

# Potenciando los entornos lúdicos de programación con las interfaces naturales de usuario

Pedro Damián-Reyes, María Andrade-Aréchiga

Universidad de Colima, Facultad de Telemática,  
Colima, México

{damian, mandrad}@ucol.mx

**Resumen.** En este trabajo se expone la integración de la educación lúdica basada en un lenguaje de programación para niños y las interfaces naturales de usuario (NUI) por medio del sensor Kinect de Microsoft. El objetivo final que se persigue con el proyecto es evaluar el impacto que puede tener la inclusión las NUI en un entorno educativo lúdico de la enseñanza de la computación básica. Como resultado preliminar del proyecto, presentamos un prototipo funcional creado en Visual Studio que utiliza el lenguaje de programación Scratch y el sensor Kinect.

**Palabras clave:** Interacción humano-computadora, interfaces naturales de usuario, educación lúdica apoyada por computadora.

## 1. Introducción

Inglés y computación en todos los niveles educativos del sistema de enseñanza de la educación pública en México es una de las estrategias que han impulsado los gobiernos en nuestro país. Los argumentos son muchos y algunos muy válidos, los datos estadísticos que presentan los especialistas en la materia, demuestran una marcada necesidad del mercado laboral de personas con habilidades en computación e inglés, por lo que resulta oportuna la estrategia impulsada por el gobierno.

Sin embargo, en el ámbito educativo, no es suficiente con establecer leyes que obliguen a las instituciones educativas a implementar programas específicos para garantizar el éxito en los objetivos planteados en la estrategia, sino que deben acompañarse de planes y programas educativos de calidad, que garanticen la adquisición de los conocimientos y habilidades del área abordada. Junto con los planes de calidad se deben diseñar estrategias y herramientas que faciliten el desarrollo de las actividades y el logro de los objetivos de estos planes.

Para lograr este objetivo, en este documento se propone una estrategia en donde se combinan dos grandes áreas: la educación lúdica apoyada por tecnologías y las interfaces naturales de usuario para la enseñanza de la programación básica, gran línea abierta de investigación en el ámbito de tecnología educativa [1, 2], y que es la motivación esencial para el desarrollo de la propuesta.

## 2. Programación para niños

Existen en el mercado una gran cantidad de herramientas enfocadas en la enseñanza de la programación para niños, con un enfoque lúdico. Algunas de las herramientas son:

