

Teaching the Introduction to Programming using Ludic Tools

Leticia Laura-Ochoa, Mg¹

¹Universidad Nacional de San Agustín, Perú, llaurao@unsa.edu.pe

Abstract– *This article describes the experience in teaching the introduction to programming to students of the Professional School of Medicine of the National University of San Agustín using ludic tools such as Scratch, Lightbot and App Inventor, which have allowed the understanding of the basic concepts of programming in an easy and entertaining way, motivating the learning process of the students. The use of visual programming tools allowed the students to create their own games and creative applications related to their professional career, strengthening the resolution of problems and favoring their active learning.*
Keywords- *teaching-learning, active learning, playful tools, programming, digital literacy, game-based learning*

Digital Object Identifier (DOI):<http://dx.doi.org/10.18687/LACCEI2018.1.1.176>
ISBN: 978-0-9993443-1-6
ISSN: 2414-6390

Enseñanza de la Introducción a la Programación utilizando Herramientas Lúdicas

Leticia Laura-Ochoa, Mg¹

¹Universidad Nacional de San Agustín, Perú, llaurao@unsa.edu.pe

Resumen— Este artículo describe la experiencia en la enseñanza de la introducción a la programación a estudiantes de primer año de la Escuela Profesional de Medicina de la Universidad Nacional de San Agustín utilizando herramientas lúdicas como Scratch, Lightbot y App Inventor, las cuales han permitido la comprensión de los conceptos básicos de programación de manera fácil y entretenida, motivando el proceso de aprendizaje de los estudiantes. El uso de herramientas de programación visual permitió a los estudiantes crear sus propios juegos y aplicaciones creativas afines a su carrera profesional fortaleciendo la resolución de problemas y favoreciendo su aprendizaje activo.

Palabras claves— enseñanza-aprendizaje, aprendizaje activo, herramientas lúdicas, programación, alfabetización digital.

Abstract— This article describes the experience in teaching the introduction to programming to students of the Professional School of Medicine of the National University of Saint Augustine using playful tools such as Scratch, Lightbot and App Inventor, which have allowed the understanding of the basic concepts of programming in an easy and entertaining way, motivating the learning process of the students. The use of visual programming tools allowed the students to create their own games and creative applications related to their professional career, strengthening the resolution of problems and favoring their active learning.

Keywords— teaching-learning, active learning, playful tools, programming, digital literacy.

I. INTRODUCCIÓN

Se ha convertido en algo común referirse a los jóvenes como “nativos digitales” debido a su aparente fluidez con las tecnologías digitales [1],[2]. Los nativos digitales representan las primeras generaciones que crecen con esta nueva tecnología, pasan su vida rodeados y usando computadoras, videojuegos, reproductores de música digital, cámaras de video, teléfonos celulares y todos los demás juguetes y herramientas de la era digital [1]. De hecho, muchos jóvenes se sienten muy cómodos enviando mensajes de texto, jugando en línea y navegando por la Web; pero, ¿eso realmente los hace fluidos con las nuevas tecnologías? Aunque interactúan con los medios digitales todo el tiempo, pocos son capaces de crear sus propios juegos, animaciones o simulaciones. Es como si pudieran “leer” pero no “escribir” [2]. Los estudiantes universitarios de hoy en día utilizan frecuentemente estas tecnologías digitales; pero son muy pocos los que pueden crear sus propios juegos y aplicaciones.

La “programación” juega un papel fundamental en este proceso de creación e invención. Saber programar amplía las posibilidades del uso de las tecnologías, nos permite crear nuevos contenidos y herramientas y, en general nos habilita a construir conocimientos a través del uso [3]. En una sociedad digitalizada que presenta cambios acelerados a causa del uso masivo de las Tecnologías de la Información y Comunicación (TIC), se demandan estudiantes competentes, creativos, que posean conocimientos en tales campos, a pesar de que se considera que es complejo de aprender [4]. La enseñanza de la programación en la educación superior no se restringe únicamente a estudiantes de carreras afines a la computación, sino también es aplicable a cualquier otra disciplina, por lo que es recomendable que en los cursos de informática se incluya temas de introducción a la programación donde los estudiantes puedan desarrollar su pensamiento computacional y crear sus propias aplicaciones.

