

PROGRAMACIÓN CREATIVA

Alberto Pacheco González, Edgar Trujillo Preciado, José Alberto Domínguez Terrazas
Instituto Tecnológico de Chihuahua
División de Estudios de Posgrado e Investigación
Av. Tecnológico 2909
Col. 10 de Mayo, Chihuahua, Chih. CP 31310
{apacheco, edgar.trujillo, joaldote}@itch.edu.mx

RESUMEN.

Se presentan diversas estrategias didácticas relacionadas con el campo emergente de la programación creativa, con la intención de incentivar y mejorar la enseñanza de la programación. Algunas de estas herramientas permiten rescatar la relación simbiótica entre el software y hardware de un sistema usando tarjetas de bajo costo como Arduino, Intel Edison y Raspberry Pi. Existen también otras posibilidades que facilitan la práctica de la programación en vivo desde un navegador de internet en entornos mas visuales e interactivos. Finalmente, se evalúan los méritos y las limitantes de las herramientas revisadas y aplicadas en clase.

Palabras clave: alfabetismo digital, programación creativa.

ABSTRACT.

A set of creative programming tools are reviewed according to its learning outcomes. Some of these tools, as Arduino and Rapsberry Pi devices enable students to interact directly with electronic and computer hardware. Other tools support live-coding practices using interactive visualization. Finally, the merits and limitations of these tools, as well as its respective learning impact are considered.

Keywords: digital literacy, creative coding, live coding.

1. INTRODUCCIÓN

En la sociedad de la información y el conocimiento, aprender a programar ha dejado de ser asunto exclusivo de los especialistas en sistemas e informática [1]. La ubicuidad computacional actual sobrepasa la noción de la computadora “de escritorio”. Existe una extensa gama computacional, que abarca desde aquellos dispositivos personales y embebidos (e.g. reloj, teléfono, TV, cámara, lentes, sensores, etc.), hasta la más compleja maquinaria (e.g. auto, robot, avión, barco, etc.). Ante un mercado creciente de dispositivos “inteligentes” (e.g. *smart phone*, *smart watch*, *smart car*, *smart TV*, *smart sensors*, *smart clothes*), es el software quien también juega un rol clave para brindar dicha “inteligencia” al hardware.

Saber programar y diseñar software requiere tanto habilidades del pensamiento abstracto, lógico y algorítmico, como la puesta en práctica de una gran diversidad de competencias digitales (fig. 1), que aunadas al manejo de computadoras, dispositivos móviles, uso de redes sociales, *apps*, búsquedas, manejo de recursos y medios digitales. Esta adaptación continua y dinámica al medio digital ha dado pie a una nueva brecha, el analfabetismo digital, informacional y funcional [2]. Un alfabetista digital tiene la capacidad práctica de adquirir, analizar, procesar, crear artefactos, software, documentos,

medios y recursos digitales de manera diestra y productiva en su propia vida utilizando una gran diversidad de entornos digitales y dispositivos “inteligentes” propios del entorno digital de la sociedad del conocimiento [3].

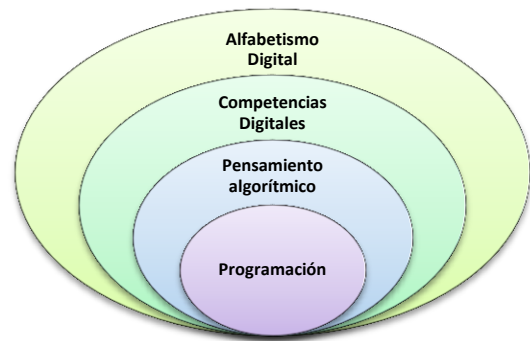


Figura 1. La programación dentro del contexto amplio del alfabetismo digital y las habilidades digitales [1].

2. PROGRAMACIÓN CREATIVA

La forma tradicional de enseñar a programar se orienta en gran medida a la formación de profesionistas en sistemas e informática, dejando muchas veces de lado la posibilidad de incorporar personas de cualquier edad o profesión. La programación creativa es un enfoque didáctico incluyente, dinámico, flexible e interesante que promueve el uso del software como un medio para representar expresiones originales y creativas, combinando un enfoque algorítmico de análisis, diseño y resolución de problemas, que termine por generar una solución creativa y factible de ser implementada en software [4].