- Scratch (<https://scratch.mit.edu>). Es una de las plataformas de mayor éxito en la educación inicial de la computación. Fue creada y es mantenida por el grupo Lifelong Kindergarten del laboratorio de medios del MIT. Ayuda a los jóvenes a aprender a pensar creativamente, razonar sistemáticamente, y trabajar colaborativamente — habilidades esenciales para la vida en el siglo XXI. Con Scratch es posible programar historias interactivas, juegos y animaciones — y compartir las creaciones con otros usuarios en la comunidad en línea. Scratch está diseñado especialmente para edades entre los 8 y 16 años, pero es usado por personas de todas las edades. Millones de personas están creando proyectos en Scratch en una amplia variedad de entornos, incluyendo hogares, escuelas, museos, bibliotecas y centros comunitarios. La programación en Scratch se basa en arrastrar y pegar bloques icónicos que representan instrucciones, eventos, objetos, variables
- Simple (<http://www.simplecodeworks.com>). Es un lenguaje de programación para niños. El objetivo es facilitar la escritura de programas de computación por medio de instrucciones sencillas. Se integra de un compilador que transforma las instrucciones de Simple a lenguaje C++ y posteriormente genera el archivo ejecutable. El lenguaje se basa en la modularidad y reutilización de software. Se encuentra disponible sólo para Windows.
- Stencyl (<http://www.stencyl.com>). Es un entorno diseñado para crear juegos sin necesidad de escribir código. El entorno es muy amigable con el usuario y bastante intuitivo. Todas las instrucciones, escenarios, eventos, objetos, etc. son representados con íconos que se arrastran y pegan en el editor. El editor es una extensión del utilizado en Scratch, el cual fue mejorado con cientos de nuevos bloques y funcionalidades listas para utilizarse. Los juegos creados para ser para Windows, Mac, Linux, Android, iOS (iPhone/iPad) y flash.
- Game Maker (<http://www.yoyogames.com/studio>). Es una plataforma para la creación de juegos que garantiza la generación del producto en un 80% menos del tiempo y costo invertido en otras herramientas del mismo tipo. Está diseñada para programadores novatos y expertos en la creación de juegos. El entorno de desarrollo está diseñado para crear juegos sin tener conocimientos de programación. Para los programadores expertos incluye un lenguaje de programación basado en scripts llamado GML, que les permite manejar todos los detalles del juego de manera directa. Se encuentra disponible para Windows, Mac, iOS, Ubuntu, Android, Windows Phone y HTML 5.
- Codecademy (<http://www.codecademy.com>). Es una empresa dedicada a la educación en línea. Uno de sus objetivos es llevar el aula a la Web, creando cursos en línea que los usuarios pueden tomar en el lugar y el momento en que quieran. No es una empresa enfocada particularmente en niños, pero sus cursos son básicos y utilizan una metodología de aprendizaje que los niños pueden seguir.
- Light-Bot (<http://lightbot.com>). Es un juego de rompecabezas que deben resolver utilizando lógica de programación. Su objetivo es introducir principios de programación para niños por medio del juego, en este caso, la resolución de retos por niveles. La aplicación le presenta un reto a resolver y el usuario debe seleccionar las instrucciones, representadas por íconos gráficos, para lograr el

objetivo. Se encuentra disponible para Windows, iPhone/iPad, Android, Mac y Kindle.

- Alice (<http://www.alice.org>). Es un ambiente de programación en 3D creado por la Universidad Carnegie Mellon que permite la creación de historias animadas, juegos interactivos o videos de una manera sencilla. Es una herramienta diseñada para ser el primer punto de encuentro con un estudiante interesado en la programación de computadoras. La estrategia implementada en Alice es el aprendizaje basado en problemas por medio de la creación de películas animadas y videojuegos simples. Los estudiantes utilizan la técnica de arrastrar y pegar objetos gráficos para la creación de programas, los objetos gráficos se corresponden con instrucciones de un lenguaje de programación orientado a objetos como Java, C++ y C#. Corre en sistemas operativos Windows, Mac y Linux.
- Doma de Serpientes para Niños (<http://code.google.com/p/swfk-es>). Es un libro electrónico para usuarios de 8 o más años que quieran aprender a programar. Cubre lo básico de la programación utilizando el lenguaje de programación Python 3 como base para aprender los conceptos. Es necesario instalar Python 3, el usuario debe tener conocimientos básicos para utilizar un procesador de texto plano y dominar el proceso de abrir y guardar archivos.
- KidsRuby (<http://kidsruby.com>). Es una plataforma de aprendizaje para niños utilizando el lenguaje de programación Ruby. El objetivo es crear juegos de manera divertida y sencilla. Se encuentra disponible para Windows, Mac, Linux y Raspberry Pi.
- Hackety Hack (<http://www.hackety.com>). Es un entorno de desarrollo para niños que utiliza el lenguaje de programación Ruby. Su objetivo es enseñar los elementos básicos de programación. El entorno funciona en Windows, Mac y Linux.
- Kodu (<http://www.kodugamelab.com>). Fue diseñado para enseñar a programar con tres conceptos eje: creatividad, resolución de problemas y narrativa. Permite a los niños crear juegos en la PC y el Xbox con un lenguaje de programación visual muy simple. No se requieren habilidades de diseño, ni de programador para poder crear un juego utilizando Kudo. Se encuentra disponible de manera gratuita para los sistemas operativos Windows.
- LEGO MindStorms NXT (<http://www.lego.com/en-us/mindstorms>). Es una plataforma de la marca LEGO que busca fortalecer la creatividad de los niños por medio de la programación de robots. Cuenta con un entorno de desarrollo de programas sencillos y se basa completamente en arrastrar y pegar bloques gráficos que encapsulan instrucciones para que los robots ejecuten acciones específicas. El entorno permite la conexión directa con el robot para instalar el programa y darle vida al producto. Por medio de la programación es posible utilizar los motores, sensores, elementos de comunicación y todos los aditamentos tecnológicos que tiene la marca en sus robots.
- Petit Computer [http://www.nintendolife.com/reviews/dsiware/petit\\_computer](http://www.nintendolife.com/reviews/dsiware/petit_computer). Es una versión del lenguaje de programación BASIC para las consolas Nintendo DSi/3DS. No es un entorno de desarrollo específico para niños, sin embargo, al ser creado para las consolas Nintendo su mercado objetivo son los niños usuarios de este dispositivo de juego.