El alto nivel de abstracción y la complejidad de los conceptos que deben ser aprendidos para programar es un impedimento serio para muchas personas. Para la mayoría programar es una actividad compleja y aburrida [5]. El problema se resolvería si pudiéramos hacer la programación introductoria fácil y entretenida para los estudiantes [6]. Según [6], los lenguajes inicialmente utilizados resultaban de difícil comprensión para los niños y jóvenes por su sintaxis, y la programación era por lo general introducida con actividades desconectadas de los intereses de los destinatarios, en contextos donde no se ofrecía guía frente a los errores, ni se promovía una mayor exploración.

La selección del primer lenguaje de programación es de vital importancia, pues determina el énfasis que tendrá el curso. En efecto, el lenguaje de programación determinará el énfasis que pondrá el docente en la sintaxis del lenguaje, pudiendo ser reducida esta dificultad. Al mismo tiempo debemos procurar que los programas realizados sean atractivos para los alumnos, no sólo en lo relacionado con aspectos visuales, sino que también en su utilidad [7]. Es un desafío para los docentes que imparten cursos introductorios de programación encontrar herramientas y estrategias de enseñanza que permitan motivar a los estudiantes en su proceso de aprendizaje.

Fomentar la capacidad de programación de computadoras de los estudiantes ha sido reconocido como un tema educativo importante y desafiante que está muy relacionado con el proceso de resolución de problemas. Sin embargo, la mayoría

Digital Object Identifier (DOI): <http://dx.doi.org/10.18687/LACCEI2018.1.1.176>

ISBN: 978-0-9993443-1-6

ISSN: 2414-6390

16th LACCEI International Multi-Conference for Engineering, Education, and Technology: “Innovation in Education and Inclusion”, 18-20 July 2018, Lima, Perú.

de las instrucciones de programación de computadoras existentes se enfocan principalmente en la capacitación de la sintaxis del lenguaje de programación y las habilidades de programación, mientras que los conceptos de solución de problemas a menudo se ignoran [8].

En este artículo se describe la experiencia de la estrategia pedagógica de utilización de aprendizaje basado en juegos y herramientas lúdicas como Scratch [9], Lightbot [10] y App Inventor [11] para la enseñanza de introducción a la programación en el curso de Informática Básica de la Escuela Profesional de Medicina de la Universidad Nacional de San Agustín [12].

II. MARCO CONCEPTUAL

A. Alfabetización Digital

La alfabetización digital ofrece la oportunidad de adquirir competencias que permiten conocer y comprender los procesos de comunicación a través de dispositivos digitales. En la última década, la web 2.0, los dispositivos móviles, la poderosa industria del videojuego y el resurgir del movimiento DIY (Do It Yourself) han puesto en un primer plano las necesidades y carencias en la formación sobre pensamiento computacional y programación. Como respuesta a esta demanda tanto gobiernos, empresas y organizaciones sin ánimo de lucro o instituciones educativas, han desarrollado proyectos y adoptado decisiones relacionadas con el fomento de la “codificación” en la educación [13]. La codificación es parte del razonamiento lógico y representa una de las habilidades clave que forma parte de lo que ahora se llaman “habilidades del siglo 21” [14].

En una sociedad 2.0, como la actual, la alfabetización digital no sólo implicaría la habilidad de consumo sino también de producción del contenido [15], razón por la cual es importante fomentar el aprendizaje de la programación en los jóvenes estudiantes.

B. Pensamiento Computacional

El pensamiento computacional es una metodología basada en la implementación de los conceptos básicos de las ciencias de la computación para resolver problemas cotidianos, diseñar sistemas domésticos y realizar tareas rutinarias. Esta nueva forma de abordar los problemas nos permite resolver con eficacia y éxito problemas que de otra forma no son tratables por una persona [16]. Al convertir el pensamiento en objeto mediante procedimientos, algoritmos y estructura de datos, el conocimiento personal se hace público y puede ser compartido con otros. Así el pensamiento computacional pasa a ser participativo [13].

El pensamiento computacional se relaciona con la codificación para la implementación de algoritmos que permiten la resolución de problemas.

C. Aprendizaje Basado en Juegos

El aprendizaje basado en juegos provee un método pedagógico que apoya el aprendizaje activo y experimental, y tiene el potencial de motivar y hacer partícipes a los estudiantes [17].

