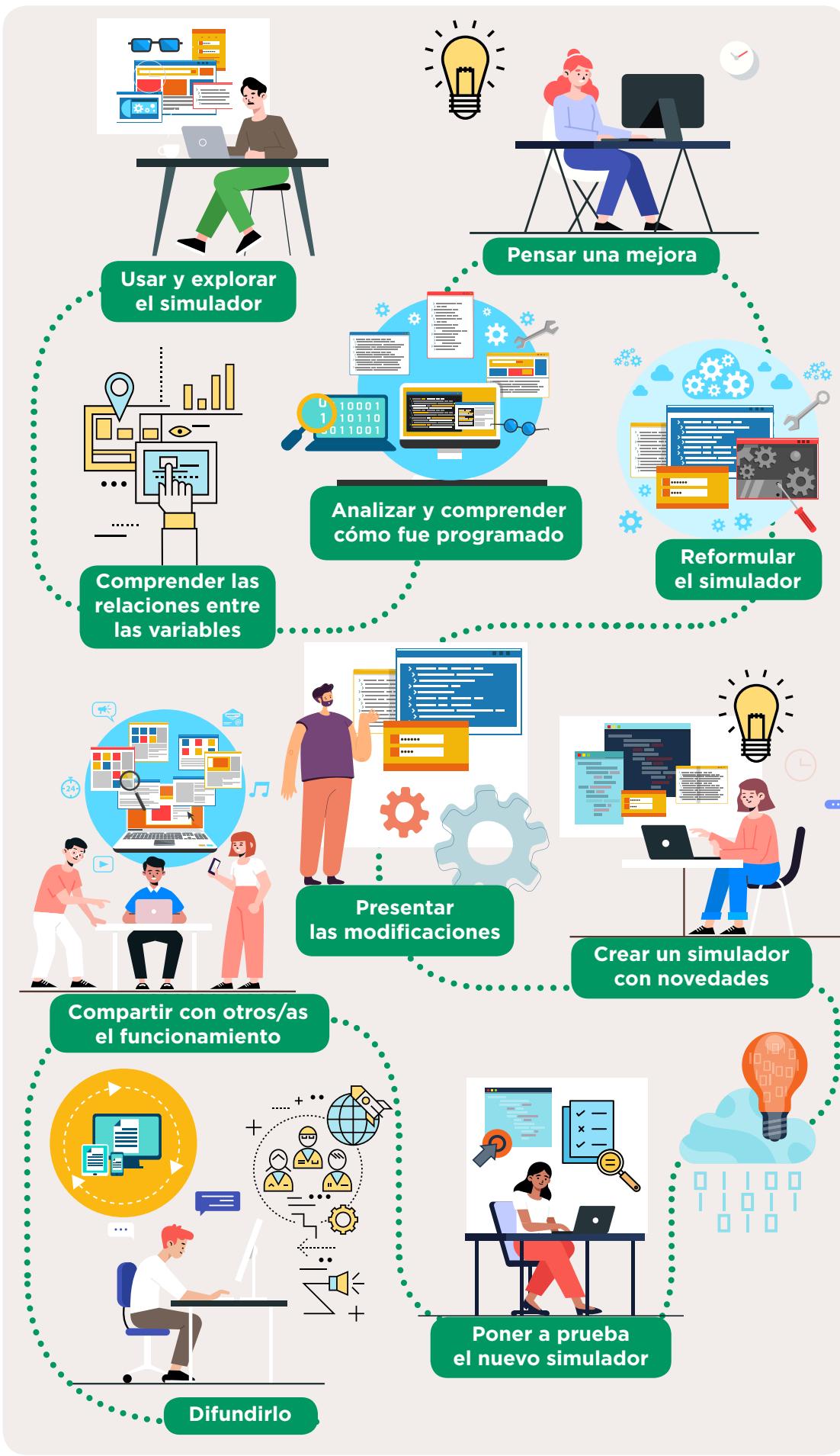
Mediante esta secuencia, los/as estudiantes trabajarán conceptos relacionados con los ecosistemas agrarios, tales como capacidad de carga, competencia interespecífica, población, predación y cadena trófica, entre otros. Abrirán la caja negra de los simuladores para comprender cómo están diseñados y cómo crear ellos/as mismos/as sus propios modelos de simulación en computadora.

Áreas del conocimiento como la biología, la física y la química computacional demuestran hoy que debemos entender el pensamiento computacional no solamente como algo propio de las ciencias de la computación, sino también integrado a otras disciplinas para dar lugar a nuevos modos de avanzar en esos campos del conocimiento y aportar soluciones que nadie había imaginado antes para problemas actuales.

Esta secuencia pretende ser una ventana por la cual asomarse a ese nuevo escenario científico y tecnológico.



# Simulaciones computacionales: el agro y su entorno



- Introducción
- Contenidos
- Escenario
- Materiales
- Actividades
- Evaluación
- Explorando fronteras
- Anexos
- Tarjetas
- Glosario
- Bibliografía

# Objetivos de aprendizaje, contenidos y capacidades

Los objetivos de aprendizaje, los contenidos y las capacidades que presenta el siguiente cuadro pertenecen a la Orientación Agro y Ambiente, y están articulados con los contenidos del *Diseño Curricular de Educación Digital, Programación y Robótica*.

Objetivos de aprendizaje	
<b>Desde la Orientación en Agro y Ambiente se propone:</b> .....	
• Comprender cómo los factores abióticos y la biodiversidad intervienen en la estructuración del espacio agrario.	
• Caracterizar la estructura, dinámica y cambios de las poblaciones vegetales y animales en un ecosistema agrario.	
• Comprender problemáticas agrarias desde una perspectiva basada en conceptos de la ecología.	
• Reconocer la relación entre la alteración ambiental causada por la acción humana y la pérdida de la biodiversidad.	
<b>Desde Educación Digital se propone:</b> .....	
• Desarrollar el pensamiento computacional como estrategia de resolución de problemas integrando la programación, la robótica y diversas tecnologías digitales para abordar y comprender problemáticas del entorno social, económico, ambiental y cultural.	
• Comprender, modificar y crear entornos de simulación para abordar problemáticas propias del campo del agro y el ambiente.	

Contenidos	Capacidades
<p><b>Ambiente agrario</b></p> <p><i>Ecología orientada a los problemas agrarios</i></p> <ul style="list-style-type: none"> <li>• Caracterización y estudio de los factores abióticos (agua, aire y suelo) en relación con las actividades agrarias.</li> <li>• El concepto de población en ecología. Estructura y dinámica de las poblaciones.</li> <li>• Propiedades de las poblaciones.</li> <li>• El efecto de la capacidad de carga en las actividades agrarias. Nicho ecológico. Recursos y factores limitantes.</li> <li>• Interacciones entre poblaciones.</li> <li>• Factores que afectan la diversidad: clima, recursos, interacciones entre especies, actividades humanas, disturbios naturales. Influencia de los cambios en las comunidades sobre los factores abióticos.</li> <li>• Ecosistemas. Modelos tróficos del ecosistema: cadenas y redes; pirámides.</li> <li>• Parámetros ecológicos a considerar en los sistemas de producción agraria.</li> <li>• Análisis e identificación de variables.</li> <li>• Apropiación de lenguajes y entornos de programación. Abordaje de diferentes elementos y estructuras de programación: <ul style="list-style-type: none"> <li>➢ Eventos</li> <li>➢ Repeticiones/iteraciones</li> <li>➢ Condicionales</li> </ul> </li> <li>• Circuito de la información en los dispositivos robóticos: entradas, procesos y salidas. Trabajo con sensores.</li> <li>• Componentes electrónicos: señales analógicas y digitales.</li> <li>• Modelización a través de entornos de simulación bifocal.</li> <li>• Formulación de esquemas de relaciones entre variables.</li> </ul>	<p><b>Comunicación</b></p> <ul style="list-style-type: none"> <li>• Habilidad para leer, escribir, hablar y escuchar de manera adecuada en cualquier situación.</li> </ul> <p><b>Pensamiento crítico, iniciativa y creatividad</b></p> <ul style="list-style-type: none"> <li>• Habilidad para adoptar una postura personal y original respecto de una problemática determinada, analizando rigurosamente la información sobre la base de los conocimientos y saberes disponibles.</li> </ul> <p><b>Ánalisis y comprensión de la información</b></p> <ul style="list-style-type: none"> <li>• Habilidad para seleccionar y trabajar profundamente sobre la información, que puede ser obtenida de distintas fuentes, con posibilidades de extraer conclusiones y transferirlas a otros ámbitos.</li> </ul> <p><b>Resolución de problemas y conflictos</b></p> <ul style="list-style-type: none"> <li>• Habilidad para enfrentarse a las situaciones problemáticas de cualquier índole desde una perspectiva positiva, logrando resolver y superar los conflictos mediante el trabajo participativo y el diálogo.</li> </ul> <p><b>Interacción social, trabajo colaborativo</b></p> <ul style="list-style-type: none"> <li>• Habilidad de encontrarse e interactuar de manera adecuada a la circunstancia con las personas que nos rodean y trabajar en equipo respetando la diversidad de opiniones, posturas y puntos de vista.</li> <li>• Pensamiento computacional: <ul style="list-style-type: none"> <li>➢ Abstracción</li> <li>➢ Descomposición en partes</li> <li>➢ Reconocimiento de patrones</li> <li>➢ Desarrollo algorítmico</li> </ul> </li> <li>• Creatividad</li> <li>• Colaboración</li> <li>• Comunicación efectiva</li> <li>• Pensamiento crítico y evaluación</li> <li>• Pensamiento de diseño</li> <li>• Modelización</li> <li>• Lectura e interpretación de gráficos.</li> <li>• Recolección, visualización, manipulación y análisis de datos</li> </ul>

# Escenario

Las prácticas agropecuarias del futuro cercano implican un uso inteligente de los recursos para minimizar los costos y el impacto ambiental. Este es el eje central de la agroecología: conocer, comprender y analizar las interacciones de los diferentes componentes de un ecosistema, para optimizar recursos y maximizar rendimientos, a la vez que se procura minimizar el impacto ambiental.

Si bien es una práctica que comenzó hace casi cincuenta años, se volvió central en las últimas décadas, sumada a las nuevas tecnologías que aportan el monitoreo y el manejo inteligente de los recursos y dinamizan la práctica agropecuaria.

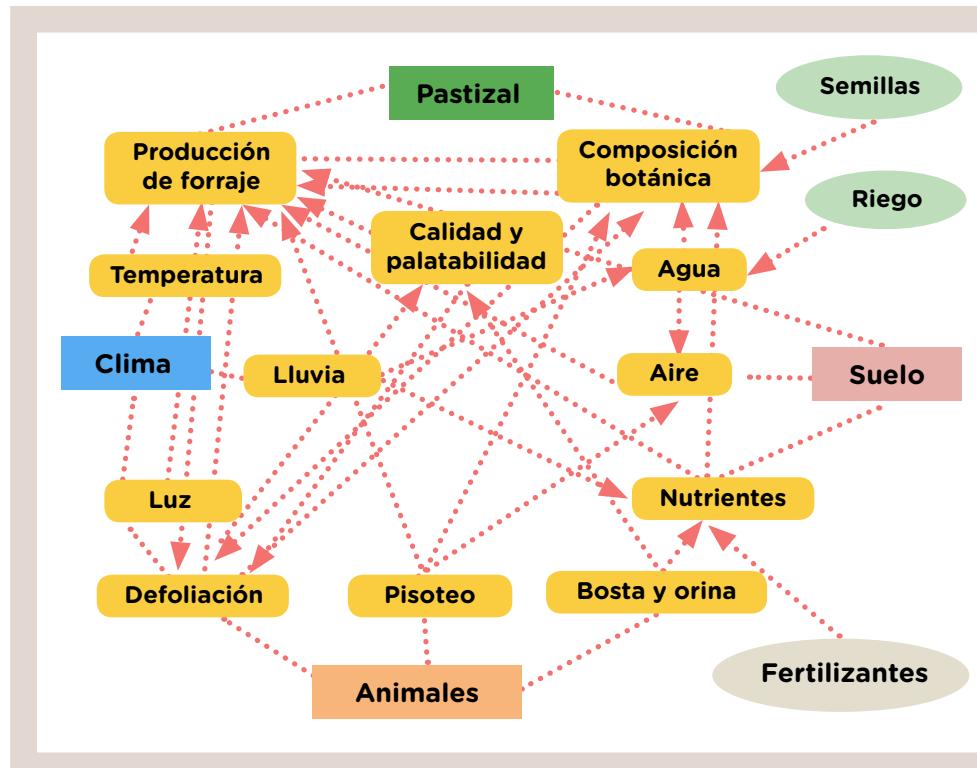
El estudio de las interacciones y la dinámica de poblaciones es aquí la clave para planificar, desarrollar y aplicar modelos de producción dentro de esta disciplina.

En la presente propuesta se toma como ejemplo la dinámica presente entre las especies vegetales y el ganado. Para esto se estudiarán las interacciones interespecíficas, así como el impacto y efecto del pastoreo y la competencia entre especies. Esto se plasmará y analizará a través del uso, la modificación y la creación de modelos de simulación computacionales que permitan formular anticipaciones y analizar posibles resultados, con el fin de evidenciar la complejidad de los sistemas agrarios y brindar herramientas a los/as estudiantes para que puedan realizar anticipaciones y diseños específicos de situaciones particulares en el ámbito rural.



- ETIQUETAS**
- [Modelización bifocal](#)
  - [Ecología agraria](#)
  - [Programación y robótica](#)
  - [Modelos de simulación computacionales](#)
  - [Interacciones interespecíficas](#)
  - [Ecosistemas](#)
  - [Sensores](#)

Comprender la complejidad de la problemática que se analiza, conocer las variables involucradas y ensayar posibles impactos de esas variables en las poblaciones de interés permite ampliar la mirada del futuro egresado o egresada, complejizar su pensamiento y brindarle herramientas indispensables para su desempeño laboral (Borrelli y Oliva, 2001).



