



ENTORNOS PARA ENSEÑAR PROGRAMACIÓN EN SECUNDARIA. NUEVOS ENFOQUES.

Nieves Carralero Colmenar.

IES Pedro Mercedes. Junta de Comunidades de Castilla-La Mancha. España.

Resumen

En los países anglosajones, especialmente en EE.UU, la asignatura de Programación está integrada dentro de la cultura educativa. Usar un lenguaje de programación como soporte a las Matemáticas, Física o Química o incluso áreas que no son exclusivas de Ciencias, es algo habitual en muchas escuelas de secundaria, y no exclusivo de los cursos especializados o las carreras universitarias. Muchos centros educativos en esos países trabajan con herramientas que ayudan a una mejor didáctica de la Programación, alejándose de los habituales entornos y lenguajes profesionales de Programación tipo Eclipse/Java o Microsoft .NET /lenguaje C, Basic, C#.

Las propuestas destinadas a educación primaria y secundaria van más en la línea de entornos gráficos que, alejando al alumno de fases tan desmotivadoras como, por ejemplo, el análisis léxico/sintáctico de compilaciones o interpretes, permiten centrar el interés de la asignatura en el uso de estructuras lógicas que permitan desarrollar aplicaciones gráficas vistosas (con mayor o menos complejidad algorítmica), enganando así al alumno.

En este artículo se introducen tres alternativas importantes destinadas a que los alumnos aprendan programación de manera amena y divertida, asentando conceptos como bucles, condicionales, eventos, etc., con el objetivo de crear entornos multimedia y juegos gráficos, huyendo de aburridos entornos de consola.

Las herramientas que se van a describir en este artículo son: Alice, Greenfoot, Scratch. No son las únicas herramientas existentes con este objetivo, pero si son de las más conocidas y apoyadas por instituciones tanto públicas como privadas.

1. ALICE



Alice es un entorno de programación destinado a trabajar con escenarios virtuales en 3D. Creado en la Universidad de Carnegie Mellon. Según [\[NEOTEO\]](#)

El proyecto Alice es una iniciativa para combatir la falta de interés en las ciencias de la computación. Su entorno innovador en programación 3D hace que el crear una animación, un juego interactivo o video sea algo fácil y motivador.

Alice tiene como objetivos principales alejar a los estudiantes noveles en el mundo de la programación de entornos “duros” con problemas poco motivadores, generalmente matemáticos, y en los que corregir errores sintácticos suelen ser el 90% de la programación.

Alice propone objetivos más motivadores, centrados en el proceso de la programación y no tanto en la sintaxis. Además, Alice propone también adquirir buenas prácticas en la programación, ayudando a que el estudiante aprenda a diseñar antes de programar y a que su desarrollo sea cooperativo, quitando vicios como “el código personalizado” que nadie, fuera del creador, puede entender en un tiempo razonable.

Tal y como indica [\[NEOTEO\]](#)

Alice es un programa para construir mundos virtuales con objetos 3D. Los objetos pueden moverse, girar, cambiar color, reaccionar al ratón y mucho más. Su interfaz interactiva genera instrucciones al arrastrar y soltar elementos gráficos (drag and drop). Estas instrucciones se corresponden a lenguajes de programación como Java o C++. Al ver en forma inmediata cómo corren los programas de animación, los estudiantes pueden entender con mayor facilidad la relación entre el código y el comportamiento de un objeto. Su entorno visual mejora la retención y el aprendizaje, evitando la frustración de una sintaxis mal utilizada. Los objetos se vuelven obvios y el estudiante puede relacionarse con ellos y la forma en que se programan.

Alice ayuda a que los estudiantes adquieran habilidades de programación como son:

- Capacidad de organizar procesos: Hacer algoritmos
- Pensamiento abstracto

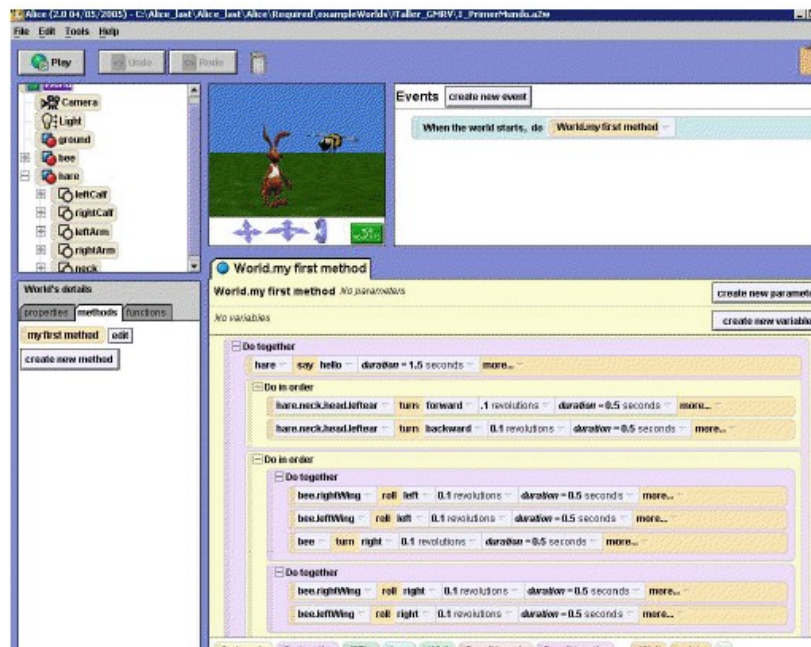
- Capacidad de integrar conocimientos y habilidades de varias áreas (Competencias).

La página principal de Alice es www.alice.org, y en ella nos podemos encontrar las diferentes versiones de Alice para descarga gratuita. Actualmente, está en fase Beta Alice 3.0 en la que ha participado EA (Electronic Arts) famosa empresa de video juegos, lo que da una idea de la importancia del proyecto Alice.

Para empezar a utilizar Alice, necesitarás crear un entorno (Alice World). Iniciando la aplicación, pincha "Open a World" en la ventana de bienvenida. También dispones de un tutorial dentro de la aplicación para conocer las utilidades de Alice. Para verlo, sólo tienes que pinchar en el botón "Start the Tutorial" en la ventana de bienvenida.

Es importante destacar que Alice no crea objetos 3D. Tiene una galería de objetos predeterminados para que utilices en tu aprendizaje, pero si quieres utilizar otros modelos tendrás que diseñarlos en otro programa y exportarlos o descargarlos de la Web.[NEOTEO]

La siguiente imagen muestra el entorno de desarrollo. Para más información ver los vídeos disponibles en http://www.alice.org/index.php?page=what_is_alice/what_is_alice



En la página web de Alice también hay disponible enlaces a escenarios, foros y muchos otros elementos indispensables para dar los primeros pasos con esta herramienta. Respecto a los tutoriales y manuales, dentro del entorno, Alice tiene un tutorial sobre su funcionamiento, y en la red se pueden encontrar varios vídeos

hechos a modo de tutorial. Luego, para guías más detalladas, Alice tiene a la venta manuales muy completos (no gratuitos).

A continuación se muestra una imagen de un juego hecho con Alice (Persiguiendo a T-Rex) disponible junto con otros muchos en [SIMPLE]:



INTERESANTE: Para profundizar más en Alice, aconsejable ver las siguientes direcciones:

www.Alice.org.

Tutorial Básico de Alice en Castellano.

<http://www.gmrv.es/juvenalia/Tutorial Alice.pdf>

2. GREENFOOT



Greenfoot es un herramienta que comparte los mismos objetivos que Alice o Scratch, es decir, enseñar a programar con entornos amigables que no se centren en la sintaxis y si en el proceso de la programación. Sin embargo, Greenfoot, es un paso más de complejidad que Alice o Scratch. Greenfoot ofrece un entorno y una manera de trabajar más cercana a los lenguajes de programación profesional ya que si tiene en cuenta la sintaxis del lenguaje y otros aspectos que Alice o Scratch no. Según [GREENFOOT]:

Comparativamente, en otra herramienta de aprendizaje de programación conocida como [Alice](#), al no existir en el entorno de programación un editor del código de programación propiamente dicho, se sorteian las dificultades inherentes al rigor sintáctico a los primeros pasos en la programación

orientada a objetos. Se trata de que el árbol no nos impida ver el bosque. Algunos profesores han encontrado que estudiantes que pueden programar en Alice se enfrentan a posteriori con dificultades a la hora de introducirse en un lenguaje de programación tradicional que usa editor de texto (para la sintaxis).

El principal inconveniente de Greenfoot es que es un lenguaje no demasiado extendido para trabajarlo en Educación Secundaria (En España) por lo que la gran mayoría de información de este lenguaje está en Inglés. Esto puede o no puede ser un problema, dependiendo del profesor y los alumnos a los que se destine.