Según [18], las principales ventajas del aprendizaje basado en juegos son:

- 1) *Motiva al alumno*
- 2) *Ayuda a razonar y ser autónomo*
- 3) *Permite el aprendizaje activo*
- 4) *Da al alumno el control de su aprendizaje*
- 5) *Proporciona información útil al profesor*
- 6) *Potencia la creatividad y la imaginación*
- 7) *Fomenta las habilidades sociales*
- 8) *Contribuye a la alfabetización digital*

Al realizar actividades basadas en juegos se pueden impulsar cambios de hábito tanto en los estudiantes como en los profesores, especialmente, este cambio puede hacer que el estudiante pase a tener un papel más activo en el proceso de enseñanza-aprendizaje y no un simple receptor de conocimiento [19] [20].

La utilización de juegos como estrategia de enseñanza motiva el aprendizaje de los estudiantes.

D. Scratch

Scratch es un entorno de programación visual que permite a los usuarios crear proyectos multimedia interactivos [5]. Un proyecto Scratch será típicamente un juego o una animación sofisticada con dibujos, iconos, sonido y otros gráficos en movimiento [21].

Los proyectos realizados por los estudiantes que utilizan Scratch muestran que el cerebro humano es capaz de comprender, diseñar y construir proyectos con muchos bloques y de muchos tipos. Este proceso de aprendizaje a través de Scratch favorece el desarrollo de competencias y habilidades para analizar y resolver problemas utilizando técnicas y estructuras que se utilizan en las ciencias de la computación [22][23][16].

Scratch brinda la posibilidad de conjugar aspectos visuales atractivos con una sintaxis minimalista, pudiendo hacer énfasis en aspectos conceptuales y operativos más que sintácticos [7]. Actualmente se observa un interés por introducir nociones de programación en contextos educativos a partir de diferentes herramientas, dentro de las cuales se destaca Scratch [24].

E. Lightbot

Lightbot es un juego tipo rompecabezas de programación. Esto significa que en esencia, es un juego de puzzle pero su mecánica de juego tiene una relación directa con los conceptos de programación sin un lenguaje escrito. La meta de Lightbot

es guiar a un robot para que ilumine todos los bloques azules en cada nivel [25].

El juego utiliza un escenario de ficción, donde los jugadores controlan un robot, cuya tarea es iluminar todos los cuadros azules en una zona determinada para caminar. Esto se hace a través de un conjunto de comandos que representan conceptos básicos de programación tales como ejecución secuencial, funciones, recursión y flujos condicionales. La etapa básica incluye conceptos básicos de diseño de programa: flujo secuencial de ejecución, depuración y prueba del programa, así como funciones y procedimiento [26].

La selección de dicha aplicación se debe a sus múltiples ventajas, se basa en bloques gráficos y es una interfaz muy sencilla e intuitiva que permite a los alumnos dar sus primeros pasos en programación [27].

La mecánica de juego que se utiliza en Lightbot es el escalado de niveles, en el cual los estudiantes a medida que avanzan de nivel se enfrentan a problemas de mayor complejidad y adquieren nuevos conocimientos y habilidades en programación.

F. App Inventor

App Inventor for Android (AIA) de Google es el lenguaje de programación visual basado en bloques más nuevo diseñado para introducir a los estudiantes a la programación a través de la creación de aplicaciones móviles (apps). AIA abre el mundo de las aplicaciones móviles a los programadores novatos. Existen historias de éxito sobre el uso de AIA para introducir a los estudiantes universitarios a la programación [28].

App Inventor es una herramienta de programación visual basada en bloques que permite crear aplicaciones móviles de manera fácil y divertida para sistema operativo Android [29].

AIA es una herramienta efectiva para introducir a los programadores principiantes a la programación, tiene una forma visual de crear programas y, por lo tanto, reduce las barreras de la entrada inicial a la programación. Pero más allá de eso, dado que AIA conecta a un programador con el mundo de las aplicaciones móviles, esta es la herramienta ideal para atraer los intereses de los jóvenes de hoy [28].

III. DESCRIPCIÓN DEL TRABAJO

Se realizará una descripción de la experiencia de utilización de herramientas lúdicas para la enseñanza de la introducción a la programación en el curso de Informática Básica impartido a los estudiantes de primer año de la carrera de Medicina en el 2017.

Las herramientas lúdicas se utilizaron con el fin de reducir la complejidad de aprendizaje de los conceptos de programación, la cual resulta difícil de comprender en los cursos introductorios de programación. Estas herramientas

permiten fortalecer la resolución de problemas y creatividad en los estudiantes de manera sencilla y divertida.

Las herramientas lúdicas utilizadas son Scratch, Lightbot y App Inventor.