Analizar el impacto del pastoreo, tanto los efectos positivos como los negativos, la competencia por el espacio de las especies vegetales, la selectividad de los herbívoros por esas especies vegetales, la consecuencia evolutiva de esta dinámica, la distribución de las especies en función del desplazamiento de los consumidores, la disponibilidad de recursos hídricos, entre otras, serán las variables por estudiar, aplicar y analizar a lo largo de esta secuencia, al mismo tiempo que los/as estudiantes desarrollan competencias digitales y científicas que se integran y complementan para brindar una mirada más amplia a problemáticas usualmente polarizadas.

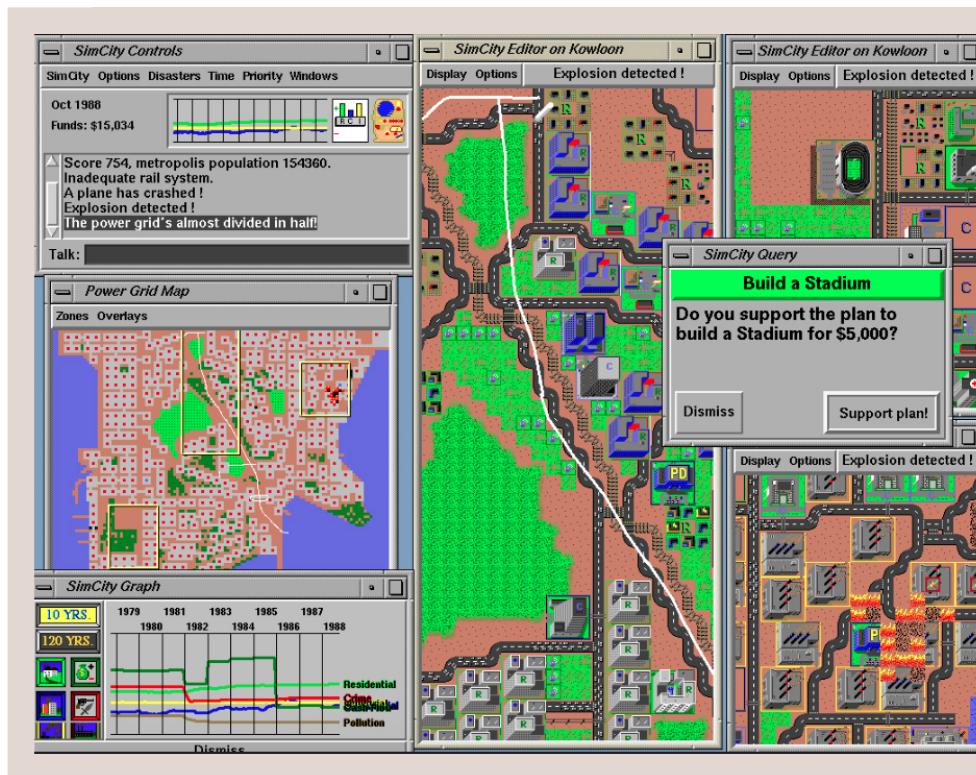
## Simulaciones computacionales y videojuegos

Uno de los videojuegos más vendidos y jugados en la historia fue [SimCity®](#), de la empresa Maxis.

El jugador o la jugadora asumía el rol de jefe/a de gobierno de una ciudad y como tal, tenía el control completo sobre ella. Debía administrar un presupuesto de dinero y ofrecer a la población servicios como edu-

cación, seguridad y salud. De pronto aparecían problemas en la ciudad y el/la jugador/a debía resolverlos haciendo obras e invirtiendo en los servicios que consideraba prioritarios.

SimCity fue un gran éxito pese a tener una característica poco habitual para un videojuego: el/la jugador/a no ganaba ni perdía, pero cada una de sus decisiones sobre las variables que representaban los servicios básicos a la ciudadanía pesaba sobre lo que ocurría. Los sucesos eran similares a los que ocurren en cualquier ciudad: por ejemplo, los/as ciudadanos/as pueden mostrarse disconformes ante determinada situación, o hay eventos como pandemias o accidentes que cambian de pronto las prioridades.



Captura de pantalla de SimCity. Autor y fuente: Xardox, Wikipedia.

En SimCity, cada uno de los pobladores y las pobladoras de la ciudad simulada toman sus decisiones de acuerdo con reglas sencillas, como por ejemplo mudarse a calles cercanas a comercios, alejarse de zonas donde hay crimen, preferir viajar en autopistas, vivir en zonas de fácil acceso. Estas decisiones son tomadas en forma individual, lo que hace que no pueda preverse con exactitud el comportamiento del conjunto y así, por ejemplo, se producen embotellamientos de tránsito, altos impuestos a las viviendas en zonas muy demandadas, crecimiento del crimen en las zonas con poca vigilancia, etcétera.

De este modo, SimCity inauguró un tipo de modelización que llamamos “basada en agentes”. En esta clase de simuladores, el/la usuario/a modifica algunas variables y el juego evoluciona de acuerdo con una

dinámica de sistemas complejos. Es decir, el sistema de pobladores/as que toman decisiones en una ciudad y afectan así las decisiones de sus vecinos/as tiene una dinámica en la que las reglas entre los/as agentes pueden ser muy sencillas, pero el comportamiento global puede mostrar sorpresas.

El concepto de simulador se basa en el hecho de que los/as usuarios/as puedan configurar algunas variables independientes (por ejemplo, el presupuesto disponible para invertir en un determinado servicio u obra en la ciudad) para ver qué sucede con otras variables dependientes (por ejemplo, el ánimo de la población o el nivel de crítica de los artículos de un diario).

Una variable se considera **independiente** cuando puede establecerse libremente sin depender de otras (siempre dentro de algunos límites), y es **dependiente** cuando su valor depende de otras variables.

A su vez, algunas de las variables independientes podían ser modificadas por el jugador. Esas variables independientes y manipulables eran las únicas que el jugador podía modificar.

En el juego SimCity se presentan muchos aspectos vinculados a la vida de las ciudades como, por ejemplo, la necesidad de realizar obras ante distintas necesidades de la ciudad. Sin embargo, en la vida real de las ciudades un jefe/a de gobierno cuenta con el apoyo de un equipo de ministros/as y asesores/as para tomar esas decisiones. Es decir, que los/as diseñadores/as de **SimCity** decidieron incluir algunos aspectos de las ciudades reales, pero en el modelo del juego eligieron dejar otros de lado.

Esta acción deliberada es lo que, en el idioma de la **modelización** (el proceso de diseñar y construir modelos), se llama **abstracción** y es uno de los pilares del pensamiento computacional, junto con la separación de un problema en partes más pequeñas, el reconocimiento de patrones y el pensamiento algorítmico.

El modelizador construye un modelo para representar una parte del mundo (objeto, fenómeno o sistema) con un propósito determinado. Y en ese intento de representación elige tomar en cuenta ciertos aspectos y dejar de lado otros.

Existen muchos tipos de **modelos** y muchas formas de clasificarlos (por su función, por el medio por el cual se representan, por la porción del mundo que representan, etcétera). Por ejemplo, tomando en cuenta la clasificación por el medio en el cual se representan los modelos, algu-

nos autores hablan de **modelos materiales**, como por ejemplo un torso humano de plástico, o de **modelos matemáticos** para representar, por ejemplo, la ley de la gravedad.

En esta secuencia didáctica se trabajará sobre **modelos de simulación computacionales**, modelos que representarán sistemas (ecosistemas, más precisamente) a través de medios digitales. Los **modelos de simulación** son una clase especial de modelos que permiten no solo explicar o comprender un fenómeno, un proceso o un evento, sino también ayudar a predecirlo.

## Materiales necesarios

En varias actividades de esta secuencia se deben programar modelos de simulación.

Para hacerlo se utilizarán dos entornos: [NetLogo](#) y [NetTango](#).

En NetLogo se programa en modo texto tipeando los comandos, mientras que en NetTango se programa de manera visual arrastrando bloques que encajan unos debajo de otros.



NetTango (programación por bloques)	NetLogo (programación por texto)
<b>Programación del comportamiento de una especie herbívora</b>	
	<pre> to moverse   fd 1 ; avanzar un paso hacia adelante   rt 10 ; girar a la derecha 10 grados end  to alimentarse   if random 100 &lt; 20 [     if pcolor = green [set energía energía + 1]   ] end </pre>

Comparación de la sintaxis entre NetTango (izquierda) y NetLogo (derecha).

En NetLogo se puede programar en línea o descargando el programa, en cambio, para programar en NetTango es imprescindible la conexión a Internet.

En algunas de las actividades de esta secuencia didáctica se deberá descargar e instalar el software NetLogo. En otras actividades será necesaria la conexión a internet.

NetLogo\* es un software libre y puede descargarse de manera gratuita. Tiene su interfaz en castellano, aunque los comandos para programarlo están en inglés. Si bien son mínimos los conocimientos de inglés necesarios, se sugiere consultar a la/el docente de Inglés. También se pueden consultar las tarjetas el [Anexo 1](#).

Para crear un programa en NetLogo o para modificar uno ya creado, pueden utilizar tanto la [versión descargable](#) como la versión web llamada [NetLogo web](#).

\* NetLogo fue creado por Uri Wilensky, de la Universidad de Northwestern, en los Estados Unidos, como derivación de la versión original de StarLogo de Mitch Resnick, creador de Scratch.



**NetLogo** es descendiente del primer lenguaje de programación desarrollado para educación: el lenguaje LOGO, de Seymour Papert, que consistía en un personaje con forma de tortuga al cual se le podía dar órdenes, como avanzar tantos pasos, girar a la derecha tantos grados e ir dibujando a medida que se movía. El lenguaje LOGO fue muy popular en las escuelas cuando comenzó a integrarse la tecnología en la década de 1980. Pueden acceder al video [“Seymour Papert presentando Logo en 1971”](#) para ver cómo era el lenguaje LOGO original.

El producto que se va a crear será un modelo de simulación computacional utilizando el mismo lenguaje, [NetLogo](#), pero con dos entornos diferentes para llevar a cabo la programación: **en modo texto y en modo visual por bloques**.

A continuación se describen los diferentes niveles de logro para el **modelo de simulación**: básico, intermedio y avanzado.

Niveles de logro		
Nivel básico	Nivel intermedio	Nivel avanzado
 <ul style="list-style-type: none"> <li>Modelo de simulación <b>programado por bloques</b> con una especie herbívora que se alimenta de A1 y A2, donde <b>el herbívoro prefiere A1 sobre A2</b>, que, además, le aporta diferentes niveles de energía. La especie se mueve, se alimenta, se reproduce y muere. La especie gana energía al alimentarse y gasta energía al moverse. Cuando llega a un nivel determinado de energía se reproduce, cuando se queda sin energía muere.</li> <li>El modelo permite variar la cantidad inicial de la especie y la proporción inicial de alimentos A1 y A2.</li> </ul>	 <ul style="list-style-type: none"> <li>Modelo de simulación en <b>NetLogo</b> de una especie herbívora que incorpora la humedad del suelo como recurso variable del cual depende la disponibilidad de A1 y A2. Hay un gráfico donde se ven los cambios en las tres poblaciones a través del tiempo.</li> </ul>	 <ul style="list-style-type: none"> <li>Modelo de simulación que incorpora modelización bifocal mediante un sensor de humedad conectado a una placa Arduino, de modo que las condiciones externas modifican el modelo. Por ejemplo, si el sensor detecta un cierto nivel de humedad, los alimentos A1 y A2 crecen a un ritmo diferente.</li> </ul>

# Itinerario de actividades

## Actividad 1. Uso y modificación de un modelo simple de interacción productores-consumidor

En la primera parte de esta actividad, se propone que los/as estudiantes usen un modelo de simulación similar al que construirán como Nivel 1 para comenzar a familiarizarse con él y con las dinámicas poblacionales y las interacciones; y que lo usen también para simular diferentes situaciones problemáticas agrarias, anticipando y analizando los resultados en las poblaciones.

En la segunda parte de esta actividad, se propone la modificación de un modelo de simulación base utilizando el lenguaje **NetLogo** para cambiar la especie o el alimento, agregar una variable independiente en forma de deslizador y un gráfico para ver el cambio en las poblaciones, y por último, experimentar los cambios realizados.

## Actividad 2. Causalidad y simulación en un ecosistema simple

En la primera parte de esta actividad, se propone la creación de un mapa causal para la construcción de un simulador para una especie que se alimenta de un único alimento, se mueve, se reproduce y muere.

En la segunda parte de esta actividad, los/as estudiantes crean un modelo de simulación en **NetTango** (programando con bloques) para una especie que se alimenta de un único alimento, se mueve, se reproduce y muere. Luego realizan experimentos con ese modelo.

## Actividad 3. Palatabilidad de los alimentos para una especie herbívora

En esta actividad se propone crear un modelo de simulación con bloques para una especie que se alimenta de A1 y A2, donde A1 y A2 le aportan diferentes niveles de energía y donde el herbívoro prefiere el que le provee mayor cantidad de energía. La especie se mueve, se alimenta, se reproduce y muere. Gana energía al alimentarse y gasta energía al moverse; cuando llega a un nivel determinado de energía se reproduce, cuando se queda sin energía muere. El modelo permite variar la cantidad inicial de la especie y la proporción de los alimentos A1 y A2.

### ETIQUETAS

Simulación  
Uso  
Modificación  
Ecosistema  
Interacciones

### ETIQUETAS

Causalidad  
Simulación  
Palatabilidad  
Creación  
Agentes

### ETIQUETAS

Simulación  
NetLogo  
Herbívoro  
Energía  
Humedad

## Actividad 4. Humedad y competencia entre especies herbáceas en el ecosistema pampeano

En la primera parte de esta actividad, se propone la creación de un modelo de simulación en NetLogo con una especie que se alimenta de A1 y A2, e incorpora la humedad como variable independiente y un gráfico para mostrar los cambios en las poblaciones.