Según [GREENFOOT]:

Greenfoot es un ambiente interactivo de desarrollo Java para propósitos educacionales. Permite la creación de aplicaciones gráficas en dos dimensiones, como simulaciones y juegos. Siendo relativamente sencillo y en poco tiempo crear simples videojuegos en Greenfoot, es igualmente posible construir sofisticadas simulaciones de complejos sistemas, con la posibilidad de usar algoritmos de inteligencia artificial o conectividad a bases de datos, por poner algunos ejemplos.

El modelo de programación Greenfoot consta de una clase World- Mundo (representado por un área de pantalla rectangular) y objetos Actor que están presentes en el mundo y pueden ser programados para actuar de forma independiente. El mundo y los actores están representados por objetos Java y definidos por las clases de Java. Greenfoot ofrece métodos para programar fácilmente a estos actores, incluido el método de movimiento, rotación, cambios de apariencia, la detección de colisiones, etc.

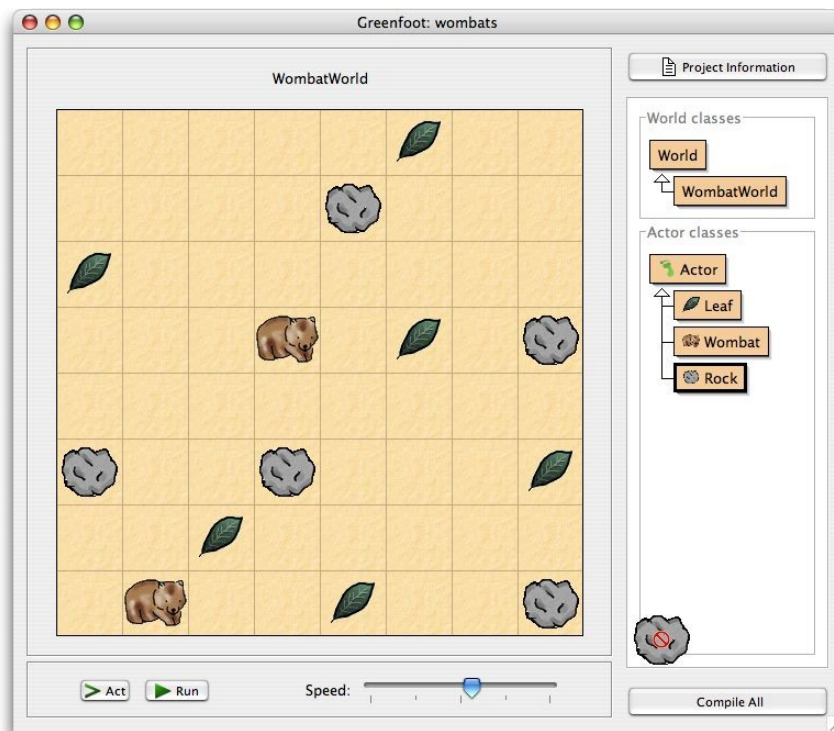
La programación en Greenfoot en su forma más básica consiste en dos subclases de clases incorporadas, mundo y Actor. Una instancia de la subclase Mundo representa el mundo en el que se desarrolla la acción. La subclase Actor tiene objetos que pueden existir y actuar en el mundo.

La Ejecución es básicamente un bucle que repetidamente invoca el método de cada actor. La programación de un escenario, por lo tanto, consiste principalmente en la implementación de métodos para que los actores se muevan en el escenario.

La principal diferencia de Greefoot con respecto a Alice o Scratch es que se usa un lenguaje de programación La aplicación se hace en Java estándar, aunque Greenfoot ofrece API de una serie de tareas comunes, tales como animación, sonido, al azar, y la manipulación de imágenes. Todas las bibliotecas estándar de Java se pueden utilizar también.

Greenfoot, al igual que Alice o Scratch está disponible tanto en Windows como en Linux.

Una imagen del entorno es la siguiente:



INTERESANTE: Para profundizar más en Greenfoot, aconsejable ver la siguiente dirección:

<http://www.greenfoot.org>.

3. SCRATCH



Según la información del producto Scratch [SCRATCH], esta aplicación está enfocada hacia niños de 8 años en adelante para que desarrollen habilidades de aprendizaje en programación. A medida que van creando sus proyectos, los creadores afirman que los chicos irán aprendiendo conceptos matemáticos/informáticos fundamentales, además de ir obteniendo experiencia en el concepto de “diseñar” un proyecto.