A. Generalidades

El curso de Informática Básica en la Escuela Profesional de Medicina de la Universidad Nacional de San Agustín de Arequipa – Perú, es dictado en el segundo semestre y tiene una duración de 17 semanas. Actualmente el curso tiene 2 horas teóricas y 2 horas de laboratorio semanales.

B. Contenido

En la Tabla I se muestra el contenido del curso de Informática Básica, donde se observa que en la tercera unidad de aprendizaje se desarrollan conceptos de introducción a la programación.

TABLA I
CONTENIDO DEL CURSO DE INFORMÁTICA BÁSICA

<p>PRIMERA UNIDAD DE APRENDIZAJE: FUNDAMENTOS DE INFORMÁTICA</p> <ol style="list-style-type: none"> 1. Estructura del computador 2. Sistema Operativo 3. Explorador del Windows 4. Almacenamiento de datos 5. Redes e Internet
<p>SEGUNDA UNIDAD DE APRENDIZAJE: HERRAMIENTAS DE OFFICE Y GOOGLE APPS</p> <ol style="list-style-type: none"> 1. Procesador de texto 2. Hoja de cálculo 3. Software de presentación 4. Herramientas de Google Apps
<p>TERCERA UNIDAD DE APRENDIZAJE: INTRODUCCIÓN A LA PROGRAMACIÓN</p> <ol style="list-style-type: none"> 1. Introducción a la Programación 2. Instrucciones de selección 3. Instrucciones repetitivas 4. Métodos

C. Scratch

Scratch se utilizó para explicar los conceptos básicos de programación dibujando figuras geométricas, haciendo uso de coordenadas, variables, condicionales, bucles. También permitió explicar a los estudiantes la utilidad de las estructuras iterativas realizando el reemplazo de varios bloques de código que se repetían por bucles.

En la Fig. 1 se muestra un ejemplo básico de programación en Scratch.

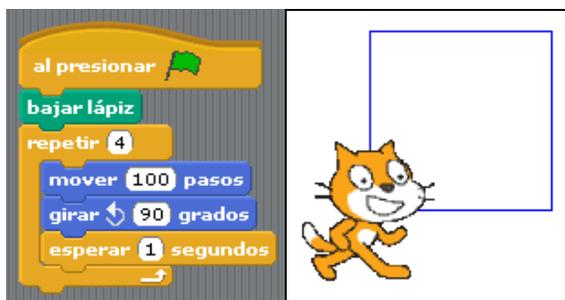


Fig. 1 Programa básico en Scratch.

La programación de Scratch consiste en unir bloques para definir el comportamiento de los objetos, además permite ejecutar instrucciones de forma paralela.

Los estudiantes de medicina desarrollaron programas, juegos e historietas con Scratch comprendiendo y utilizando instrucciones secuenciales, de selección y repetitivas.

D. Lightbot

Se utilizó el videojuego Lightbot para reforzar los conceptos de programación como algoritmos (secuencia de pasos finitos), instrucciones secuenciales e introducir el uso de funciones de manera divertida, logrando la atención y motivación en los estudiantes.

En la Fig. 2 se muestra las instrucciones utilizadas en Lightbot.

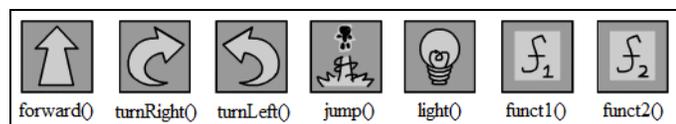


Fig. 2 Instrucciones de Lightbot

La meta del videojuego es programar al robot para que se mueva y encienda todos los bloques azules utilizando las instrucciones de Lightbot. En el nivel inicial (Fig. 3) se les explica el objetivo del juego y conjunto de instrucciones que pueden unir para crear una secuencia de pasos en la sección del método principal, la cual puede contener hasta doce instrucciones.

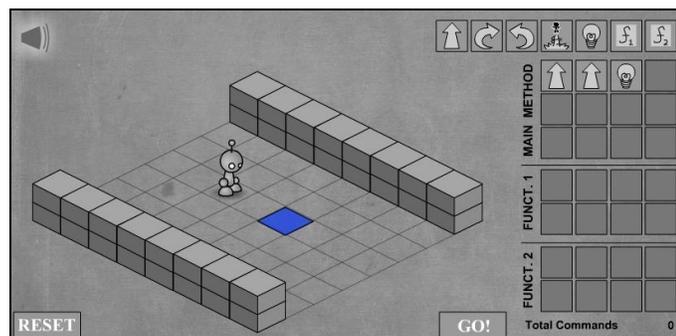


Fig. 3 Nivel inicial de Lightbot

Mientras se va avanzando de nivel se les explica las diferentes formas de resolver un problema, por ejemplo, en la Fig. 4 se les muestra una primera forma y se les hace notar que cuando un conjunto de instrucciones se utiliza varias veces, se puede crear funciones para agrupar estas instrucciones que se repiten y ser llamadas desde el método principal, la cual es la segunda forma de resolver el mismo problema (Fig. 5).