En sus inicios, la programación creativa surgió bajo un enfoque orientado al arte generativo [5], que atrajo a artistas, diseñadores, arquitectos y músicos interesados en extender y aplicar sus habilidades creativas. Las más recientes tendencias comienzan a incorporar técnicas didácticas, aplicaciones y lenguajes de programación más accesibles.

La programación creativa se caracteriza también por aplicar hasta el límite de las posibilidades tecnológicas tanto los medios digitales (visualización, multimedia, simulación, música, etc.) como las diversas posibilidades de interacción multimodal (texto, voz, superficies táctiles, gestos faciales o corporales, etc.) donde se obtiene una retroalimentación más inmediata, interactiva y multimedia que promueve la

experimentación, curiosidad y creatividad sin la necesidad de escribir demasiado código complejo y reduciendo la curva de aprendizaje y el bagaje de conocimientos técnicos [4, 5].

A finales de la década pasada, las oportunidades de empleo en las áreas de Tecnologías de Información (TI) crecieron de manera constante, según un estudio del Depto. de Empleos de Estados Unidos predice que entre 2008 y 2018 existirán 1.4 millones de empleos relacionados con TI [6]. En años recientes, se han presentado diversas iniciativas para promover el aprendizaje de la programación [7]. La disponibilidad del código abierto, la proliferación de plataformas de programación en línea, la práctica de la programación en vivo [8], la robótica, el desarrollo de proyectos de hardware con Arduino, Raspberry Pi, Intel Curie, Galileo y Edison entre otros, ofrecen en su conjunto, un panorama muy prometedor y atractivo para comenzar a programar desde edades cada vez más tempranas.

3. PROBLEMÁTICA

El Tecnológico Nacional de México, en sus 266 planteles, forma profesionistas en las áreas de Ingeniería. Programación es una materia básica del programa de estudios. Desafortunadamente, dentro de nuestro plantel, en los últimos 5 años, ha existido un alto índice de reprobación, alcanzando un valor promedio de 31% y una tendencia lineal de 33 a 30% (fig. 2).

Período	Alumnos cursando la materia	Aprobados	Reprobados	Índice de reprobación (%)
ene-jun 2010	472	302	170	36%
ago-dic 2010	429	297	132	31%
ene-jun 2011	269	177	92	34%
ago-dic 2011	400	294	106	27%
ene-jun 2012	426	290	136	32%
ago-dic 2012	455	309	146	32%
ene-jun 2013	428	304	124	29%
ago-dic 2013	489	359	130	27%
ene-jun 2014	434	267	167	38%
ago-dic 2014	485	354	131	27%
Promedios	429	295	133	31%

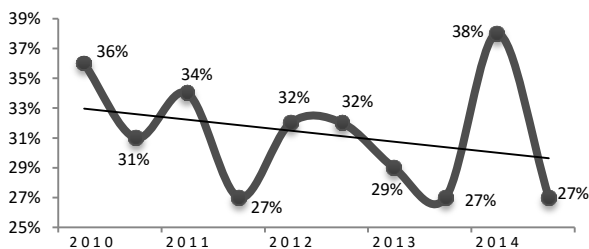


Figura 2. Índice de reprobación en Programación 2010-2014.

Durante la era del sistema operativo Microsoft DOS® para la computadora PC (1981-1995), era posible elaborar programas que accedieran los recursos de *hardware* de la computadora, e.g. puertos, memoria, interrupciones, video, teclado, red, etc. Sin embargo, a partir del sistema operativo Windows® 95 de Microsoft, se eliminó el acceso directo al *hardware*, perdiéndose con ello una excelente oportunidad formativa gracias a la relación simbiótica *hardware-software*.

4. ADOPTANDO LA PROGRAMACIÓN CREATIVA.

Convencidos de las bondades de la programación creativa para elevar la motivación, rendimiento y puesta en práctica, un grupo de maestros de programación de nuestra institución inició una exploración de diversas estrategias de aprendizaje, mismas que se presentan a continuación.

4.1. Computadora Raspberry Pi

En 2006 surge el primer diseño de la computadora Raspberry Pi [9] y en 2012 se lanza a la venta la primera placa SBC. En 2009, se crea la Fundación Raspberry Pi para “promover el estudio de las Ciencias de la Computación y temas relacionados, sobre todo a nivel escolar básico, y para recuperar la diversión de aprender computación”. Raspberry Pi es una computadora de bajo costo (US\$35) que abarca desde un modelo A (2012) de 256Mb de RAM hasta más reciente modelo Pi 2 B (2015) con 1Gb de RAM, un ARM quad-core Cortex-A7 de 900MHz. Todos los modelos corren bajo Linux (Raspbian) y se programa en Python, C y Ruby, además de brindar entornos muy didácticos para aprender a programar como Scratch y usando música con Sonic Pi.