En Scratch, la idea es dar la posibilidad de que por medio de una mezcla de sonidos, imágenes y animaciones, se puedan lograr productos interactivos y creativos. Scratch abre un nuevo mundo de posibilidades, ya que ellos podrían subir su contenido creado a internet, compartirlo, y así incentivar a otros a lograr algo similar o mejorar lo realizado.

Scratch funciona para Windows, Linux y Mac. En el sitio oficial de la aplicación (<http://scratch.mit.edu/>) se pueden subir trabajos y presentarlos en comunidad para ser

compartidos por otros usuarios. Además, en los últimos dos años, desde el propio MIT (creado de Scratch) y otras organizaciones se han convocado concurso para los mejores desarrollos usando Scratch.

La principal aportación de Scratch es que está destinado a usuarios de temprana edad, lo que lo hace directamente aplicable como herramienta didáctica destinada a enseñar programación en particular e informática en general, a alumnos de primaria y secundaria. Cada vez más, los alumnos están interesados en la programación como medio creador de diferentes utilidades, aplicaciones y juegos, apareciendo este interés cada vez más en edades tempranas. Scratch ofrece una perspectiva avanzada en conocimiento pero sencilla en manejo.

Para aquellos que llevan muchos años en la docencia de la Informática en las aulas y que no han entendido hasta el momento lo que ofrece Scratch, se puede decir que Scratch es un sustituto avanzado de herramientas “ancestrales” como Logo.

En los siguientes puntos se muestra alguna de la funcionalidad de Scratch y algunas orientaciones didácticas.



Funcionalidad básica

Scratch trabaja desde local, es decir, es una aplicación que se descarga de Internet y se tiene que instalar en los ordenadores de los alumnos/as. No es una aplicación Web 2.0, por lo que no se necesita una conexión a Internet para poder trabajar con ella. La siguiente dirección es para descargar la aplicación <http://scratch.mit.edu/>

La aplicación tiene versión en castellano y otros idiomas, además del Inglés.

La funcionalidad básica es [Mundoeduca]:

- Es programación orientada a objetos, con las estructuras básicas como: asignaciones, condicionales, bucles y, funciones.
- Interfaz intuitiva y muy gráfica.
- Existencia de un banco de recursos en el propio programa: objetos, personajes ,escenarios, sonidos...
- La acciones de los personajes se forman apilando piezas a modo de puzle.
- Visualización instantánea de lo que hacemos en el programa.
- Interacción con los objetos y personajes que introducimos con el ratón y teclado, es decir, una vez que creamos nuestro proyecto el usuario puede interactuar moviendo los objetos en la pantalla
- Posibilidad de difundir nuestro proyecto en la web.

Uso en el aula.

Como ya se ha comentado, Scratch tiene muchas posibilidades tanto en el aula de primaria como de secundaria. Debido a que tiene una interfaz gráfica basada en bloques, Permite hacer aplicaciones multimedia con diferentes niveles de complejidad, y esta es la característica que la hace más atractiva para el alumnado, ya que, mientras hacen un multimedia, aprenden, sin saberlo a programar. Sin duda, este es el enfoque didáctico más atractivo a la hora de usar esta herramienta.



The screenshot shows the Scratch website's project gallery. At the top, there is a navigation menu with links for 'Inicio', 'proyectos', 'galerías', 'soporte', 'foros', 'acerca de', 'mis cosas', and 'Idioma'. Below the menu, a welcome message says 'Bienvenido, [corcosuk](#) | [Salir](#)'. A search bar is also present. The main content area is titled 'Proyectos cargados recientemente' and includes a link to '¿Quieres saber como [cargar tus proyecto?](#)'. There are four filter buttons: 'más reciente', 'Lo más visto últimamente', 'Lo más popular últimamente', and 'Lo más reiventado últimamente'. Three project cards are visible: 1. 'merrill animation03' by 'rmerrill', featuring a 'BAD LUCK' video thumbnail. 2. 'club penguin run!' by 'snivy123', featuring a penguin game thumbnail and a description: 'penguin runs from an axew axew runs how it should but the penguin is diffrent up is w down is s right is d and left is a'. 3. 'Middleton-Animation03' by '001087', featuring a 'Cartoon' thumbnail.