En la segunda parte de esta actividad se usa un modelo de simulación de competencia entre especies herbáceas en un ecosistema de pastoreo.

## Actividad 5. Modelización bifocal en un ecosistema utilizando un sensor de humedad

En esta actividad se trabaja con la incorporación de un sensor de humedad conectado a una placa Arduino para realizar una modelización **bifocal** (Blikstein y Wilensky, 2010) en **NetLogo** y analizar la influencia de la humedad en el crecimiento de los alimentos A1 y A2.

### ETIQUETAS

**Humedad**  
**Gráficos**  
**Competencia**  
**Variables**  
**Pastoreo**

### ETIQUETAS

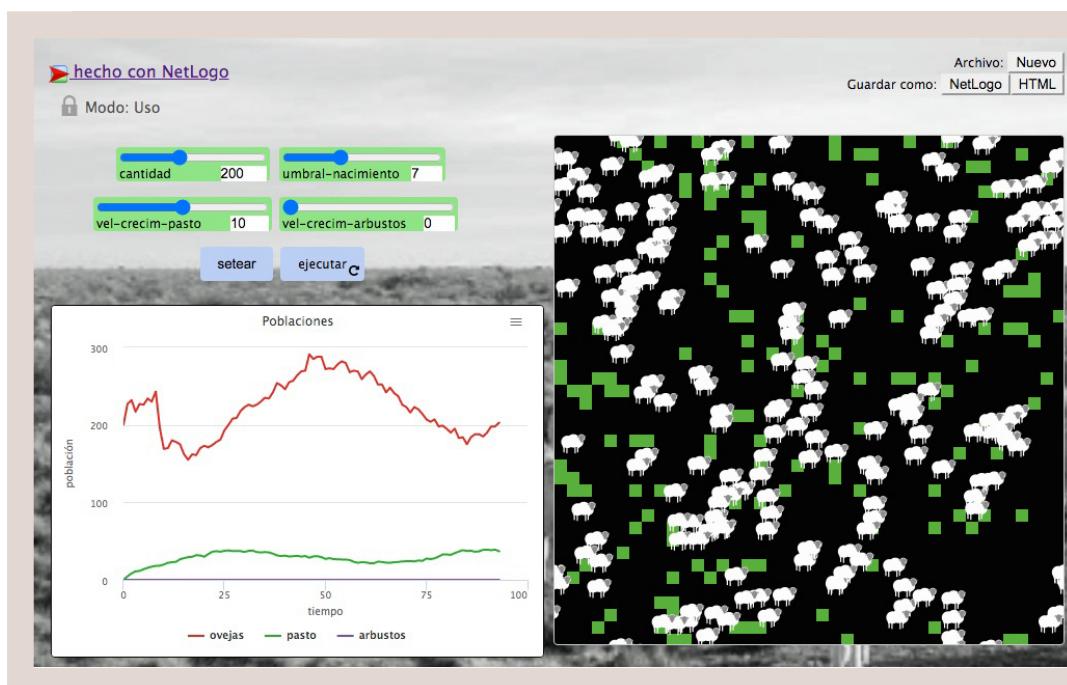
**Robótica**  
**Arduino**  
**Sensores**  
**Bifocal**  
**Humedad**

# Orientaciones para el desarrollo de las actividades

## Actividad 1. Uso y modificación de un modelo simple de interacción productores-consumidor

### Primera parte. Usar un modelo de simulación de un ecosistema simple

En esta primera parte se propone el trabajo sobre un simulador que explora un ecosistema simple conformado por ovejas, pasto y arbustos pequeños, tal como ocurre en gran parte de la estepa patagónica. Las ovejas se mueven al azar y el pasto y los arbustos también crecen al azar. La oveja come, se reproduce y muere.



Simulador de estepa patagónica con ganado ovino.

Accedan al [modelo de simulación ovejas/pastos/arbustos](#) y observen los siguientes parámetros:

- CANTIDAD controla el número inicial de ovejas.
- El deslizador UMBRAL-NACIMIENTO establece el nivel de energía que deben alcanzar las ovejas para reproducirse.
- El [deslizador](#) VEL-CRECIM-PASTO controla el ritmo al cual crece el pasto.
- El deslizador VEL-CRECIM-ARBUSTOS controla la velocidad de crecimiento de los arbustos.



Glosario

## Cómo utilizar el modelo

En primer lugar, ensayan con estos deslizadores para familiarizarse con las variables y la información que brinda la simulación en el gráfico correspondiente.

Luego establezcan el número inicial de ovejas, la velocidad de crecimiento del pasto y de los arbustos, así como el valor del umbral de nacimientos.

Den clic en el botón **Setear** para establecer esas cantidades elegidas y luego hagan clic en el botón **Ejecutar** para comenzar la simulación.

Sin detener la simulación, pueden variar los valores de los deslizadores para ensayar distintas combinaciones.

Para detener la simulación, vuelvan a seleccionar **Ejecutar**.

Pueden descargar el gráfico resultante accediendo al menú ubicado en la esquina superior derecha del gráfico. También pueden descargar los datos de la simulación en formato CSV desde el mismo menú.



Menú del gráfico para descargarlo como imagen.

## Análisis del simulador

En primer lugar, ensayan con estos deslizadores para familiarizarse con las variables y la información que brinda la simulación en el gráfico correspondiente.

Para analizar las variables y las interacciones que se presentan, respondan las siguientes preguntas:

1. ¿Qué interacciones se representan en el simulador: interespecíficas o intraespecíficas? Expliquen en qué datos se basa la respuesta.

- 2.** ¿Qué come la oveja del simulador: arbusto o pasto? ¿Cómo lo pueden averiguar?
- Ensayan con los deslizadores que consideren que pueden brindar esta información. ¿Por qué seleccionaron esos deslizadores?
  - Detallen qué diseño experimental se usó (qué variables se modificaron, cuáles se dejaron fijas, etcétera) para ponerlo a prueba. Compartan los diseños con sus compañeros y compañeras. ¿En qué se parecen y en qué se diferencian?
- 3.** En el simulador, para la población de las ovejas, ¿es indistinto comer un alimento que otro? Para responder a este interrogante, se sugiere mantener fijas las otras variables (cantidad inicial de ovejas y nacimientos) y seguir las instrucciones:
- Establecer en 10 la velocidad de crecimiento de arbustos y de pasto. Luego, ejecutar el simulador y dejar que se estabilice la población de ovejas.
  - Sin detener la simulación, bajar a 0 la velocidad de crecimiento del pasto y dejar que se equilibre la población.
  - Luego subir a 10 la velocidad de crecimiento del pasto y bajar a 0 la de arbustos. Dejar que se equilibre la población y volver a apretar el botón **Ejecutar** para pausar la simulación.
- 4.** Observen cómo reaccionó la población de ovejas en cada caso y comparen las poblaciones resultantes. Elaboren una pequeña explicación, señalando qué datos aportados por el simulador tuvieron en cuenta.

## Segunda parte. Modificar un modelo base de simulación creado con NetLogo

El modelo trabajado en la primera parte de esta actividad (ovejas, pasto y arbustos) fue creado utilizando **NetLogo**.



**Importante**

En esta segunda parte, van a modificar un modelo muy sencillo para potenciarlo y que les ayude a responder preguntas que el modelo anterior no podía.

En general, el trabajo en el aula con modelos de simulación computacionales, como los del [proyecto PhET, de la Universidad de Colorado](#), consiste en cambiar valores de ciertos parámetros variables y ver cómo responde el sistema.

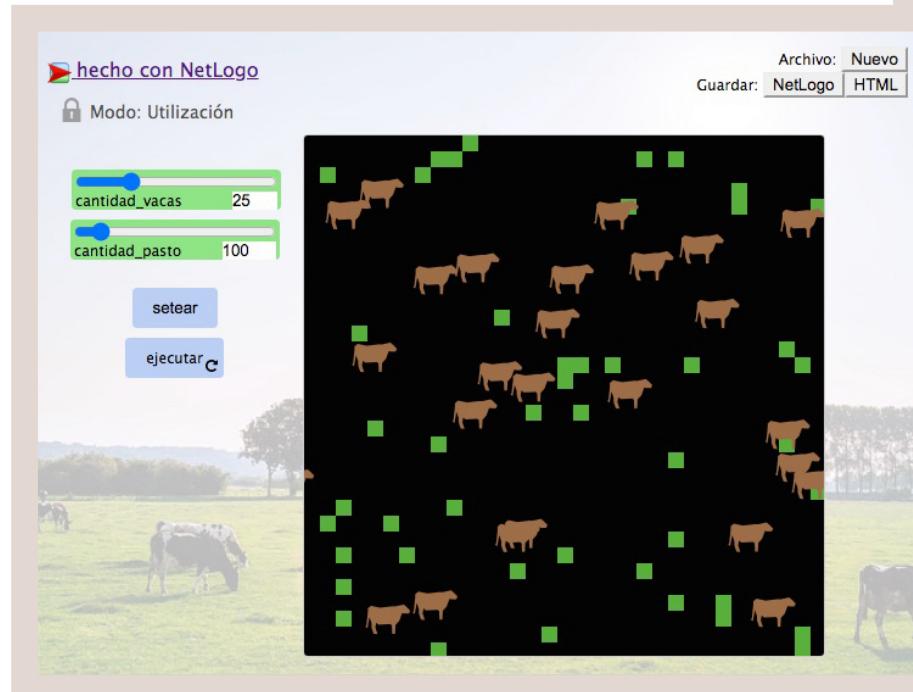
De esa manera se trabajó en la primera parte de esta actividad, donde se podía ver qué sucedía con la población de ovejas cuando, por ejemplo, se variaba la velocidad de crecimiento del pasto o de los arbustos.

Este trabajo con simuladores, en el que se **modifican las señales de entrada** (como la velocidad de crecimiento del pasto) y se **observan las señales de salida** (como el gráfico que representa el cambio en la población de ovejas), se conoce como “**caja negra**”. Se lo llama así porque no puede observarse lo que pasa entre la señal de entrada y la de salida, no puede verse cómo está construido el modelo para saber por qué cambió de esa manera el sistema.

Eso es lo que se intentará realizar en esta actividad: ir un paso más allá y “abrir la caja negra” para ver cómo está construido el simulador y agregarle nuevas características que les interesen.

**NetLogo** es un entorno de programación que nos permite ver cómo están construidos los simuladores y, además, modificarlos. Se podría decir que es como una caja de cristal, donde se puede mirar adentro.

Exploren el siguiente [modelo de pastoreo de ganado vacuno](#) durante unos breves minutos para luego avanzar al paso siguiente, que es la decodificación del modelo.



Modelo de simulación básico de pastoreo vacuno.

## Analizar cómo está construido el modelo de pastoreo

El proceso de analizar cómo está construido un programa —en este caso, un modelo de simulación— en cualquier lenguaje de programación se conoce como “decodificación”. Es un proceso donde se explica y se analiza paso a paso cada línea del programa.

Se sugiere, en este punto, consultar el tutorial “Cómo decodificar el modelo de pastoreo” en el [Anexo 2](#) para analizar paso a paso este modelo de pastoreo.



Luego de leer el documento sobre cómo decodificar el modelo de pastoreo, se sugiere volver a la interfaz gráfica del modelo para ejecutarlo, tratar de interpretar cómo los comandos analizados se van ejecutando, cuáles son más palpables y cuáles están ocultos a los ojos del usuario o de la usuaria que interactúa con el simulador.

## Preguntas al modelo de pastoreo

En la primera parte analizaron un modelo de pastoreo ovino, donde las ovejas tenían a disposición dos especies vegetales. En esta parte de la actividad, modificarán un simulador partiendo de un modelo más sencillo que se irá complejizando, para luego realizar un análisis similar.

Utilicen el simulador [modelo base-pastoreo](#) para contestar las siguientes preguntas. Si necesitan ayuda sobre cómo utilizar el simulador, pueden consultar la Guía de Ayuda en la parte inferior del simulador.

### Para observar

¿Cómo es el movimiento de las vacas? ¿Varía la cantidad de vacas con el tiempo? ¿Siempre que pasan sobre una parcela con pasto lo comen? ¿Qué cosas de este modelo no son como en la realidad? ¿Tiene sentido que sea así?

### Para experimentar

¿Qué pasa con el pasto a medida que transcurre la simulación? ¿Hay alguna relación entre la cantidad de ganado y el tiempo que tarda el pasto en agotarse? ¿Qué experimento podrían diseñar con este simulador para contestar estas preguntas?

¿Qué cosas se podrían incluir o mejorar en este modelo para representar otros aspectos del mismo fenómeno de pastoreo de ganado vacuno?



De aquí en adelante realizarán una serie de mejoras al modelo de pastoreo, de manera guiada. Si quisieran trabajar por su cuenta e intentar mejoras sin seguir la guía a continuación, se recomienda consultar los siguientes recursos:

- [Diccionario de NetLogo](#) (en español), en el sitio de la Universidad de Northwestern.
- [Guía de programación de NetLogo](#) (en español), en el sitio de la Universidad de Northwestern.
- [Introducción a NetLogo](#) (en español), en el sitio de la Universidad de Northwestern.
- [Tutorial de NetLogo](#) (en español), en el sitio de la Universidad de Northwestern.

## Extender el modelo

Pueden modificar un programa en NetLogo vía web desde el modelo mismo (no disponible en algunos casos) o a través de la versión descargable de NetLogo.

Para conocer cómo es ese procedimiento, se sugiere ver el videotutorial [“Cómo modificar un programa en NetLogo”](#).