- Raspberry Pi (<https://www.raspberrypi.org>). Aun cuando no es un entorno de programación esta arquitectura hardware ha facilitado la enseñanza de la programación básica. Es una mini computadora de hardware y software libre de bajo costo. Esta arquitectura ha sido utilizada para enseñar elementos básicos de computación y programación. Se considera factible presentar la arquitectura y funcionamiento general de una computadora por medio de la interacción directa con el hardware y los conceptos de programación se utilizan al programar esa computadora. La capacidad de ésta permite instalar el sistema operativo Linux y el lenguaje de programación C, C++.

### 3. Scratch como herramienta lúdica

La educación lúdica apoyada por tecnología ha demostrado tener un impacto positivo en los estudiantes, como lo señalan numerosos estudios que han sido realizados. Ejemplo de esto son los que se basan en el uso de juegos serios en contextos educativos cuyos resultados son muy positivos [1, 2, 3, 4, 5].

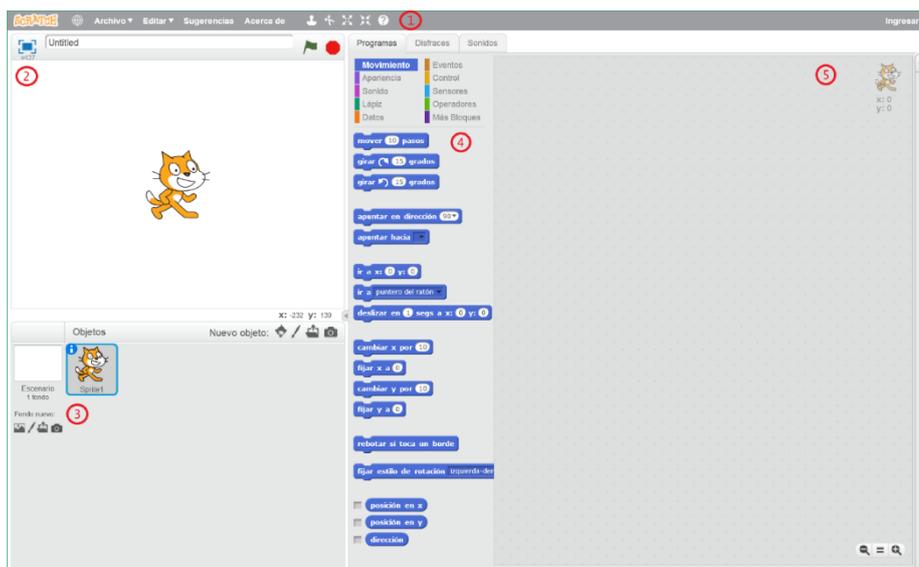


Fig. 1. Interfaz de Scratch.