4.2. La Plataforma Arduino

El estudiante italiano Massimo Banzi creó Arduino en 2005 [10] como una plataforma *open-hardware* flexible, de bajo costo y fácil de usar para el desarrollo de prototipos de electrónica. Basado en microcontroladores Atmel AVR, su tarjeta y configuración se han convertido en un estándar de la industria. El verdadero poderío de Arduino no es su hardware, sino su software —el IDE de Arduino. Varias universidades como Stanford, Carnegie Mellon y MIT usan Arduino en sus clases y Google mismo ha desarrollado el Kit ADK para enlazar un dispositivo Android con la plataforma Arduino.

4.3. Intel Galileo, Edison y Curie

Galileo gen. 2 es una tarjeta similar a Raspberry Pi con una sub-arquitectura Intel Pentium de 32 bits [11], sus puertos son compatibles con Arduino y ofrece un IDE y un SDK para IoT (Internet de las cosas). Otras tarjetas de Intel son: Edison basado en Intel Atom y el chip Curie del tamaño de un botón para soportar el desarrollo de *smart clothes* y *smart sensors*.

4.4. El Robot Finch

Lauwers y Nourbakhsh de la universidad de Carnegie Mellon han creado un robot programable via puerto USB y una serie de recursos didácticos para apoyar la enseñanza de la programación y las Ciencias de la Computación aceptando una docena de lenguajes y entornos de programación [12, 13].

4.5. El Lenguaje de Programación Processing

Processing fue creado en 2001 por los estudiantes Ben Fry y Casey Reas en MIT Media Lab [4]. Es un proyecto de código abierto pensado en la enseñanza de la programación dentro de un contexto visual, creativo y muy interactivo. Existen una gran comunidad de desarrolladores, usuarios, librerías y varios dialectos del lenguaje en Java, Javascript y Python. Es también destacable que el entorno de Arduino y Khan Academy usan alguna variante del lenguaje Processing.

4.6. Khan Academy

En 2006, Salman Khan publica lecciones de matemáticas grabadas en videos en el canal Khan Academy de YouTube. Debido al gran éxito obtenido, en 2009 lanza una plataforma abierta que contiene más de 6,500 videos de matemáticas, física, química, biología, computación, finanzas, economía, arte y humanidades en 36 idiomas, más de 27 millones de usuarios registrados de 200 países consultan más de 580 millones de lecciones y realizan alrededor de 3,800 millones de ejercicios bajo la supervisión de 350,000 maestros registrados [14]. En 2012 libera un entorno de programación basado en Processing provisto de lecciones de programación. Dicho entorno y sus lecciones fueron diseñadas para facilitar el aprendizaje de la programación y actualmente participan más de 1'300,000 jóvenes al mes [15].

4.7. Marionetas Digitales

El uso de marionetas puede ser de gran ayuda para narrar historias y ser a la vez un medio muy interesante para el aprendizaje [16]. El uso adecuado de las marionetas puede representar para los estudiantes un medio lúdico, elocuente, y flexible para explorar y demostrar lo aprendido. Una presentación basada en marionetas realizada por estudiantes puede sintetizar lo aprendido dentro de una narrativa de una forma artística, dramática y multimodal (fig. 5).

4.8. La Hora del Código

La iniciativa de Code.org denominada HourOfCode.com plantea una serie de retos y entornos didácticos donde han llegado a participar más de 60 millones de estudiantes de todas las edades [17]. Entre las plataformas que han participado en Code.org figuran: studio.code.org, Khan Academy, Codecademy, Google CF First, Blockly Games, Kodable, Scratch onli-ne, etc. Los retos incluyen personajes de películas y videojuegos, tales como: Plantas vs. Zombies™, Angry Birds™, Flappy Bird, Frozen™ y Disney Infinity Games™.

5. RESULTADOS DE LA PUESTA EN PRÁCTICA

Partiendo del objetivo de desarrollar las competencias para las ciencias fundamentales y la ingeniería desde un punto de vista práctico y buscando motivar al estudiante a interactuar por medio de la programación con el mundo real, diversos estudios sobre la enseñanza de la informática han demostrado que la construcción de prototipos y el ejercicio en equipo aceleran el aprendizaje del alumno [18].