### 1. Ejercicio no guiado, nivel 1: Reemplazar

Para realizar estos cambios solamente hay que sustituir unas palabras por otras. Deben reemplazarse las vacas por otro herbívoro, como la oveja (o cambiar la raza y la forma); achicar un poco el tamaño de los agentes-ovejas (cambiar de 3 a 2.5); cambiarle el color (de marrón a blanco); hacer que el pasto se pinte de un color más clarito, como el verde lima (los colores pueden expresarse también como números, el

color verde lima es el número 65), y realizar la modificación de que cuando las ovejas comen el pasto, que el terreno quede de color marrón en vez de negro. Además, pueden hacer que los giros que realizan los agentes al moverse sean más amplios (que sean de hasta 90 grados en vez de hasta 5).

## 2. Ejercicio no guiado, nivel 2: *Copiar y pegar*

Luego de haber reemplazado vacas por ovejas, agreguen nuevamente vacas para tener dos especies pastoreando. Creen la misma cantidad de ovejas que de vacas y hagan que las vacas se muevan de manera similar a las ovejas.

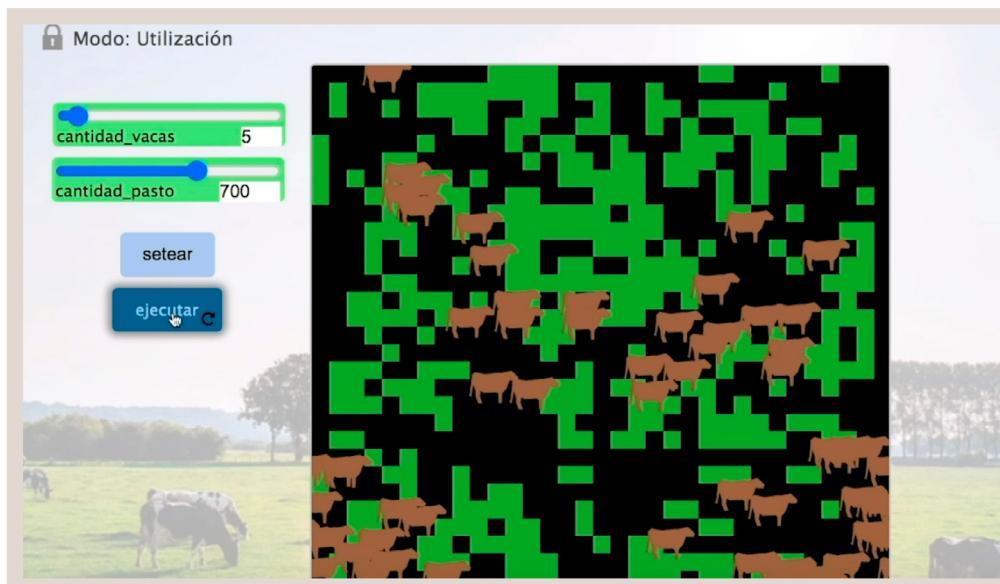
### Ayuda

Una forma de lograrlo es renombrar el procedimiento original “moverse” como “moverse-ovejas” y luego copiarlo y pegarlo y renombrar la copia como “moverse-vacas”. Los procedimientos “comer-pasto” y “dar-un-paso” no deben ser copiados ni renombrados, para lograr que se muevan ambas razas de agentes. ¿Se mueven igual las vacas que las ovejas?



## 3. Ejercicio guiado, nivel 3: *Inventar*

En los ecosistemas reales, los animales necesitan energía para vivir. Cuando se alimentan, ganan energía que van utilizando (o sea, perdiendo) para diferentes tareas, como desplazarse o reproducirse. En este ejercicio guiado se verá cómo incorporar estas características al modelo base. Una vez realizadas estas modificaciones, guardar el archivo en formato .nlogo con el nombre **pastoreo-mejorado.nlogo**, para utilizarlo en la **actividad 4**. Para ver cómo realizar estas modificaciones, se recomienda mirar el videotutorial [“Cómo modificar el modelo base de pastoreo”](#).



## Actividad 2. Causalidad y simulación en un ecosistema simple

### Primera parte. Creación de un mapa causal

La propuesta es crear un organizador gráfico dinámico que permita mapear las relaciones que se establecen en un ecosistema simple de un herbívoro que se mueve, se alimenta, se reproduce y muere, y que necesita energía para estos procesos.

[SageModeler](#) es una herramienta gratuita, disponible en idioma español, para crear diagramas conocidos como *mapas causales*, que ilustran las relaciones entre los factores interviniéntes en un problema, proceso o evento que se quiere explorar.

La instancia de crear un [mapa causal](#) antes de crear el simulador ayudará a estructurar el proceso para la segunda parte de esta actividad.



En [Mapa causal de pastoreo](#) pueden acceder a un modelo que contiene una serie de factores pero que no contiene las relaciones. Establezcan las relaciones entre los factores existentes siguiendo la [guía de ayuda](#) para la confección de un mapa causal. También pueden acceder a la guía de ayuda desde el interior del propio documento.

Luego de establecer las relaciones entre los factores que se encuentran en el archivo de SageModeler, se sugiere realizar una captura de pantalla del mapa a modo de referencia para realizar la segunda parte de esta actividad. El mapa causal ayudará a identificar los procesos que se deben modelizar para crear el modelo de simulación con bloques.

### Segunda parte. Creación de un modelo de simulación programando con bloques

NetTango es un entorno de programación por bloques que genera código de NetLogo. Esto permite hacer más accesible la programación en este entorno.

Pueden acceder al [modelo para programar por bloques una especie herbívora](#) del entorno NetTango.



Modelo de simulación programable de una especie herbívora.

**1. Área de programación del simulador.** Arrastrar los bloques ubicados en el lateral derecho para programar los cuatro eventos de color naranja. Crear un modelo de simulación para un animal herbívoro que se alimenta de A1 y A2, en el cual A1 le provee más energía que A2. Cuando el herbívoro come se produce defoliación. Modelizar esta situación utilizando el bloque “Defoliar”. En este modelo, el suelo desprovisto de vegetación se representa con el color marrón. Una vez colocados los bloques, no olvidar dar clic en el botón **Recompilar**.

**2. Área de simulación.** Configurar los valores para la velocidad de regeneración de alimento, la cantidad disponible de los alimentos A1 y A2 y la población inicial del herbívoro. Una vez configurados estos valores, ejecutar el simulador cliqueando en el botón **Setear** y luego en **Ejecutar**.

Analicen lo que ocurre con el herbívoro y lo que se observa en el gráfico en el simulado, teniendo en cuenta los siguientes interrogantes:

- ¿Son esperables esos resultados? ¿Por qué?
- ¿Hay algo que esperarían ver y no están viendo?
- ¿Qué sucede con la población de herbívoros con el tiempo?
- ¿Qué valores utilizaron para la regeneración de alimentos? ¿Por qué?
- ¿Qué situaciones podrían modelizar utilizando valores altos, medios o bajos de la regeneración de alimentos?

## Actividad 3. Palatabilidad de los alimentos para una especie herbívora

En la actividad anterior, trabajaron con un modelo donde una población de herbívoros interactuaba con las poblaciones vegetales, y analizaron cómo estas fueron cambiando en respuesta a esa interacción. Las redes tróficas representan un modelo simplificado de las relaciones de alimentación y de flujo de energía en un ecosistema.

Pero en la naturaleza, estas relaciones no son lineales. Los consumidores responden de forma diferente a las diversas poblaciones con las que interactúan. Esto suma complejidad a las interacciones y tiene consecuencias evolutivas sobre el ecosistema.

Los herbívoros, por ejemplo, son selectivos con su alimento. Cada población de plantas disponibles brindará un conjunto de nutrientes y una cantidad de energía diferente. Este aspecto es considerado por los consumidores, quienes van cambiando su consumo en función de la respuesta placentera que se produce en sus organismos.

La **palatabilidad** es un concepto que se podría definir como “el placer [...] que un animal experimenta al consumir un determinado alimento o fluido” (Church, 1979). Existe evidencia de que los animales pueden aprender a complementar sus dietas con alimentos de distinta palatabilidad (Distel y Villalba, 2007).

### Reprogramación del modelo

Se propone reprogramar el simulador usando NetTango (que permite hacerlo **cambiando los bloques**) para mostrar la preferencia del herbívoro por uno de los dos alimentos, que no sea lo mismo comer uno u otro (uno es más apetecible que el otro), y que ambos le proporcionen diferente cantidad de energía y que esta sea poca.

Una vez reprogramado el simulador es importante probar si lo que se programó realmente funciona. Estas preguntas pueden ayudar a ese objetivo:

- ¿Qué experimento se podría diseñar para comprobar que el simulador funciona efectivamente de esa manera?
- ¿Qué se espera que suceda con la población de herbívoros a lo largo del tiempo?

## Escenarios

Realicen experimentos con el simulador cambiando los bloques, para probar uno de los escenarios que se indican a continuación, y describan qué ocurre con la población de herbívoros a lo largo del tiempo:

- Un virus afecta la reproducción del herbíboro (pista: usar el bloque **probabilidad** en el procedimiento **reproducirse**).
- Hay un agente tóxico en el ambiente, el cual afecta el apetito del herbíboro (pista: usar el bloque **probabilidad** en el procedimiento **alimentarse**).
- Inventar un escenario propio.

## Actividad 4. Humedad y competencia entre especies herbáceas en el ecosistema pampeano

Existe otro factor en la dinámica de poblaciones que puede ser determinante a la hora de disponer de alimento para el ganado: la competencia entre poblaciones vegetales. Los recursos disponibles en un ambiente serán determinantes para favorecer o no la competencia de algunas especies. Cada especie, a su vez, requerirá mayor o menor cantidad de recursos ambientales disponibles como, por ejemplo, el agua en el suelo.

Las especies vegetales llamadas **mesofíticas**, de hojas anchas y membranosas, son poco tolerantes a la sequía y cuando se encuentran en períodos de poca humedad (estrés hídrico) su crecimiento se reduce (Couso y Fernández, 2012). Pero, por otro lado, cuando la cantidad de agua disponible es óptima, estas especies suelen tener una alta capacidad para ocupar el espacio disponible, pues su tasa de crecimiento vegetativo relativo (reproducción asexual) es alta.



Por otra parte, las especies **xerofíticas** son más tolerantes a la sequía y se caracterizan por tener hojas más pequeñas, contener espinas o pequeños “pelos” que les proveen protección contra la pérdida de humedad a través de sus partes verdes. Estas plantas suelen tener una tasa de crecimiento más lenta, tanto en producción de semillas (reproducción sexual) como a través de sus raíces (reproducción asexual).

## Primera parte. Crear un modelo de simulación en NetLogo con una especie que se alimenta de A1 y A2, e incorporar la humedad como variable independiente y un gráfico para mostrar los cambios en las poblaciones

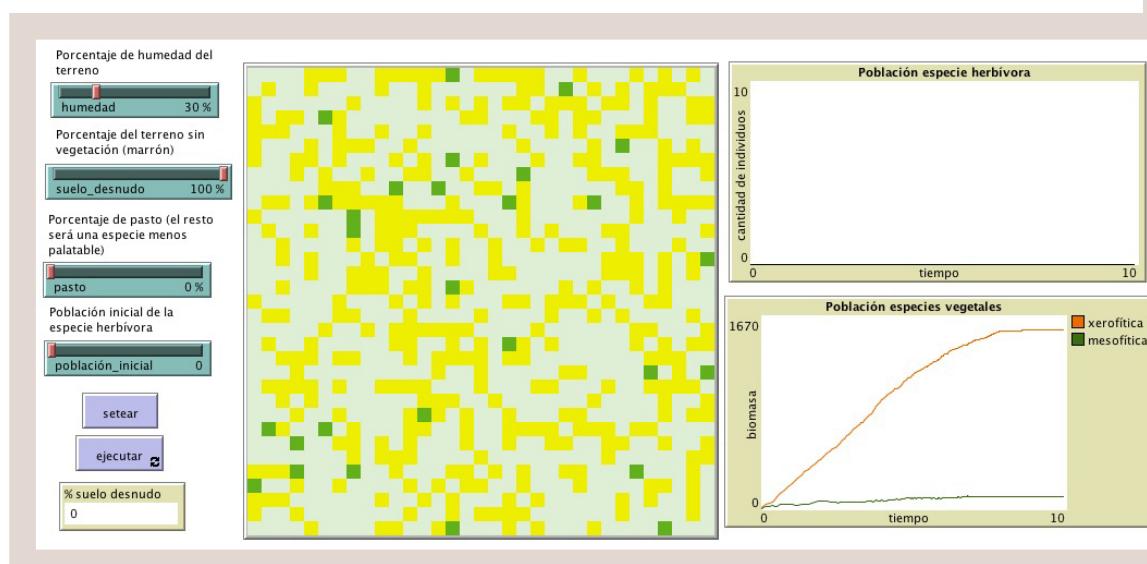
Para llevar a cabo esta parte de la actividad, sigan estos pasos:

1. Abrir la versión descargable de NetLogo.
2. Recuperar el archivo **pastoreo-mejorado.nlogo** realizado en la segunda parte de la actividad 1.
3. Seguir las indicaciones del [tutorial actividad 4, parte 1](#).
4. Guardar el archivo con las modificaciones realizadas con el nombre **pastoreo-final.nlogo**. Este archivo es el producto solicitado como nivel de logro intermedio.
5. Guardar el archivo con formato NetLogo web (HTML) y abrirlo con un navegador. Probar el modelo. Subirlo a un sitio al que se pueda acceder (si cuentan con uno).
6. Comparar la ejecución desde dentro de NetLogo y desde un navegador. ¿Se observa alguna diferencia?



## Segunda parte. Usar un modelo de simulación de competencia entre especies herbáceas en un ecosistema de pastoreo

Se utilizará un modelo similar al creado en la primera parte de esta actividad, con algunas características especiales. A continuación se muestra una captura de pantalla del [modelo de simulación El caldenal](#).