Lo anterior nos permite asegurar que una estrategia para fortalecer la educación de la programación básica consiste en utilizar una herramienta que le permita al estudiante sentirse partícipe de un juego, y no de un entorno educativo. Es en este punto en donde entra el valor del lenguaje Scratch, ya que este lenguaje se basa en la creación de aplicaciones de software como si de un juego se tratase, aunque en realidad fue diseñado para crear historias animadas y juegos de una manera muy sencilla, haciendo sentir al programador que es parte de un juego.

El lenguaje Scratch ha sido ampliamente utilizado en ambientes educativos, como lo demuestran numerosos proyectos a nivel internacional, por ejemplo, la

implementación en instituciones educativas de Cali, Colombia [1], la introducción en diferentes naciones en desarrollo como India, Africa, Ecuador, Nicaragua, Paraguay, Uruguay, entre otros [2], en el sistema educativo de la República Bolivariana de Venezuela [3], también las instituciones de educación de naciones desarrolladas utilizan Scratch como herramienta educativa, por ejemplo, Kennedy-Longfellow School in Cambridge, Massachusetts [4], entre muchas otras. Es tanta la popularidad del producto que se organiza en el mundo un evento anual llamado “Scratch Day”

Como es posible darse cuenta, Scratch es una poderosa herramienta para la educación de la programación básica. Su funcionamiento se considera muy sencillo, y por ello se describen los principales elementos que integran este ambiente de desarrollo.

El ambiente de desarrollo puede ejecutarse en línea o descargarse a la computadora (<https://scratch.mit.edu/scratch2download>), los sistemas operativos de la versión para computadora son Windows, Mac OS y Linux.

Para el presente caso, se optó por utilizar la versión Web debido a que se puede integrar en un navegador desde una aplicación desarrollada en Visual Studio y utilizar la programación con el sensor Kinect.

El entorno de desarrollo es muy agradable a la vista, como se puede apreciar en la Fig. 1.

Las partes que lo integran son:

1. Barra de menú textual, permite seleccionar las opciones generales relacionadas con la manipulación de archivo (nuevo, guardar, subir de computadora, descargar a computadora, revertir), con el proceso de edición del proyecto (recuperar borrado) y las clásicas opciones de sugerencias y acerca de. También incluye unos íconos de algunas acciones comunes al momento de escribir un programa (duplicar, borrar, crecer, encoger y ayuda).
2. Interfaz de usuario, en ella se visualiza el funcionamiento del proyecto en ejecución y permite la interacción con el usuario de la aplicación. También se localizan el botón de ejecución que corresponde a la bandera verde y el de detener la ejecución con el ícono rojo.
3. Ventana de objetos, se muestran los elementos que se incluyen en la aplicación que corresponde a objetos y escenarios. Los objetos son los elementos que le dan vida a la aplicación, corresponden a lo que es una instancia de objeto en la POO, tienen propiedades y comportamiento basado en eventos. Los escenarios son un tipo especial de objeto que se utiliza como fondo de la aplicación, cuenta con las mismas características de los objetos. Scratch proporciona una biblioteca de objetos predefinidos que pueden ser utilizados, o el desarrollador puede crear sus propios objetos, incluso puede tomar una foto con la cámara de la computadora para diseñar sus objetos.
4. Elementos de programación, en esta sección se muestran los diferentes bloques que son utilizados para escribir la programación de la aplicación, pueden ser: instrucciones de movimiento, de apariencia, de sonido, de control, entre otros. Cada categoría presenta bloques diferentes que se traducen en instrucciones, eventos, acciones que representan las reglas de negocio de la aplicación.
5. Editor de código, es el área en donde se colocan los bloques de programación para la generación de la aplicación.

Para crear una aplicación se debe seleccionar el objeto y agregar los códigos correspondientes, por ejemplo, si se quiere crear la clásica aplicación “¡Hola mundo!” es necesario hacer lo siguiente:

1. Seleccionar el objeto principal localizado en la ventana de objetos (3 en Fig. 1), en nuestro caso el gatito llamado “Sprite1”.
2. En el panel de elementos de programación (4 en Fig. 1), seleccionar la clasificación “Eventos”, arrastrar al panel de edición de código (5 en Fig. 1) el bloque llamado “al presionar (bandera verde)”. Este bloque de código es el inicio del programa.
3. Seleccionar la clasificación “Apariencia”, arrastrar el bloque “decir Hello” y modificar la palabra “Hello” por “¡Hola mundo!”.
4. Ejecutar la aplicación haciendo clic en el ícono de la bandera verde en la interfaz de usuario (2 en Fig. 1).