Para ver qué hay hecho por otros en Scratch, la página de Scratch ofrece proyectos compartidos por la comunidad para libre disposición. Todos estos proyectos se pueden usar de dos maneras:

1. Como un recurso multimedia más para el aula. Por ejemplo, si hay un recurso que tiene un enfoque didáctico adecuado, se puede usar tal cual como un recurso más para el aula.
2. Como la semilla de un proyecto mayor. Un grupo puede coger uno de estos proyectos y extenderlo con funcionalidad propia para hacerlo más completo o potente. Al ser proyectos libre para poder editarlos, la reutilización y extensión ofrece muchas posibilidades.

Por otro lado, la herramienta puede usarse de muchas maneras, pero todas ellas cumplen con un objetivo claro: fomentar el pensamiento algorítmico en los alumnos/as:

- A. Trabajar con los alumnos/as conceptos básicos de programación: asignación de variables, condicionales y funciones para interactuar con un elemento gráfico.

Gracias a la programación con bloques, a las funciones predefinidas en Scratch y a su librería de objetos, ésta es una buena plataforma para hacer los primeros pinitos en programación, por ejemplo, para hacer un sistema de preguntas y respuestas.

- B. Trabajar con los alumnos/as aspectos avanzados de la programación: repeticiones (while, repeat, for) y aspectos de programación orientada a objetos - POO.

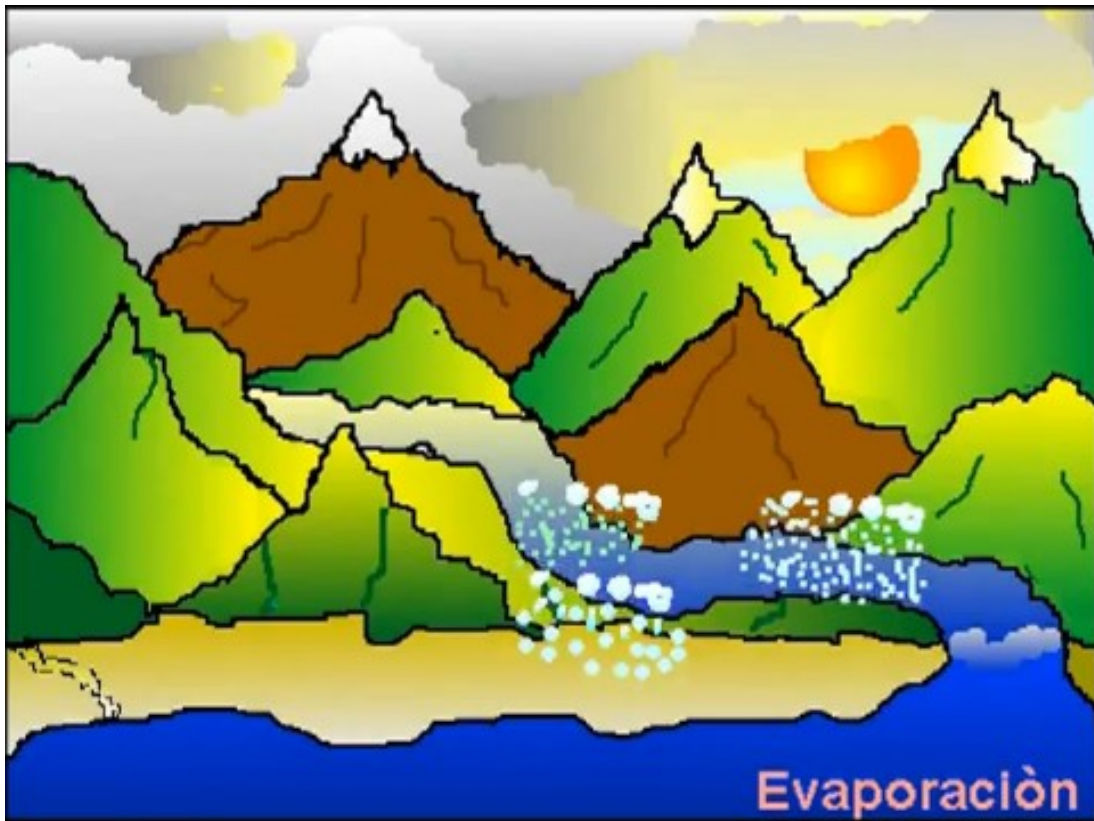
Esta perspectiva permite hacer aplicaciones más complejas, en donde los bucles y la interacción entre objetos encuentran su principal objetivo. Por ejemplo, hacer un sistema para que un objeto se mueva en una u otra dirección según decida el usuario (por teclado). Esta manera de usar Scratch es más adecuada para 3º y 4º ESO ya que se necesita la capacidad de esas edades para abstraer las estructuras de programación.