Modelo de competencia entre una especie mesofítica y una especie xerofítica.

En este modelo se puede analizar no solo el efecto del pastoreo, sino también las diversas respuestas que se dan entre las poblaciones vegetales según la **humedad** del suelo. Se trabajará en dos etapas:

### 1. Sin pastoreo

Colocar como condiciones iniciales fijas: **Suelo desnudo, 100% / Población inicial, 0**. En los siguientes escenarios en un sistema agrario, relacionados con la humedad del suelo: a) Estrés hídrico; b) Humedad media; c) Lluvias abundantes.

- Piensen y respondan: ¿Qué población se espera que aumente? ¿Qué población no podrá reproducirse? ¿Por qué?
- Determinen el nivel de humedad que corresponde en cada escenario. Usen la simulación y cotejen las respuestas del punto anterior con lo observado en el simulador utilizando los gráficos resultantes.

### 2. Con pastoreo

Colocar como condiciones iniciales fijas: **Suelo desnudo, 50% / Población inicial, 20**.

Ensayan con el simulador los escenarios del punto anterior insertando ahora la población herbívora. Observen qué ocurre con las tres poblaciones que interactúan.

- ¿En qué condiciones se logra mantener una población estable de herbívoros?
- ¿Cómo cambia la respuesta de la población de herbívoros a medida que se aumenta el porcentaje inicial de mesófitas? ¿A qué puede deberse esto?
- Justifiquen sus respuestas, basándose en los gráficos que resultan en cada simulación.

## Actividad 5. Modelización bifocal en un ecosistema utilizando un sensor de humedad

En esta actividad se trabajará con la versión descargable de NetLogo para conectarla a un sensor de humedad con una placa Arduino.

Debido a una incompatibilidad entre la versión de NetLogo 6.2.0 para Windows 10 de 64 bits y la placa Arduino, si se utiliza dicha versión del sistema operativo (Win 10 64 bits), entonces se debe descargar la [versión 6.2.0 de NetLogo para 32 bits](#).



## Influencia del agua en especies herbáceas y arbustivas del caldenal

Uno de los ecosistemas más estudiados del país sobre las diversas formas de aprovechamiento de la **humedad del suelo** que hace cada población es el del caldenal pampeano<sup>1</sup>.

En esta actividad, se propone aplicar el enfoque de la **modelización bifocal** (Blikstein y Wilensky, 2010), es decir, que lo que ocurra en el mundo físico, por ejemplo con alguna variable que mida un sensor, tenga una correspondencia con lo que se ve en la pantalla del simulador.

La propuesta consiste en trabajar con una [placa robótica Arduino](#) y un [sensor de humedad](#) y probar de qué manera influye la humedad del suelo en el crecimiento de las especies, y cómo impacta esto en la población de ganado vacuno.

La actividad se dividirá en tres etapas:

- Modificación del modelo en NetLogo para que se comunique con Arduino.
- Conexión de NetLogo con Arduino y testeo del sensor.
- Modelización bifocal para simular escenarios diferentes de humedad medidos por el sensor.

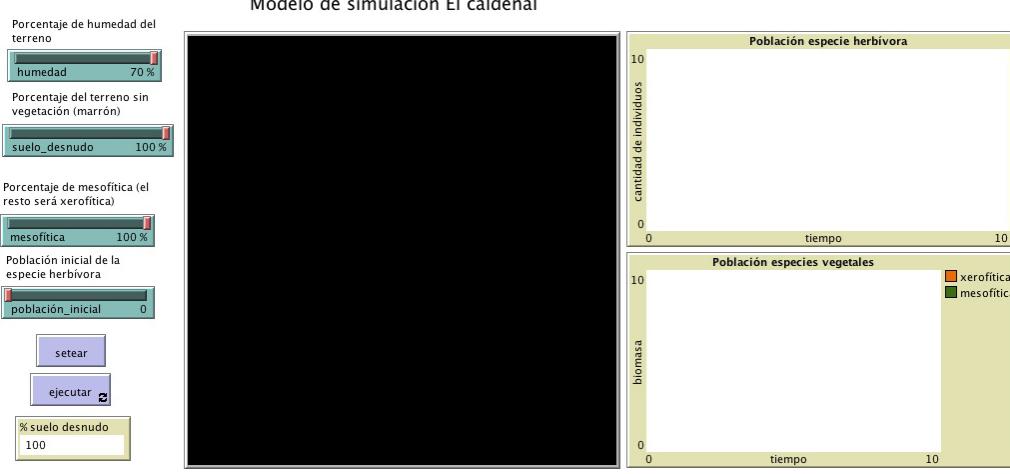
### Modificación del modelo en NetLogo para que se comunique con Arduino

1. Bajar el [software NetLogo](#) a sus computadoras. NetLogo es multiplataforma, es decir, puede instalarse en cualquiera de los sistemas operativos más comunes de la actualidad, tanto en GNU/Linux como en Windows o Mac OS X.
2. Luego de haber bajado el software NetLogo, descargar el archivo [caldenal.nlogo](#).
3. Instalar NetLogo.
4. Ejecutar NetLogo y abrir el archivo **caldenal.nlogo** recientemente bajado. Al abrirlo, debería verse una interfaz similar a la que se muestra a continuación:

<sup>1</sup> El caldenal pampeano es un área geográfica de la Argentina donde predomina el caldén, un árbol endémico de nuestro país, considerado patrimonio natural de La Pampa. El caldén es un árbol de corteza gruesa y rugosa que puede crecer en suelos arenosos y que, además, resiste la sequía.



Glosario



Modelo NetLogo de El caldenal.



Tutorial

- Este modelo es similar al trabajado en la [segunda parte de la actividad 1](#) cuando, a partir de un modelo base de pastoreo, se propuso la modificación para contemplar otras características de este tipo de ecosistemas (pueden volver a ver el videotutorial [“Cómo modificar el modelo base de pastoreo”](#)).
- Para completar los pasos restantes, se sugiere consultar el tutorial [“Conectar modelo de NetLogo El caldenal con Arduino”](#).



- El modelo resultante de esta modificación será el que se utilizará a continuación, conectado a la placa Arduino con el sensor de humedad.

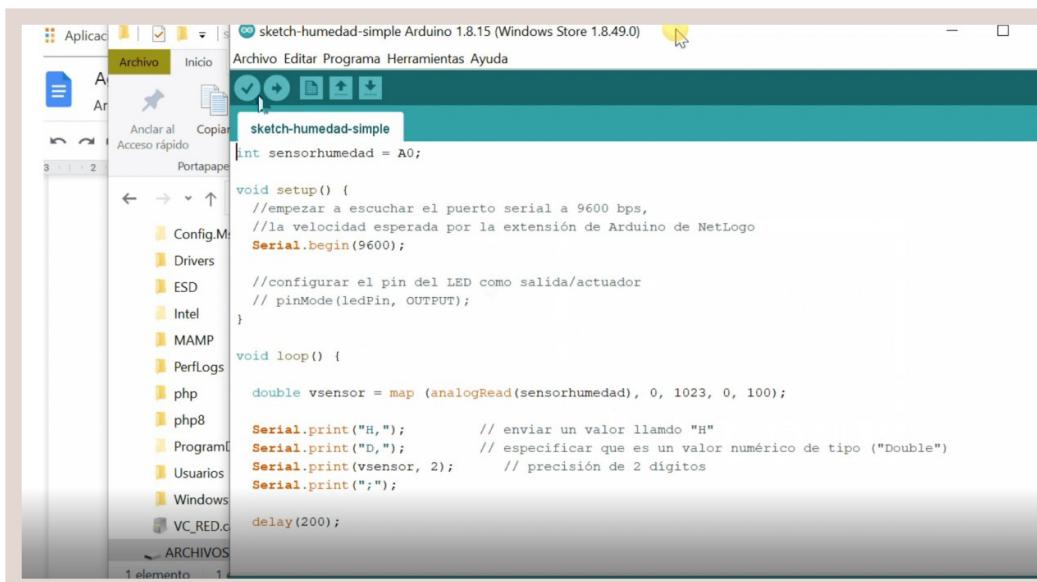
## Conexión de NetLogo con Arduino y testeo del sensor

Arduino puede conectarse a **NetLogo** a través del cable USB que viene en el kit Arduino, y esa conexión le servirá para darle energía a la placa y para enviar los datos que el sensor de humedad vaya leyendo. Para que esa comunicación entre **NetLogo** y **Arduino** exista, se debe utilizar un pequeño programa que se baje a la placa Arduino para programarla y ordenarle que lea los datos del sensor de humedad.

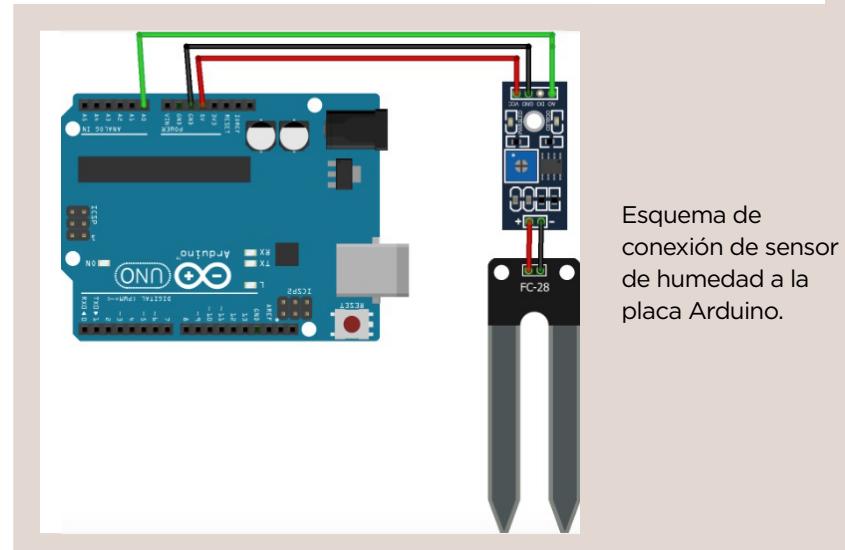


Importante

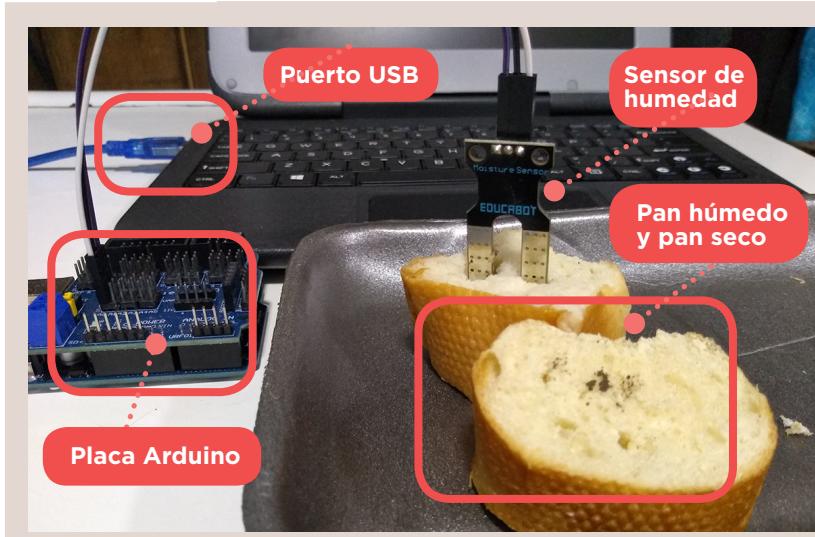
1. Descargar el [software IDE de Arduino](#).
2. Descargar el archivo [sketch-humedad-simple.ino](#).
3. Abrir este archivo (sketch-humedad-simple.ino) en el IDE de Arduino y transferirlo a la placa. Para abrirlo, lo más conveniente es hacer doble clic directamente en el archivo, para que se abra dentro del IDE. Pueden consultar el videotutorial [“Cómo pasar un INO al Arduino”](#) (sin sonido).



4. Conectar el sensor de humedad a la placa Arduino de la siguiente manera:



5. Conectar la placa Arduino a la computadora a través del puerto USB y el sensor de humedad a dos recipientes, uno con tierra húmeda y otro con tierra seca, o a cualquier otro material que permita introducir la sonda del sensor (los dos terminales con punta). Como se muestra en la imagen, se utilizaron trozos de pan seco y pan húmedo para simular diferentes condiciones de humedad:



Escenario de prueba para dos niveles diferentes de humedad con sensor conectado a una placa Arduino.

6. Ejecutar **NetLogo** (si se está usando Windows 10 de 64 bits, bajar la versión de [NetLogo 6.2 de 32 bits](#)), abrir el archivo recientemente modificado y renombrado como **caldenal-con-arduino.nlogo**.
7. Comenzar con el sensor al aire, cliquear en el botón **Setear** y luego en el botón **Ejecutar** y observar el valor del monitor “humedad”; debería marcar el valor “1”, ya que el sensor está en el aire.
8. Colocar el sensor en el material elegido (en el caso de la imagen de arriba es un trozo de pan) y verificar si cambia el valor del monitor “humedad”.

9. Si marca un valor razonable —digamos superior al 30%—, dependiendo del medio en que se coloque, entonces el modelo ya está listo para probar diferentes escenarios en la etapa siguiente.

## Modelización bifocal

Para esta etapa se va a utilizar el escenario de prueba anterior, es decir, los dos recipientes con tierra o dos trozos de pan, para simular dos condiciones diferentes de humedad.