#### **4. Interfaces naturales de usuario**

Las interfaces naturales de usuario (en inglés Natural User Interface o NUI) son aquellas que permiten la interacción humano computadora sin utilizar dispositivos de entrada que el usuario deba tocar físicamente, en lugar de ello el usuario hace uso de movimientos gestuales con su cuerpo para realizar la interacción, por ejemplo, movimiento de las manos, del cuerpo o el sonido. Las NUI proporcionan la capacidad de interactuar con una máquina usando únicamente el cuerpo humano [10].

En la actualidad existen varios dispositivos que son utilizados para implementar las NUI, por ejemplo:

1. Leap Motion Controller (<https://www.leapmotion.com>), es una pequeña barra que se conecta a la computadora y detecta el movimiento de las manos para interactuar con la máquina.
2. Myo Gesture Control Armband (<https://www.thalmic.com/myo>), es una banda que se coloca en el antebrazo y por medio de los movimientos en el brazo realiza la interacción con la computadora.
3. Google Glass (<https://developers.google.com/glass>), son unos lentes creados por Google por medio de los cuales los usuarios solicitan y reciben información o servicios utilizando su voz, localización, video, entre otras variables que los lentes perciben.
4. Microsoft HoloLens (<https://www.microsoft.com/microsoft-hololens/en-us>), es la primera computadora holográfica sin ataduras que permite combinar el mundo real, con hologramas de alta definición, para generar una experiencia de realidad aumentada mixta.
5. Microsoft Kinect Sensor, es un dispositivo que integra una gran cantidad de sensores que detectan el movimiento corporal del usuario que se coloca enfrente de él, al mismo tiempo captura el sonido permitiendo la interacción utilizando instrucciones de voz (<https://www.microsoft.com/en-us/kinectforwindows>).

Particularmente el sensor Kinect de Microsoft ha sido utilizado en muchos proyectos relacionados con la educación, el entretenimiento y la salud, como se puede

constatar en la gran variedad de aplicaciones que han sido desarrolladas con objetivos didácticos [11].

En el presente caso se seleccionó el sensor Kinect porque ofrece una plataforma de desarrollo accesible y fácil de utilizar, ya que se integra de manera natural con el Visual Studio. El Visual Studio es la plataforma de desarrollo de software de Microsoft.

Para iniciar con el desarrollo de aplicaciones que utilicen el sensor Kinect se debe contar con los siguientes requisitos:

1. Sensor Kinect de Microsoft.
2. Visual Studio 2013 o superior (<https://www.visualstudio.com>).
3. Kinect para Windows SDK 2.0 (<https://www.microsoft.com/en-us/download/details.aspx?id=44561>).

## **5. Scratch utilizando una NUI**

Se desarrolló una aplicación en donde se integran el lenguaje Scratch y el sensor Kinect para la manipulación de la interfaz.

La aplicación despliega el entorno de desarrollo de Scratch y el usuario escribe un programa utilizando el sensor Kinect, como se puede apreciar en la Fig. 2.



**Fig. 2.** Aplicación con Scratch y Kinect.

Se planteó un escenario en donde se pretende medir la facilidad de uso de la aplicación tomando como base la enseñanza de algunos conceptos básicos de programación, tales como:

1. Objetos, identificar el concepto de objeto en el desarrollo de una aplicación de software.
2. Secuencialidad, conocer la ejecución secuencial de instrucciones para lograr un objetivo.