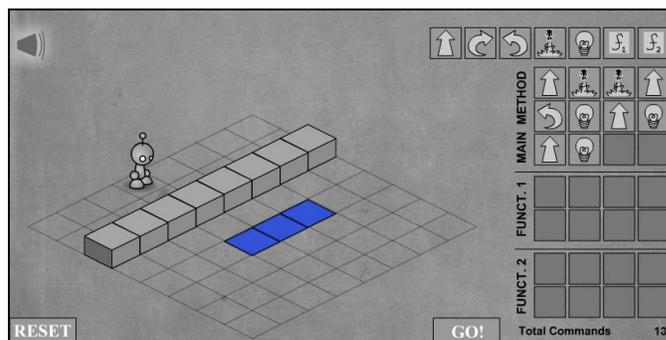


Fig. 4 Instrucciones que se repiten

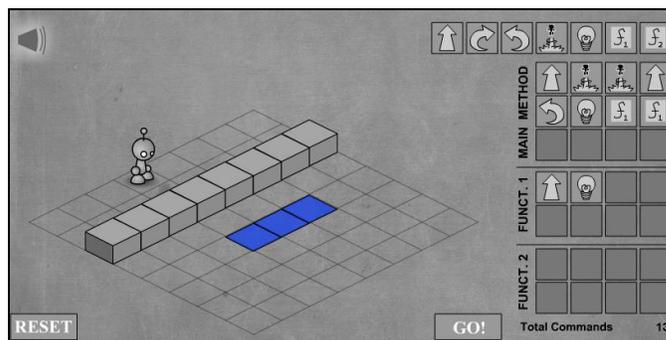


Fig. 5 Introducción del concepto de funciones

En los siguientes niveles notan que la cantidad de instrucciones que se requiere va incrementando y la complejidad del problema aumenta. En la Fig. 6 se observa que desde el método principal se llama a las funciones F1 y F2, las cuales se encargan de realizar una serie de instrucciones para encender los bloques azules respectivos, profundizando el concepto de modularización de programa, la cual consiste en dividir el programa en pequeños módulos independientes.

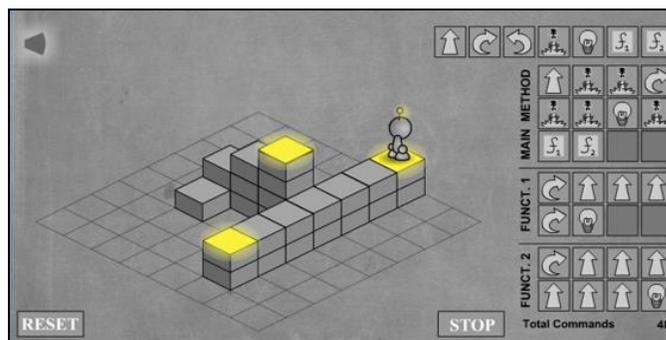


Fig. 6 Utilización de funciones

Los estudiantes pudieron entender de manera divertida y entretenida la utilidad de las funciones, reutilización de código, modularización de programas, así mismo comprendieron que hay varias soluciones para resolver un problema y que algunas son más eficientes que otras.

E. App Inventor

Luego de que los estudiantes de medicina adquirieran los conocimientos básicos de programación con Scratch y Lightbot, se eligió utilizar App Inventor debido a que los jóvenes interactúan todo el tiempo con sus dispositivos móviles, instalando y consumiendo juegos y aplicaciones. App Inventor les facilitaría la creación de sus propios juegos y aplicaciones que luego instalarían en sus dispositivos móviles, siendo más interesante para ellos y motivando su aprendizaje.

Con App Inventor los estudiantes de medicina aprendieron un nuevo entorno de programación visual basada en bloques poniendo en práctica el uso de variables, procedimientos (llamados también métodos o funciones), instrucciones secuenciales, condicionales e iterativas.

App Inventor ayudo a los estudiantes a elaborar aplicaciones creativas afines a su carrera profesional para sus dispositivos móviles, fortaleciendo la resolución de problemas.