5.1. Prácticas con Arduino

Para ello se han diseñado prácticas basadas en Arduino [19], como por ejemplo, para la lectura de un sensor de temperatura, reportando datos a la computadora PC por puerto USB (fig. 3). En esta práctica participaron 6 alumnos de un grupo de programación para la carrera de electrónica y electromecánica. Ellos aplicaron sus conocimientos de programación (expresiones y ciclos) y electrónica (sensores, convertidores, puerto serie, acondicionamiento de señal, etc.), partiendo de un kit Arduino de bajo costo y usando una porción pequeña de código escrito en un lenguaje de programación de alto nivel (fig. 4). En este primer experimento fue posible observar que la práctica consumió mayor tiempo que un ejercicio basado exclusivamente en software, e.g. adquirir componentes, alambrar circuito, investigar configuración de puertos, etc. Sin embargo, se obtuvo una experiencia de aprendizaje más formativa, cooperativa, completa e integral.

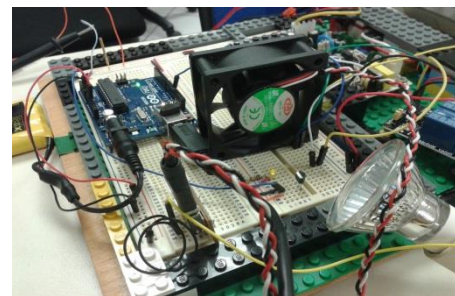


Figura 3. Circuito de la práctica del sensor de temperatura.

5.2. Marionetas Digitales Programables

Usando un entorno para programar marionetas digitales con Processing (fig. 5), derivado del entorno de la plataforma de Khan Academy [20], con un grupo de 15 estudiantes de segundo semestre de Ingeniería Industrial en la modalidad de Educación a Distancia, se logró cubrir en menor tiempo las primeras unidades (50% menos), como fue el caso del tema de datos primitivos, constantes y variables, que como se aprecia en la (fig. 5), mediante un programa muy simple fue posible fijar una posición en los brazos de la marioneta (constantes expresadas como ángulos de rotación) y mover dicha marioneta según la posición del cursor (variables mouseX/Y para obtener posición del mouse).

```
temp-vent
1
2 float temperatura = 0; //variable tipo flotante para la temperatura
3 void setup()
4 {
5   Serial.begin(9600); //inicia comunicacion serial
6   pinMode(12,OUTPUT); //configuracion del pin 12 como salida para encender ventilador
7 }
8
9 void loop()
10 {
11   //Calcula la temperatura usando como referencia 5v
12   temperatura = (5.0 * analogRead(0)*100.0)/1023.0;
13   Serial.println(temperatura); //escribe la temperatura en el serial
14   delay(1000); //espera 3 segundos para la siguiente medicion
15   //esto enciende y apaga el ventilador
16   if (temperatura > 32.0) //cambiar numero en funcion de lo que se desee
17     digitalWrite(12, LOW);
18   else
19     digitalWrite(12, HIGH);
```

Figura 4. Código en C para lectura de la temperatura.

```
var avatar = new Avatar("pantera");
avatar.rotate(1,40);
avatar.rotate(3,-30);
var draw = function() {
  background(200);
  avatar.draw(mouseX, mouseY);
};
```

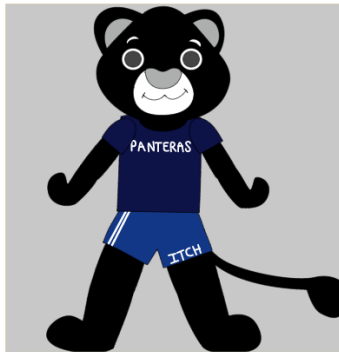


Figura 5. Programando una marioneta digital.

6. CONCLUSIONES.

El aprender a programar no es una tarea sencilla para un estudiante principiante. El abandono y el nivel de reprobación de este tipo de cursos es bastante alto. A pesar de que pueden ser muchas las razones que pueden influir en esta situación, creemos que pueden abatirse estos indicadores buscando estrategias didácticas y herramientas para motivar, enriquecer y diversificar el aprendizaje.