Una vez que esté todo conectado y el simulador en pantalla, realicen dos experimentos con dos niveles diferentes de humedad, uno seco y otro húmedo, de la siguiente manera:

1. Establecer las condiciones iniciales de un ambiente con una población de ganado inicial moderada (**50 individuos**), en un terreno con 50% de suelo desnudo y con el deslizador de la especie mesofítica en 50% (lo cual significa **25% de mesofítica y 25% de xerofítica**). Cabe que el porcentaje de mesofítica es sobre el área con vegetación, no sobre el área total. En este caso, al haber 50% de suelo desnudo, solo queda el otro 50% con vegetación. Por eso el elegir el 50% de mesofítica resulta en un 25% del total del terreno de mesofítica y 25% de xerofítica (porque es la mitad de la mitad).
2. Simular también qué sucede con la población de vacas y con la biomasa de las especies vegetales en situación de sequía (**30% de humedad**).\* ¿Cómo se podrá explicar el resultado?
3. Repetir el paso anterior, pero en un escenario de lluvias abundantes (**humedad 70%**).
4. Repetir los escenarios anteriores, pero en situación de sobrepastoreo (**cantidad inicial de ganado de 100 individuos**). ¿Qué sucede con las especies vegetales en una situación de sobrepastoreo con sequía? ¿Y con lluvias? ¿Es diferente la situación? ¿Cómo se podría explicar?



\* Como los valores de la humedad provendrán de la medición del sensor conectado a la placa Arduino, deben ser aproximados a los sugeridos.

# Orientaciones generales para la enseñanza y la evaluación

Esta secuencia didáctica se nutre de múltiples aportes y experiencias. Por un lado, los trabajos de investigación del profesor Mitch Resnick, del Instituto de Tecnología de Massachusetts (MIT, por sus siglas en inglés), conocido por ser el creador del entorno de programación Scratch que, entre muchos otros avances, advierte en su libro *Tortugas, termitas y atascos de tráfico* (2001) que los modelos de simulación computacionales creados con StarLogo (NetLogo en nuestro caso) sirven más para la **estimulación** (de la mente) que para la **simulación** (de los eventos) propiamente dicha. Su idea es que los modelos de simulación computacionales sirvan para ver cómo las personas (en este caso, los/as estudiantes) piensan acerca de las criaturas o eventos que modelizan (ovejas, lobos, vacas, pasto, ardillas, poblaciones, competencia entre especies, etcétera), convirtiéndose de esta manera en “objetos para pensar”.

Por otro lado, la modificación y creación de simulaciones computacionales mediante el trabajo en grupos de estudiantes ofrece una excelente oportunidad para la negociación entre pares de los significados involucrados en la descripción del fenómeno por modelar, sus conexiones causales y su particular dinámica. Esta interacción entre pares permite consensuar la forma en que se concibe el fenómeno y, de ese modo, cada estudiante puede ajustar sus propias representaciones mentales de esa parte de la naturaleza. Este tipo de sincronización entre pares, que tiene su paralelo en las discusiones en el seno de las comunidades científicas, es fundamental para la comprensión profunda de los fenómenos naturales. El consenso que se logre en la descripción es un paso ineludible previo a toda decisión de intervención dirigida a lograr modificaciones en el entorno.

## Programación por pares

Es una estrategia didáctica en la cual se trabaja parecen parejas con roles que se van invirtiendo cada determinado lapso de tiempo. Uno de los miembros cumple el rol de **conductor** y es quien teclea los comandos y **maneja** la computadora. El otro cumple el rol de **GPS** y es quien **orienta** al conductor, sugiriendo tal o cual comando, instrucción, operación, etcétera. Por ejemplo, a la hora de seguir un tutorial, el GPS será quien lo lea y le vaya indicando al conductor qué hay que ir haciendo, y el conductor lo irá ejecutando. Luego de un tiempo, se invierten los roles.

## Estocasticidad en NetLogo y NetTango

Los modelos de simulación que se trabajan en esta secuencia utilizan el abordaje de modelización basada en agentes. Este abordaje permite una mirada compleja sobre los fenómenos y procesos modelizados, pero a la vez, accesible para los/as estudiantes, ya que se utilizan operaciones algebraicas sencillas, en vez de ecuaciones diferenciales como en el caso del modelado clásico.

La modelización basada en agentes establece reglas sencillas sobre el comportamiento de los agentes e incorpora cierto nivel de aleatoriedad. Esto resulta en un **modelo estocástico**, donde a pesar de setear los mismos valores de entrada, los resultados obtenidos no son siempre iguales (como sucede en los modelos **deterministas**).

En los modelos deterministas, se establecen propiedades a un conjunto de individuos (por ejemplo, la tasa de reproducción de las ovejas), mientras que en los modelos basados en agentes se establecen reglas sencillas de interacción entre los individuos entre sí y entre los individuos y el ambiente. Estas reglas son a nivel de cada individuo, no de la población. Las propiedades de la población, de este modo, emergen como resultado de esas interacciones sencillas.

## Uso de los datos generados en la simulaciones

En todos los modelos de simulación trabajados en esta secuencia es posible descargar una imagen en diferentes formatos (PNG, JPG, SVG) con la captura de pantalla del gráfico, y también descargar los datos generados en formato CSV para su posterior tratamiento con una planilla de cálculos. Estas operaciones se realizan a través del menú del gráfico, como muestra la imagen.



## Soluciones posibles

Se sugiere proponer a los/as estudiantes que recorran la secuencia tal como se plantea sin revelar (al menos en principio) las soluciones posibles, para evitar que concreten el desafío como si fuera una receta. El camino que intencionalmente se propone que recorran promueve el desarrollo de habilidades como la descomposición de un problema en sus partes, la detección y solución de errores, el desarrollo del pensamiento algorítmico, el pensamiento lateral y la creatividad, entre otras.

Para orientar la tarea de la/el docente se describen en los siguientes enlaces las **posibles soluciones** a las actividades y desafíos planteados durante la secuencia. Es importante remarcar que son **posibles** soluciones, ya que uno de los objetivos de esta secuencia es transmitir la idea de que hay muchas formas de resolver un mismo problema, que no existe una única respuesta correcta.

### Actividad 1, segunda parte. Modificación del modelo base de pastoreo

Pueden acceder y descargar un posible modelo [solucion-A1-P2 en formato HTLM](#) (se abre con un navegador) y [solucion-A1-P2 en formato nlogo](#) (se abre con el programa NetLogo)\*.



\* Una vez descargados los archivos, ninguno de los dos formatos necesita conexión a internet para ejecutar los modelos de simulación.

### Actividad 2, primera parte. Mapa causal de una especie herbívora con SageModeler

Pueden acceder a un posible modelo de [mapa causal](#) (se requiere conexión a internet).

### Actividad 2, segunda parte. Modelo de simulación con NetTango

Pueden acceder a un posible modelo de [especie herbívora para ntango](#).



\*\* Para poder plasmar esta solución, deben utilizar el [modelo para programar por bloques una especie herbívora](#).

### Actividad 3. Modelo de simulación con NetTango sobre palatabilidad

Pueden acceder a [una captura de pantalla](#) de una solución posible\*\*.

### Actividad 4, primera parte. Modelo de simulación en NetLogo con deslizador de humedad y gráfico

Pueden descargar un [modelo en NetLogo](#) como posible solución.

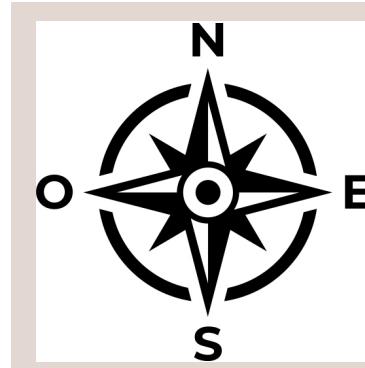
### Actividad 5, primera parte. Modelo de simulación en NetLogo modificado para tomar los datos del sensor de humedad conectado a placa Arduino

Pueden acceder a un posible [modelo de solución](#).

## Rutina de pensamiento

Para encauzar el desarrollo de esta propuesta se puede aplicar una rutina de pensamiento denominada **puntos de la brújula**, que permite explorar varias facetas de una propuesta y ayuda a los/as estudiantes a desarrollarla.

Esta rutina consiste en dibujar la base de una brújula en el pizarrón con las letras iniciales de los cuatro puntos cardinales: E, O, N y S, es decir, una rosa de los vientos simplificada. Cada letra (o dirección) representa respectivamente los siguientes estados: **E = Entusiasmo**, **O = Obstáculo**, **N = Necesidad de saber**, **S = Sugerencias**.



Brújula.

A la vez, cada dirección incluye preguntas que el/la docente hace a los/as estudiantes, cuyas respuestas se registran para obtener una ayuda visual. Las preguntas son las siguientes:

**E = Entusiasmo.** ¿Qué te entusiasma de esta propuesta? ¿Cuál es su ventaja?

**O = Obstáculo/preocupación.** ¿Qué encontrás como obstáculo o preocupación de esta propuesta? ¿Cuál es su desventaja?

**N = Necesidad de saber.** ¿Qué más necesitás saber o descubrir acerca de esta propuesta? ¿Qué información adicional te ayudaría a desarrollar esta propuesta?

**S = Sugerencias** o postura para avanzar. ¿Cuál es tu sugerencia, postura u opinión actual con respecto a esta propuesta? ¿Cuál debe ser el siguiente paso para desarrollar esta propuesta?

Esta rutina de pensamiento fue desarrollada por el Proyecto Zero, de la Escuela de Posgrado en Educación de la Universidad de Harvard (Ritchhart *et al.*, 2014; Ritchhart y Church, 2020).

En esta secuencia la rutina se puede utilizar en una o más instancias, por ejemplo cuando se propone a los/as estudiantes crear los modelos. Son instancias en las que pueden manifestar claramente lo que forma parte de su entusiasmo y de su obstáculo (direcciones E y O de la rutina). Sin embargo, es probable que necesiten conocer más acerca del comportamiento de las especies que seleccionen para construir esos modelos, o del modo en que otras variables —como la humedad— operan sobre las primeras (dirección N de la rutina).

Es precisamente allí cuando la *expertise* de la/el docente juega un rol valioso, estratégico, para comunicar a los/as estudiantes saberes, que los/as asistan en el progreso de la secuencia.

La rutina desarrolla una atmósfera profundamente inclusiva; por un lado, el/la docente aprecia y respeta la consideración de los/as estudiantes, y por el otro, la intervención de cada estudiante es necesaria, enriquecedora y valorada también por sus pares. Se construye así una red de relaciones que los/as estimula a profundizar su proceso de abstracción, tanto para manifestar sugerencias (dirección S de la rutina) como en la toma de decisiones para avanzar en un sentido u otro, incluso con mayor compromiso y eficiencia.

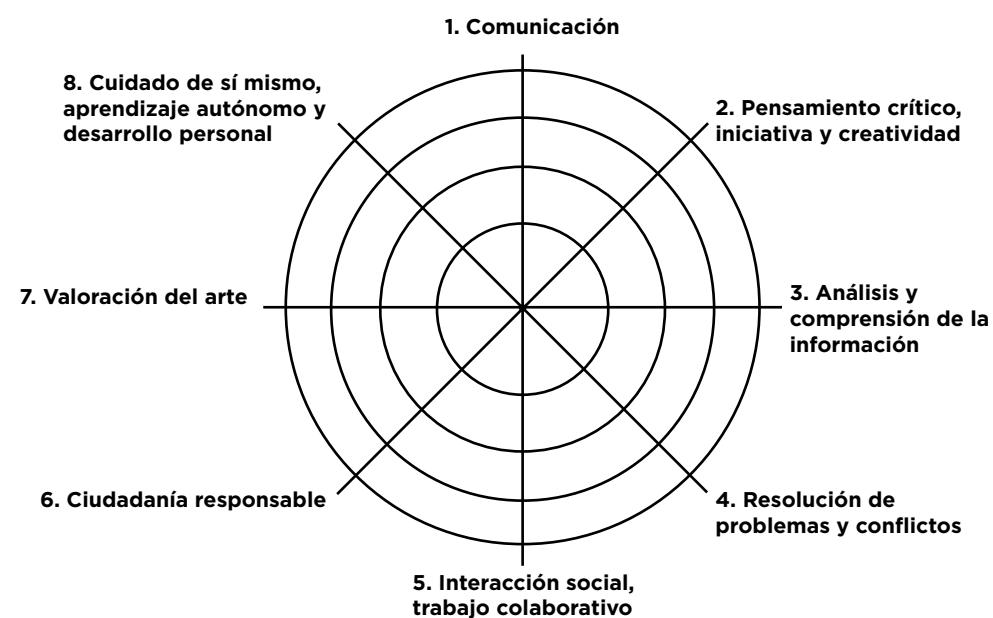
Para la **evaluación** de los aprendizajes alcanzados se sugiere hacer un seguimiento del trabajo realizado en las distintas etapas. El/la docente puede habilitar a cada grupo a realizar su explicación y justificación del modelo, las motivaciones y estrategias seleccionadas, como parte de la evaluación.

Por lo tanto, puede elaborar una rúbrica que dé cuenta de los aspectos que considere relevantes, tomando en consideración que no hay un único resultado posible o correcto, sino que puede haber múltiples resultados en función del modelo y las estrategias seleccionados por los/as estudiantes.

La modalidad de evaluación así concebida permite identificar los logros de cada estudiante y de cada grupo, y facilita a la/el docente su asistencia para que aquellos/as puedan concretar el producto final esperado.

Otra herramienta que se sugiere es la denominada **diana de evaluación**. Se trata de una opción de autoevaluación y coevaluación, es decir, es una opción participativa. A la vez es visual y permite incorporar diferentes aspectos o criterios a evaluar que el/la docente define de antemano.

