

Programando con Scratch – guía del tutor v1.2

¿A QUIÉN VA DIRIGIDO “PROGRAMANDO CON EL GATTO”?

En principio, la escribí para niños de entre 7 y 10 años. Los niños no tienen por qué haber tenido ningún contacto con el ordenador previamente, pero si lo han tenido, por supuesto, ayuda. Conceptos como *arrastrar* con el ratón, *pinchar*, el sólo hecho de que el ratón mueva algo en pantalla, pueden ser difíciles de entender si no se han aprendido de antemano. En todo caso, las primeras lecciones del tutorial están enfocadas a familiarizarse con ese tipo de gestos ratoniles; en caso necesario, sería conveniente dedicarles más tiempo (o hacer *ex profeso* unas lecciones adicionales).

¿QUÉ HACE FALTA PARA COMENZAR?

Bajarte el Scratch, claro. La primera opción es obtener una beta antigua del sitio EducarEx <http://squeak.educarex.es/Squeakpolis/70> (ir a Zona de Descarga). La segunda es escribir al equipo de desarrollo, cuya dirección está en la página oficial de Scratch: <http://scratch.mit.edu/download>, y rellenar un formulario. Generalmente, contestarán positivamente y darán acceso a la versión más actual. Sin embargo, el tutorial está desarrollado basado en la versión de junio de 2005 (la disponible en la web de la Junta) y es posible que haya alguna diferencia.

La versión descargable desde la junta está lista para usar en Windows. Funciona incluso en Win 98. En Linux, hay que descargar un componente adicional, el Squeak: <http://www.squeak.org/>. Lo puedes hacer con tu programa de descarga habitual, emerge, yum o apt-get. En ubuntu/Debian/(GNU|Guada)Linex no tienes más que hacer apt-get. En Fedoras, SuSes y similares tendrás que descargarlo. Aunque siempre puedes compilarlo. Parece ser que las últimas versiones ya no existen para Linux, pero supongo que todavía se podrán descargar las anteriores.

Para ejecutar Scratch, simplemente ejecuta Squeak desde el directorio donde has guardado Scratch. Scratch es una *imagen* de Squeak, con lo que se cargará automáticamente.

Evidentemente, hace falta un ordenador por niño. Se puede hacer también con dos ordenadores por niño, pero eso puede dar lugar a problemas de coordinación que es mejor evitar. En ese caso, intentar que alguno de los niños mayores se sienta con uno de los pequeños, y asignar tareas a cada uno de ellos, en función del dominio que tenga cada uno. Tened en cuenta que Scratch se puede ejecutar incluso en ordenadores relativamente antiguos, o en portátiles, por lo que podéis *reciclar* ordenadores que tengáis en casa sin mucho uso para esta labor.

No hace falta un ordenador para el profesor, pero si un ordenador con pantalla suficientemente grande o visible para dar alguna explicación. Por lo mismo, no

hace falta un proyector, pero si se tiene en casa un home cinema, y puedes adaptar la clase para darla ahí, mejor que mejor. La cuestión es que la clase parezca una clase, no darla en el cuarto de juegos de los niños, por ejemplo. Es importante que los niños se pongan en una actitud de aprendizaje, por eso hay que dar esa apariencia. Por la misma razón, debéis llevar tanto tú como ellos ropa medianamente formal; es decir, nada de pijama, despeinado, con la cara sucia. A clase se va con la ropa de clase.

Finalmente, al arrancar Scratch salen los bloques del puzzle en inglés. Para cambiarlo a español, hay que ir al menú Extras -> Set Blocks Language -> Spanish. Puedes hacerlo tú, pero en mi experiencia es de las primeras cosas que aprenden los niños. Incluso, al cabo de cierto tiempo, se conocen ya los bloques de forma que no les hace falta ni mirar el idioma.

¿CÓMO ESTÁ ESTRUCTURADO EL TUTORIAL?

Bueno, los niños son niños (tautología que es estrictamente cierta). No puedes aburrirlos con largas explicaciones, y lo mejor es que hagan las cosas por sí mismos. Así usan su memoria funcional, y recuerdan todo mejor. Por eso hay explicaciones cortas, luego actividades (que serán actividades dirigidas por el tutor, y que los niños tendrán que ir haciendo), y finalmente ejercicios, que el tutor tendrá que corregir.

Por último, conviene repasar, cada clase o cada dos clases, y siempre a la vuelta del fin de semana. Así los niños acaban aprendiendo cosas a base de repetirlas.

No hace falta encargar “deberes” o actividades fuera de clase. Sin embargo, conviene *mandarlos* para el fin de semana, aunque no se hagan; por lo menos, lo mantienen en la mente durante ese tiempo.

¿QUÉ RITMO SE DEBE SEGUIR?

Cada grupo tiene su propio ritmo, como es natural. Pero incluso en clases heterogéneas, conviene marcar un ritmo, y ayudar a los que se queden más atrasados, mandando actividades adicionales a los más avanzados. En mi experiencia, un par de lecciones en cada clase de una hora es un ritmo asequible, e incluso sobra tiempo a veces para repasar o hacer ejercicios adicionales (en las partes más fáciles).

Las clases no deben tener más de una hora, incluso aunque se repitan todos los días.

¿CÓMO DEBO HACERLO?