En este primer trabajo se demostró que existen alternativas muy diversas que van desde la construcción de prototipos de hardware hasta la manipulación de avatars o robots. Dichas actividades no solo han llamado la atención de los alumnos, sino que han resultado ser mas productivas en la formación profesional del alumno distinguiéndose sobretodo por el hecho de representar un reto al alumno, presentar casos más cercanos a un problema real y por que demandan en buena medida la cooperación de varios estudiantes para implementar una solución creativa al problema.

Este trabajo esta aún en progreso y se espera desarrollar una serie de prácticas y actividades de aprendizaje aplicando los principios de la programación creativa y desde luego, desarrollar los instrumentos más adecuados para evaluar el posible impacto en el aprendizaje del estudiante, esperando desde luego, que a mediano plazo, se tenga una disminución en el índice de reprobación y deserción en dicha asignatura.

7. REFERENCIAS.

- [1] Wojcicki, E.: Should My Kid Learn to Code? *The HuffPost Education*. [Online]. Disponible desde 14/07/2015: <http://huff.to/1MgHv15>
- [2] Martí, M., D'Agostino, M., Veiga de Cabo, J., Sanz-Valero, J.: Alfabetización Digital: un peldaño hacia la sociedad de la información, *Medicina y Seguridad del Trabajo*, Vol. 54(210), pp. 11–15, Mar. 2008.
- [3] Ferreira, E.: Alfabetización digital. ¿De qué estamos hablando?, *Educação e Pesquisa*, 2011. [Online]. Available: <http://www.redalyc.org/articulo.oa?id=29819096014>
- [4] I. Greenberg, D. Xu, and D. Kumar, *Processing: Creative Coding and Generative Art in Processing 2*. Apress, 2012.
- [5] C. Roberts, M. Wright, J. Kuchera-Morin, and T. Höllerer, "Gibber: Abstractions for Creative Multimedia Programming," in *Proc. of the ACM Int. Conf. on Multimedia*, New York, NY, USA, 2014, pp. 67–76.
- [6] U.S. Department of Labor, Bureau of Labor Statistics, 2011
- [7] Code.org: Blurbs and Stats, 2014. Disponible en: <http://hourofcode.com/resources/stats>
- [8] Rubin, M. J.: The effectiveness of live-coding to teach introductory programming. En: 44th ACM technical symposium on Computer science education (SIGCSE), pp. 651–656. New York, NY: ACM Press, 2013. DOI: 10.1145/2445196.2445388
- [9] Raspberry Pi, <http://www.raspberrypi.org/>
- [10] Arduino, <http://arduino.cc/>
- [11] Intel Galileo Gen 2 <http://www.intel.es/content/www/es/es/do-it-yourself/galileo-maker-quark-board.html>
- [12] Lauwers, T, Nourbakhsh, I.: "Designing the Finch: Creating a Robot Aligned to Computer Science Concepts", Proceedings of the Twenty-Fourth AAAI Conf. on Artificial Intelligence, Atlanta, 2010, 1902-1907
- [13] BirdBrain Tech, The Finch Robot, 2010, <http://www.finchrobot.com>
- [14] Morrison, B. B., DiSalvo, B.: Khan academy gamifies computer science. En: 45th ACM Technical Symposium on Computer Science Education (SIGCSE), pp. 39–44. New York: ACM Press. 2014
- [15] Resig, J.: Redefining the Introduction to Computer Science. Disponible desde 2012 en: <http://ejohn.org/blog/introducing-khan-cs>
- [16] Leite, L. M., Orvalho, V.: Inter-Acting: Understanding interaction with performance-driven puppets using low-cost optical motion capture device. En: *Int. Journal of Adv. Computer Science*, 3(2). 2013. Recuperado de <http://goo.gl/KiGle4>
- [17] Code.org: Blurbs and Stats. Disponible desde 2014 en: <http://hourofcode.com/resources/stats>
- [18] R. Balasubramania, Hands-on Introduction to Computer Science at the Freshman Level, Proceeding *SIGCSE '14* Proceedings of the 45th ACM technical symposium on Computer science education, Pages 235-240, ACM New York, NY, USA
- [19] Trujillo, E.: Práctica de Programación: Un sensor de temperatura. Disponible desde 2015 en: <http://home.itch.edu.mx/~edgar.trujillo/docs/programacion>
- [20] Pacheco, A., Gonzalez, E.: Marionetas Digitales Programables, Memorias del Encuentro Nacional en Computación (ENC), Ensenada, Baja California, México, 2015.