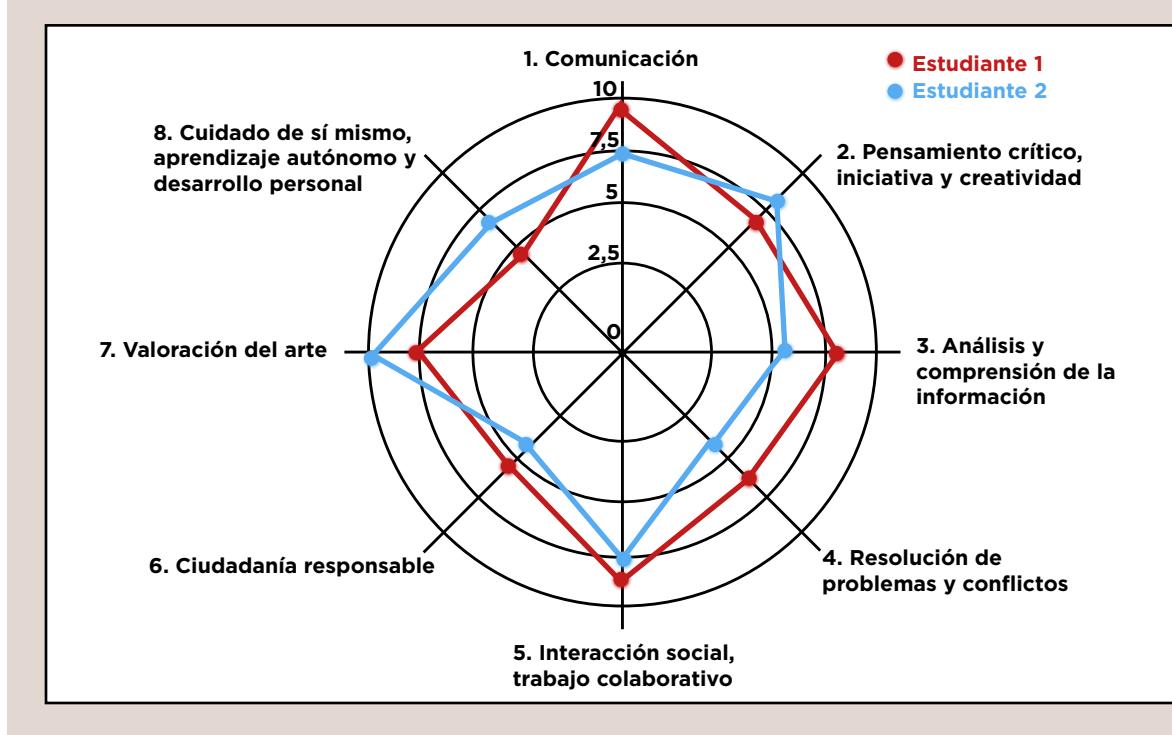
Se puede dibujar la diana en la pizarra o entregarla a cada grupo o a cada estudiante en diferentes formatos, ya sea en papel o en formato digital. La diana en sí misma consiste en anillos concéntricos divididos en secciones mediante líneas o ejes. Cada anillo tiene un rango de puntuación y cada eje representa un aspecto por evaluar, tal como muestra el siguiente ejemplo.



Ejemplo de diana de evaluación.

El/la docente puede adaptar este ejemplo de diana de evaluación a sus objetivos, a los aspectos específicos de cada curso o de ciertos grupos dentro de un curso, o incluso a cada estudiante en particular que elija considerar.

Los/as evaluadores (docente y estudiantes) marcan puntos en cada eje, es decir, evalúan teniendo en cuenta el puntaje asignado a cada anillo.



Luego se unen los puntos de cada evaluador/a y queda conformado un mapa de evaluación, que permite apreciar visualmente los aspectos con mayor fortaleza y los que hay que fortalecer.

El procesamiento del gráfico propiamente dicho se puede complementar con una planilla de cálculo, si el/la docente lo prefiere. Ambos elementos (diana en gráfico y planilla) pueden formar parte del material al que los/as estudiantes deberían tener acceso libre.

El uso de la diana de evaluación puede aplicarse al principio de la secuencia y funcionar como diagnóstico y, al final, como evaluación de logro de los aspectos considerados.

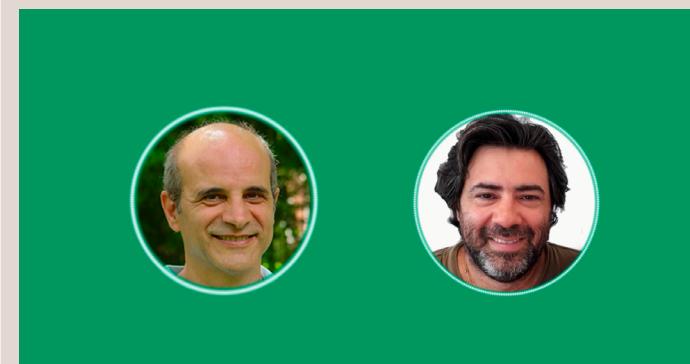
Dado que esta secuencia establece tres niveles de logro, cada docente puede decidir si sus estudiantes los desarrollan todos o solo parte de ellos. En función de esto, también podrá valorar el trabajo realizado con diferentes herramientas de evaluación, tanto las sugeridas aquí como otras de su preferencia.

Ejemplo de diana completa con la evaluación de dos estudiantes.



## Explorando fronteras

Para poder ver cómo se conectan los temas de esta secuencia didáctica con el mundo real, entrevistamos a Martín Oësterheld y Pablo Cipriotti, dos investigadores de la Universidad de Buenos Aires, más específicamente de la Facultad de Agronomía. Martín es ingeniero agrónomo, docente de la UBA, PhD e investigador del Conicet. Una de sus especialidades es la interacción entre plantas y herbívoros. Pablo es doctor en Ciencias Agropecuarias, especialista en ecología de poblaciones y modelos de simulación basados en agentes. Conversamos con ellos sobre la importancia de los modelos de simulación, el enfoque de la modelización basada en agentes, y sobre la importancia de trabajar con modelos y gráficos en el nivel secundario. Fue un lujo haber podido conversar con ellos. Esperamos que disfruten de la entrevista.



Entrevista

## Anexos

### Anexo 1. Tarjetas



### Anexo 2. Documento Decodificación del modelo de simulación en NetLogo



### Anexo 3. Glosario



[Link para descargar tarjeta imprimible](#)

## Anexo 1. Tarjetas



### Tarjeta A. Traducción de los comandos más usados en NetLogo

COMANDO	TRADUCCIÓN
breed [vacas vaca]	breed = raza del agente
vacas-own [energía]	los agentes de la raza vaca tienen energía como propiedad
ca	sigla de <i>clear-all</i> = borrar todo
set humedad 10	set = establecer el valor de una variable
ask n-of 100 patches [set pcolor green]	ask = preguntar/dicir <i>n-of</i> = <i>number of</i> , cantidad de patches = parcelas pcolor = color de parcela
ask patches with [pcolor = black] [set pcolor yellow]	with = con (decirle a todas las parcelas de color negro que se pinten de color amarillo)
set shape "vaca"	establecer la figura “vaca” para ese agente
set size	establecer el tamaño del agente
fd	forward = avanzar
rt	right = derecha
lt	left = izquierda

#### Para tener en cuenta:

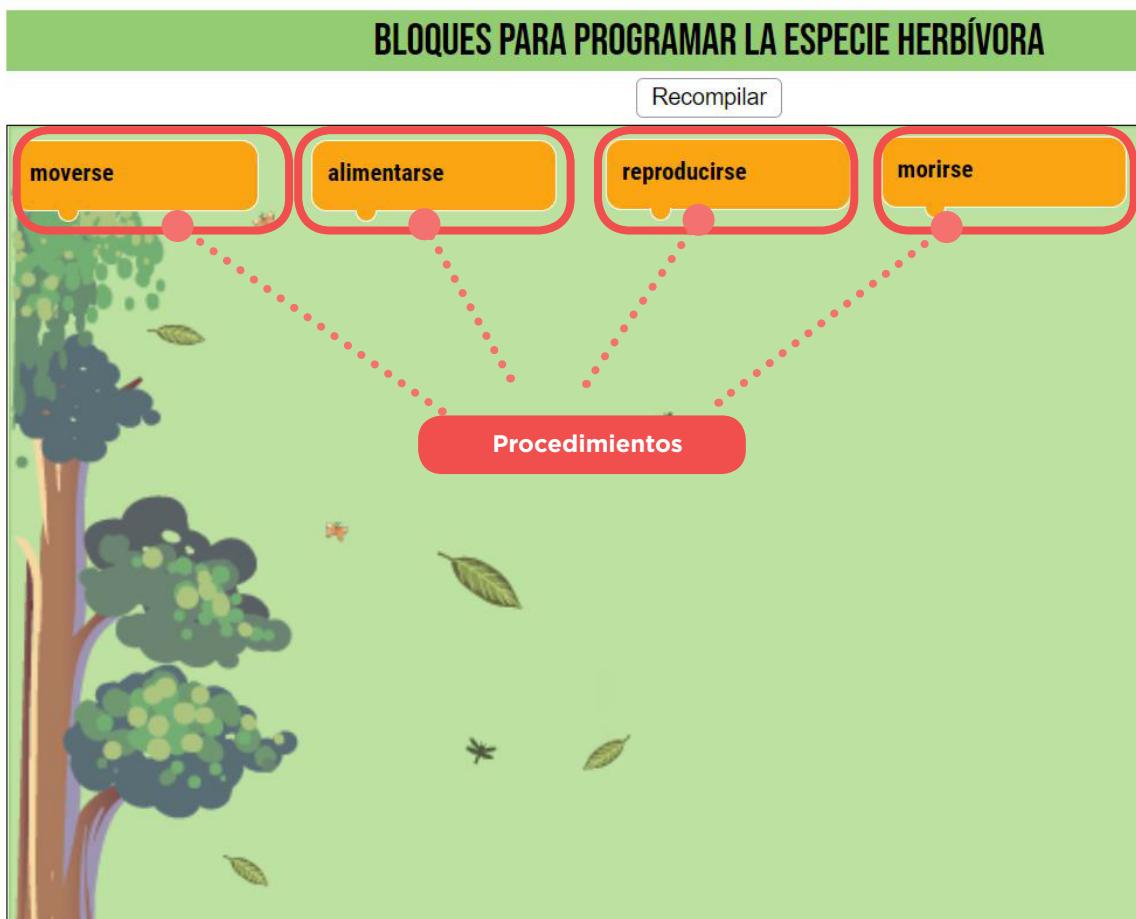
Si bien la interfaz de NetLogo está en español, los comandos son en inglés. En la tabla se muestran los comandos más comunes, pero hay muchos más. También en el sitio oficial de NetLogo puede consultarse el diccionario de las llamadas “primitivas” de NetLogo o instrucciones. Además de saber qué significa cada instrucción o comando, es muy importante conocer la gramática, es decir, dónde van corchetes o paréntesis, por ejemplo.



[Link para descargar tarjeta imprimible](#)



## Tarjeta B. Bloques de evento en NetTango



El ciclo de vida de la especie herbívora se compone de cuatro **procedimientos** o **eventos**: *moverse*, *alimentarse*, *reproducirse*, *morirse*. Observar que la forma de estos bloques impide que haya algo arriba de ellos, pero sí puede haber algo debajo. Y esa es la idea: ir acomodando debajo los bloques de comando, para programar las acciones correspondientes de cada uno de esos **eventos**.

Debajo del evento *moverse* se colocarán los bloques relativos al movimiento de cada individuo, como también lo que sucede con la energía cuando un individuo de una especie se mueve (pierde energía).

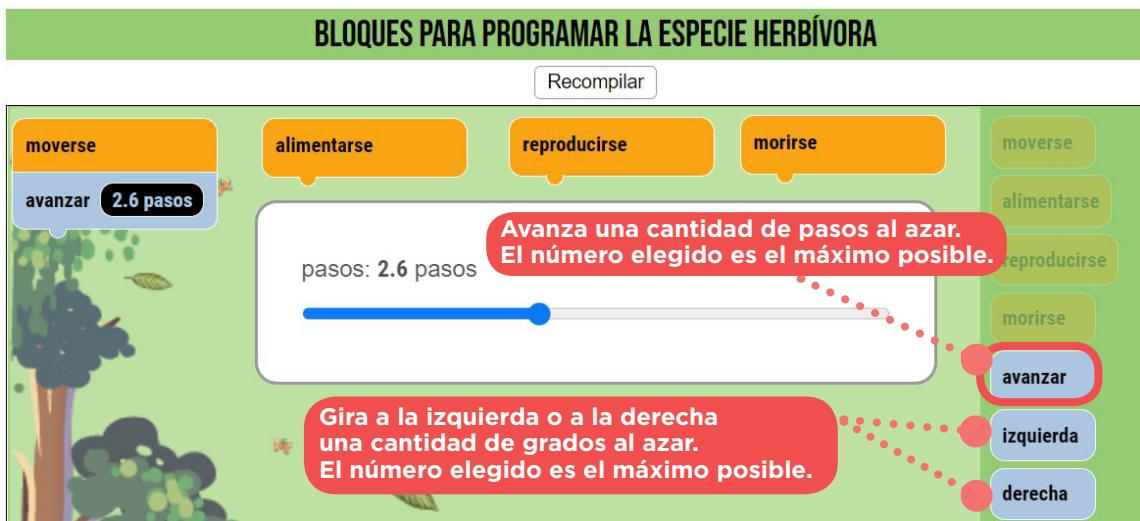
Debajo de *alimentarse* irán los bloques relativos a la alimentación de cada individuo. En esta simulación, el alimento está representado por parcelas de color verde y amarillo, que representan dos tipos de alimento. Y lo mismo para los restantes eventos.