Yo lo he enseñado en casa, a familiares y a amigos. Si lo haces así, debes dejarles bien claro que están en clase, que tienen unos horarios y unas reglas que cumplir durante la clase. Yo también he encontrado positivo enmarcarlo dentro de una especie de “Campus”. A los niños les gusta tener organizado el tiempo,

incluso en vacaciones; así no se aburren. El resto pueden ser actividades domésticas (ordenar su habitación) o lúdicas (dar un paseo en bici o bañarse en la piscina), pero deben acostumbrarse a tener, todos los días a esa hora, la clase.

Por supuesto, si eres un profesor y lo usas en clase, todo eso lo tienes ya. Pero si eres un padre o madre y quieres hacerlo con tus hijos (lo que recomiendo vivamente), yo he encontrado positivo hacerlo de esta forma.

¿TENGO QUE SABER YO PROGRAMACIÓN?

No sabría decirte. Yo creo que no. Pero te conviene seguir el tutorial, y por lo menos aprender qué es lo que puedes hacer y lo que no. Verás que Scratch es muy fácil, incluso para el que no tiene ni idea de programación, a la vez que introduce conceptos de programación de una forma lúdica y divertida.

PERO ¿LOS NIÑOS APRENDEN?

Los niños aprenden de forma inevitable, y les queda un poso incluso un año después. Hay conceptos que no han aprendido en clase que les cuesta cierto trabajo: ángulos, coordenadas; posiblemente, las notas musicales. Los niños tienen una visión muy *gestalt* de los puzzles: muchas veces hacen puzzles con todas las piezas que han aprendido, aunque algunas no sirven; les cuesta cierto trabajo entender la secuencia del programa y que los bucles se repiten; la simultaneidad de ciertas acciones también les cuesta trabajo. Como se ha dicho, es cuestión de incidir en ello, repetir, mandar ciertos ejercicios todos los días para comenzar la clase, comentar programas que se bajen del sitio del MIT (que están disponibles directamente desde el menú de Open, eso lo descubren inmediatamente).

Ten en cuenta que, inevitablemente, también aprenderán cosas sobre ti y tu relación con ellos. Van a aprender que, una hora al día durante cierto tiempo, estás dedicado a ellos, les estás prestando atención y te preocupas; van a aprender a cómo reaccionas cuando hacen las cosas bien y cuando las hacen mal, van a aprender que, incluso aunque seas su padre o madre o tutor, si les estás dando clase hay ciertos límites que no deben rebasar: prestar atención, hacer los ejercicios, tenerlo todo listo a su hora.

¿ALGÚN CONSEJO MÁS?

Mi experiencia es que la diferencia entre niños y niñas no es tanta como la que hay entre un niño y otro. La programación no es para niños, y yo he tenido excelentes alumnas. Tampoco he visto actitudes diferentes para acercarse a la programación por parte de niños y niñas.

Conviene que, al final del curso, organicéis un *teatrillo*, en realidad, un proyecto de programación. Tendrán que escribir el guión: organizar una secuencia de actividades, decidir qué va a hacer cada uno de los actores, y finalmente, representarlo. Con lo contenido en el tutorial es fácil hacerlo: los gatos pueden moverse,

maullar, decir cosas. Si hace falta alguna pieza adicional no mencionada, el tutor puede ayudar al alumno con eso. Y, finalmente, representarlo en presencia de los padres o tutores, y entregar un diploma. El diploma puedes hacerlo como te dé la gana; hay excelentes plantillas para Power Point (que puedes usar en OpenOffice). A los niños les gusta también participar en el diseño del diploma, pídeles consejo y opinión.

En vez de un teatrillo, podría ser un juego, pero quizás esto sea un poco más complicado. En cualquier caso, la idea es que tanto los niños como los padres se den cuenta de lo que pueden hacer, y lo muestren a los demás; también que sean capaces de hacer algo desde el principio, desde la idea hasta el final.

Por otro lado, conviene que el tutorial no quede ahí, sino que continúe a lo largo del tiempo, con recordatorios semanales, por ejemplo, o con pequeños ejercicios. En cualquier caso, es conveniente que se deje disponible el resto del tiempo a los niños para que puedan usarlo cuando les apetezca, sin muchas restricciones, así mantendrán fresco el manejo básico, y pueden incluso descubrir cosas por sí solos; los niños tienen una curiosidad infinita, y eso les conduce a aprender cosas por sí solos que sería difícil contarles de otra forma.

COMENTARIOS Y CRÍTICAS

El tutorial es libre con licencia GFDL, o sea que puedes hacer con él lo que te permite la licencia, que es mucho. Especialmente, puedes modificarlo y publicar las modificaciones, y te animo a que lo hagas (si me lo dices, mucho mejor). Si observas algún error en el mismo, o quieres contarme tu experiencia o hacer alguna sugerencia, mi correo electrónico es jj (en) merelo (punto) net.

DISPONIBILIDAD

Esta guía, y la guía a la que acompaña, estarán disponibles en <http://geneura.ugr.es/~jmerelo/tutoriales/programando-con-scratch/programando-con-scratch.zip>

LICENCIA

Este tutorial tiene licencia GFDL, cuyo texto puedes obtener en <http://gugs.sindominio.net/licencias/gfdl-1.2-es.html>

Copyright (c) 2006, 2007 J J Merelo. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU, Versión 1.2 o cualquier otra versión posterior publicada por la Free Software Foundation; sin Secciones Invariantes ni Textos de Cubierta Delantera ni Textos de Cubierta Trasera. Una copia de la licencia está incluida en la sección titulada GNU Free Documentation License.