Tanto los ejemplo A como B están destinados a enseñar al alumno/a a programar de manera explícita, a resolver construir algoritmos y familiarizarse con la programación orientada a objetos - POO.

Sin embargo, un enfoque más atractivo válido tanto para Primaria como para Secundaria sería, con la excusa de desarrollar una aplicación multimedia que muestre un determinado contenido educativo, manejar la programación de Scratch para desarrollarlo. De esta manera, el objetivo que percibe el alumno/a es el de conseguir un fin (material) y no el de aprender a programar. Así se consigue que los alumnos/as interioricen el uso de la programación como una herramienta TIC más y no como algo que hay que aprender con independencia de para qué puede servir.

A continuación se enuncian algunos ejemplos reales de multimedias hechos por alumnos/as [PROYECTOS]:

Ciclo del agua: Es un multimedia que muestra el ciclo del agua en todas sus fases: Evaporación, Condensación, Precipitación e Infiltración. Todo se hace de manera automática (no hay interacción con ningún usuario). Sin embargo, tiene movimiento y para ello se necesita programar: El agua se evapora, se crean las nubes, se agrupan, cae la lluvia. Una imagen de este ejemplo que muestras cómo el agua se evapora es el siguiente:



Fases de la Luna: Este proyecto muestra las fases de la luna mientras se desplaza. Este proyecto si tiene interacción con el usuario mediante botones, al permitir “Animar” el modelo o “Salir” de la aplicación. Una imagen del proyecto es la siguiente:



Juego de Salud bucal: Este es un típico juego de coger cosas que caen del cielo. En este caso, es un niño que tiene que coger sólo los alimentos que son buenos para su salud bucal. Se maneja como cualquier juego de ordenador mediante flechas. Una imagen es la siguiente:



4. Conclusiones

Las tres aplicaciones mostradas ofrecen tres enfoques diferentes para trabajar la programación en las aulas de secundaria.

Scratch ofrece el enfoque más sencillo. La sencillez del entorno y el tipo de juegos que se pueden crear lo hace muy adecuado para edades tempranas. Los estudiantes con Scratch se centran más en los procesos de la programación orientada a objetos, asimilando las estructuras de control y los eventos, sin desmotivarse crean código complejo, ya que todo se hace mediante bloques. Además, Scratch tiene mucha información en castellano, lo que es importante para una mejor difusión.

Alice es el segundo nivel de complejidad. Aunque la filosofía de trabajo es similar a Scratch, al trabajar con objetos 3D hace que los estudiantes tengan que planificar mejor un diseño de la aplicación a generar y usar algoritmos más complejos. Pero, pese a todo, es un entorno muy cómodo, sobre todo para estudiantes que han trabajado con Scratch.

Greenfoot, es el de mayor nivel de los tres. Al usar Java como lenguaje de programación, se consigue que los estudiantes se enfrenten a los problemas de la

sintaxis. Sin embargo, al aplicarse el lenguaje sobre entornos gráficos (juegos) la motivación es mayor que si se hiciera en una consola. Lo más adecuado es usar Greenfoot en Bachillerato, sobre estudiantes que ya conocen Scratch o Alice como entornos de programación.

5. Bibliografía

[SCRATCH] <http://scratch.mit.edu/>

[NEOTEO] <http://www.neoteo.com/alice-aprende-a-programar-con-objetos:>

[SIMPLE] <http://www.ingenieriasimple.com/alice/index.html>

[GREENFOOT] <http://es.wikipedia.org/wiki/Greenfoot>

[GUÍA] Guía Didáctica para Profesores Sormenezko Zerbitzuak / Servicios Creativos www.isuriarte.com <http://isuriarte.com/wp-content/uploads/2010/10/ScratchGu%C3%ADaDid%C3%A1cticaProfesores.pdf>

[PROYECTOS]

http://www.ceibal.edu.uy/UserFiles/P0001/ODEA/ORIGINAL/100910_scratch2.elp/algunos_ejemplos.html

[MUNDOEDUCA]: <http://www.mundoeduca.es/scratch/index.htm>