[Link para descargar tarjeta imprimible](#)



## Tarjeta C. Bloques de movimiento en NetTango



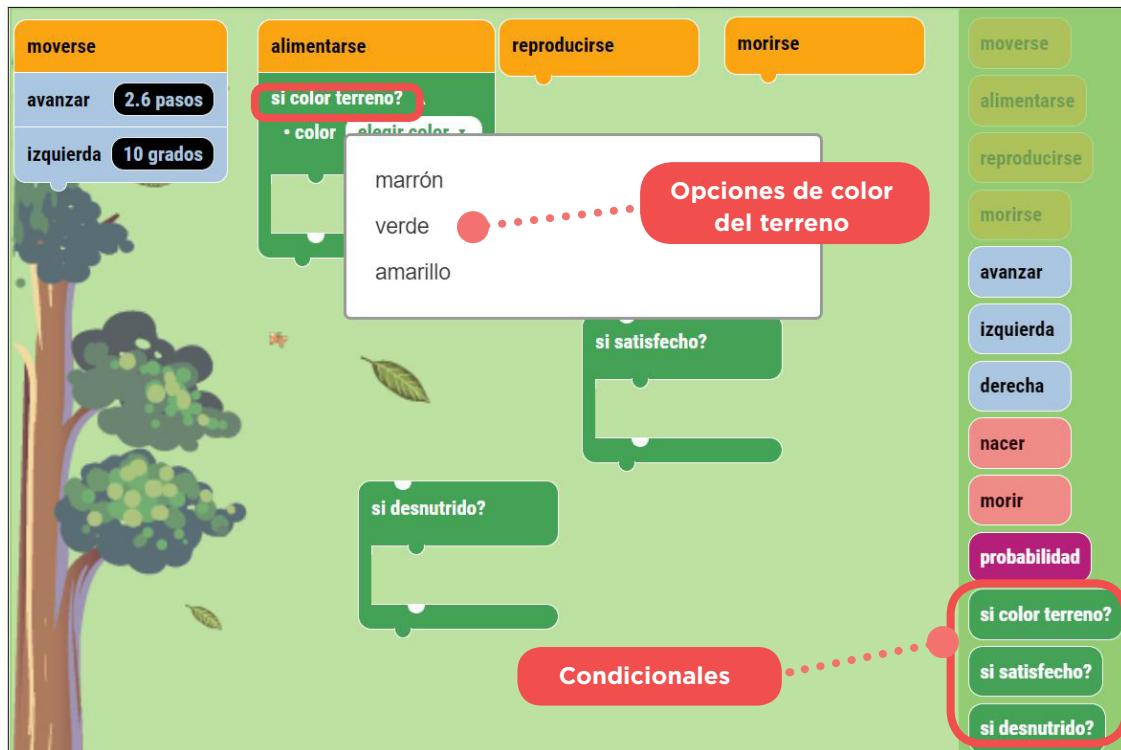
Los bloques de movimiento permiten programar a cada individuo para que avance, gire a la izquierda o a la derecha. Al arrastrar un bloque de movimiento debajo de un procedimiento (lo lógico sería debajo de “moverse”, tal como muestra la figura), se muestra un deslizador que permite seleccionar el valor máximo para ese movimiento (grados o pasos, según corresponda). En el caso de la figura, la especie, a cada ciclo del reloj, avanzará entre 0 y 2.6 pasos, eligiendo cada vez un número diferente.



[Link para descargar tarjeta imprimible](#)



## Tarjeta D. Bloques condicionales en NetTango



Los bloques **condicionales**, como su nombre lo indica, permiten establecer condiciones sobre lo que sucederá con los individuos ante determinadas situaciones. En este micromundo hay tres bloques condicionales: uno para sensar de qué color es el terreno (marrón, verde, amarillo) y luego otros dos para verificar la necesidad de alimento. El condicional *si satisfecho?* devuelve valor verdadero si el individuo acumula 25 unidades de energía (o más). El condicional *si desnutrido?* evalúa la condición como verdadera cuando el individuo tiene 0 unidades de energía.

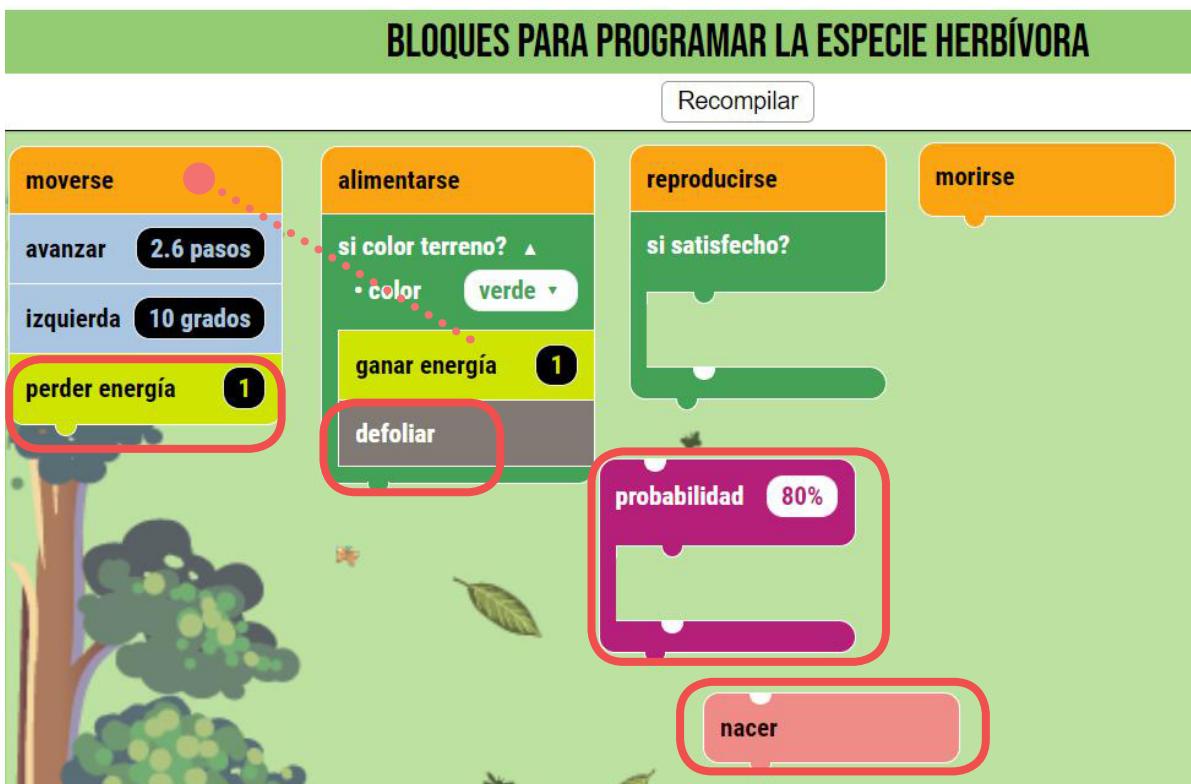
Con estos condicionales se puede establecer que si el alimento es verde, por ejemplo, el individuo gane tantas unidades de energía. Con el *si satisfecho?* se podría determinar si el individuo está en condiciones de parir una cría.



[Link para descargar tarjeta imprimible](#)



## Tarjeta E. Otros bloques en NetTango



Los bloques ganar y perder energía no necesitan mucha explicación. Los valores de unidades de energía son elegidos al azar, y el número que se muestra es el máximo posible. Por ejemplo, a cada paso del reloj en la simulación, cada individuo podría ganar 0.2 unidades y al siguiente tic del reloj, 0.4 unidades, etc. El bloque “defoliar” cambia el color de la parcela a marrón. Podría utilizarse para que cuando un individuo come, la parcela verde se vuelve de color marrón, indicando que ya no hay más alimento allí. El bloque “nacer” hace parir una cría y el bloque “probabilidad” permite establecer que el comando debajo de él se ejecute con una probabilidad determinada. Es un bloque clave para incorporar más aleatoriedad a la simulación.

## Anexo 2. Decodificación del modelo de pastoreo en NetLogo

[Decodificación del modelo base de pastoreo en NetLogo](#)



## Anexo 3. Glosario

**Deslizador:** Control visual para establecer valores de una variable.

**Mapa causal:** Es un tipo de organizador gráfico que vincula diferentes factores en un diagrama a través de relaciones causa-efecto. Los mapas causales son un recurso útil a la hora de analizar fenómenos complejos o multicausales.

**Modelo computacional estocástico:** Modelo que incorpora aleatoriedad, donde para dos corridas del simulador, a pesar de configurar los mismos valores de entrada, pueden darse resultados diferentes.

**Modelo de simulación bifocal:** Vinculación de un modelo de simulación computacional con un fenómeno físico en tiempo real (Blikstein y Wilensky, 2017).

**NetLogo:** Lenguaje de programación basado en agentes, descendiente del lenguaje LOGO creado en 1971 por Seymour Papert y Cynthia Solomon. LOGO fue el primer lenguaje de programación de computadoras creado para el ámbito educativo.

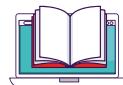
**NetTango:** Interfaz gráfica de usuario basada en bloques para programar dentro de un dominio particular con el lenguaje NetLogo. NetTango permite crear, de forma visual y sencilla, modelos de simulación programables.

**Palatabilidad:** Cualidad de un alimento de ser más o menos apetecible para una especie en función de sus aportes nutricionales.

**Placa robótica:** Dispositivo que consta de un microcontrolador, al cual se pueden conectar diferentes componentes electrónicos como resistencias, cables, diodos o sensores (por ejemplo, de luz o de sonido) y actuadores (por ejemplo, motores o luces).

**Sensor:** Dispositivo que permite captar señales del entorno, como luz, sonido y humedad, entre otras, y transformarlas en impulsos eléctricos u ópticos que pueden ser leídos por una placa robótica.

# Bibliografía



- Blikstein, P. y Wilensky, U. (2017). ["Modelización bifocal: el vínculo entre los experimentos reales y virtuales"](#). (Trad. C. Rizzi Iribarren). Trabajo original publicado en 2012 en la reunión anual de la American Research Education Association [Asociación Americana de Investigación en Educación], Chicago, Estados Unidos.
- Borrelli, P. y Oliva, G. (2001). "Efectos de los animales sobre los pastizales". En P. Borrelli y G. Oliva (editores), *Ganadería ovina sustentable en la Patagonia Austral*, pp. 99 a 128. Neuquén, Argentina: INTA Región Patagonia Sur.
- Church, D. C. (1979). "Taste, appetite and regulation of energy balance and control of food intake". En: Church, D. C. (editor), *Digestive physiology and nutrition of ruminants*, pp 281 a 290. Oregon, Estados Unidos: Oxford Press.
- Couso, L. y Fernández, R. J. (2012). ["Phenotypic plasticity as an index of drought tolerance in three Patagonian steppe grasses"](#). En: *Annals of Botany* N° 110, pp. 849 a 857. Oxford, Gran Bretaña: Oxford University Press.
- Distel, R. A. y Villalba, J. J. (2007). "Diversidad vegetal, selección de dieta y producción animal", en Revista Argentina de Producción Animal N° 27, pp. 55 a 63. Buenos Aires: Asociación Argentina de Producción Animal. <https://ri.conicet.gov.ar/handle/11336/26147>
- GCABA. Ministerio de Educación. Dirección General de Planeamiento e Innovación Educativa (2015). *Diseño Curricular para la Nueva Escuela Secundaria. Ciclo Orientado del Bachillerato. Agro y Ambiente*. Ciudad Autónoma de Buenos Aires.
- Giere, R. N. (2010). ["An agent-based conception of models and scientific representation"](#). En: *Synthese* N° 172, artículo 269.
- Resnick, M. (2001). *Tortugas, termitas y atascos de tráfico: exploraciones sobre micromundos masivamente paralelos*. Barcelona, España: Gedisa Editorial.
- Reyes, M. (2015). [Solapamiento de zonas de influencia de raíces de pastos perennes: controles en una estepa patagónica](#). Tesis doctoral. Buenos Aires: Universidad de Buenos Aires, Facultad de Agronomía.

Ritchhart, R.; Church, M. y Morrison, K. (2014). *Hacer visible el pensamiento. Cómo promover el compromiso, la comprensión y la autonomía de los estudiantes*. Buenos Aires: Paidós.

Ritchhart, R. y Church, M. (2020). *The Power of Making Thinking Visible: Practices to Engage and Empower All Learners*. San Francisco, Estados Unidos: Jossey Bass.

Snaydon, R. W. (1981). “*The ecology of grazed pastures*”. En: F. W. Morley (editor), *Grazing Animals*, pp. 79 a 104. Amsterdam, Países Bajos: Elsevier.

Weintrop, D. et al. (2020) [“El pensamiento computacional en las aulas de ciencias naturales y matemática”](#). Traducción y adaptación de C. Rizzi Iribarren. Artículo original publicado en 2016 en la revista *Journal of Science Education and Technology*, N° 25, pp. 127-147.

## Imágenes

- Página 11. Ovejas pastando, Pxfuel.com, <https://bit.ly/3zEIGWa>  
Página 13. Captura de pantalla de SimCity. Xardox, Wikipedia.com, <https://bit.ly/3kMaVhA>  
Página 43. Brújula, Wikimedia Commons, <https://bit.ly/3ocrghF>

## Recursos audiovisuales

### CmBlock 5 Explorando el entorno

Plan Ceibal, BBC Micro:bit. [bit.ly/3epHZs7](https://bit.ly/3epHZs7)  
Write you, Joey Pecoraro, Youtube Music Library. [bit.ly/3wL6g1V](https://bit.ly/3wL6g1V)

### Decisiones saludables: Un videojuego interactivo

Resumen desenfoque en el supermercado, topntp26, Freepik. [bit.ly/3wJW9up](https://bit.ly/3wJW9up)  
Drown me out, VYER, Youtube Music Library. [bit.ly/3wL6g1V](https://bit.ly/3wL6g1V)

### Decisiones saludables: Versión con puestos de control

Supermarket, Pexels / 9145, Pixabay. [bit.ly/3wEZWck](https://bit.ly/3wEZWck)  
Night vision, Bird Creek, Youtube Music Library. [bit.ly/3wL6g1V](https://bit.ly/3wL6g1V)



/educacionba

[buenosaires.gob.ar/educacion](http://buenosaires.gob.ar/educacion)