3. Sentencia condicional, aplicar de manera correcta la instrucción condicional Si...entonces... (if...else...) en el desarrollo de software.
4. Sentencia cíclica, utilizar de manera adecuada sentencias cíclicas del tipo mientras... (while...).

Para lograr lo anterior, al usuario se le pedirá que elabore los siguientes programas:

1. Hacer que el gatito avance 100 pasos, gire 90 grados hacia la derecha, avance 100 pasos, gire 90 grados hacia la derecha, avance 100 pasos, gire 90 grados hacia la derecha, avance 100 pasos y finalmente gire 90 grados. Con este ejercicio se pretende abordar los conceptos de objeto y secuencialidad.
2. Para aplicar los conceptos de una instrucción cíclica se realizará lo siguiente, repetir 5 veces el algoritmo del ejercicio 1, incrementando el número de pasos en 10 en cada vuelta completa del gatito.

## **6. Conclusiones**

Con la implementación de la aplicación presentada en este documento hemos logrado manipular un lenguaje de programación lúdico con una interfaz natural de usuario. Al incluir una NUI se espera incrementar el atractivo de la utilización del lenguaje Scratch, y con ellos apoyar la enseñanza de los conceptos básicos de computación.

Aún estamos en la etapa de evaluación de usabilidad de la aplicación y tenemos planeada la realización de un experimento en donde se valore el nivel de aprendizaje de los conceptos establecidos utilizando la herramienta y sin utilizarla. Además de comprobar qué tanto el uso del kinect hace la diferencia en cuanto a la facilidad de aprendizaje del Scratch.

## **Referencias**

1. Sáez López, J.M., Dominguéz Garrido, C.: Integración pedagógica de la aplicación MineCraft EDU en educación primaria: un estudio de caso, *Píxel-Bit. Revista de medios y educación*, nº 45, pp. 95–110 (2014)
2. Backlund, P., Hendrix, M.: Educational Games - Are They Worth The Effort? *International Conference on Games and Virtual Worlds for Serious Applications (VS-Games)*, Msida, Malta (2013)
3. Kirriemuir, J., Mcfarlane, A.: Literature Review in Games and Learning. A NESTA Futurelab Research report - report 8 (2004)
4. Marsh, T.: Serious games continuum: Between games for purpose and experiential environments for purpose. *Entertainment Computing*, Vol. 2, No. 2, pp. 61–68 (2011)
5. Guillén-Nieto, V., Aleson-Carbonell, M.: Serious games and learning effectiveness: The case of It's Deal! *Computer & Educations*, Vol. 58, No. 1, pp. 435–448 (2012)
6. López García, J.: eduteka. [En línea]. Available: <http://www.eduteka.org/modulos.php?catx=9&idSubX=278> (2015)
7. Team, S.: ScratchEd. [En línea]. Available: <http://scratched.gse.harvard.edu/stories/scratch-around-world-part-1-4-developing-scratch-developing-nations> (2015)
8. Eduteka, Scratch Venezuela: [En línea]. Available: <http://www.scratchvenezuela.com/>. (2015)

9. Team, S. ScratchEd: [En línea]. Available: <http://scratched.gse.harvard.edu/stories/cambridge-scratch-educator-meetups> (2015)
10. Wigdor, D., Wixon, D.: Brave NUI World: Designing Natural User Interfaces for Touch and Gesture. Morgan Kaufmann (2011)
11. Microsoft: Kinect for Windows, Microsoft. [En línea]. Available: <https://www.microsoft.com/en-us/kinectforwindows/meetkinect/gallery.aspx?searchv=education> (2015)
12. Expansión.com: Unidad Editorial Información Económica, S.L., [En línea]. Available: <http://www.expansion.com/2011/03/29/empleo/mercado-laboral/1301389344.html> (2015)
13. Financiero, E.: El Financiero, Grupo Multimedia Lauman, SAPI de CV: [En línea]. Available: <http://www.elfinanciero.com.mx/economia/el-56-de-profesionistas-mexicanos-tienen-nivel-medio-de-computacion.html> (2015)