Algunos de los trabajos realizados por los estudiantes fueron:

- Doctor Easy, el cual permite mostrar las causas, síntomas y recomendaciones de las enfermedades más comunes como son angina de pecho, migraña, heridas, ardor estomacal, estrés y gripe.
- Rompecabezas anatómicos, aplicación interactiva y entretenida para familiarizarse con el curso de anatomía, la cual permite a los estudiantes de medicina reconocer las piezas anatómicas y ordenarlas desde sus dispositivos móviles.
- Quiz de Anatomía, que sirve como instrumento de estudio para repasar los conocimientos adquiridos en el curso de anatomía mediante un pequeño examen de preguntas de opción múltiple.
- Medic Memory, juego que permite desarrollar la memoria visual.
- Aplicación IMC, la cual permite conocer el índice de masa corporal que se calcula en base al peso y estatura de la persona para prevenir la obesidad y enfermedades asociadas a ella, así como controlar periódicamente el peso.
- Test de Biología Molecular, permite probar conocimientos antes de una evaluación sobre temas de biología molecular.
- Hello Medicine, aplicación dirigida a estudiantes de medicina, es un juego interactivo que consiste en responder las preguntas, ya sean de elección múltiple

o para escribir, acumulando puntos. Permite reforzar conocimientos aprendidos.

- Aplicación MedWake, ayuda al estudiante de medicina a empezar su día lleno de alegría y con positivismo para que pueda sentirse capaz de lograr lo que se proponga, permite definir alarmas para organizar su tiempo y escuchar música para dormir.
- M-Calculator, calculadora médica para ayudar a los profesionales de la salud y a estudiantes de medicina con los cálculos más comunes para conocer valores como: electrolitos, peso corporal y masa, osmolaridad entre otros.

En la Fig. 7 se muestra algunas imágenes del diseño de la aplicación móvil M-Calculator.



Fig. 7 Aplicación M-Calculator

IV. RESULTADOS

Se elaboró un cuestionario de cuatro preguntas cerradas y una abierta para conocer la apreciación de los estudiantes, que se muestra en la Tabla II, la cual fue aplicada a 91 estudiantes de primer año de medicina que llevaron el curso de informática básica en el 2017.

TABLA II
ENCUESTA APRECIACIÓN DE ESTUDIANTES

<p>Pregunta 1: ¿Scratch le permitió aprender los conceptos básicos de programación (variables, condicionales, bucles)?</p> <ol style="list-style-type: none"> 1. Nada 2. Poco 3. Bastante 4. Completamente
<p>Pregunta 2: ¿La programación basada en bloques de Scratch facilita la elaboración de programas?</p> <ol style="list-style-type: none"> 5. Nada 6. Poco 7. Bastante 8. Completamente

Pregunta 3: ¿Le resulta atractivo las aplicaciones que se pueden desarrollar en App Inventor? 5. Nada 6. Poco 7. Bastante 8. Completamente
Pregunta 4: ¿Desarrollar aplicaciones en App Inventor promueve la creatividad y pensamiento algorítmico? 1. Nada 2. Poco 3. Bastante 4. Completamente
Pregunta 5: ¿Qué es lo que más le gusta de la herramienta App Inventor?

A continuación, se muestra los resultados de la encuesta desarrollada por 91 estudiantes.

En la Fig. 8, se observa que el 77% de los estudiantes encuestados opina que Scratch le permitió aprender los conceptos básicos de programación como variables, condicionales, bucles. Completamente (8%) y bastante (69%).

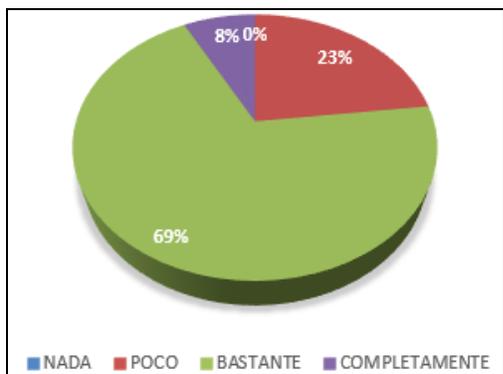


Fig. 8 Resultado de encuesta – Pregunta 1.

En la Fig. 9, se observa que el 88% de los estudiantes de medicina encuestados opina que la programación basada en bloques de Scratch facilita la elaboración de programas: Completamente (10%) y bastante (78%).

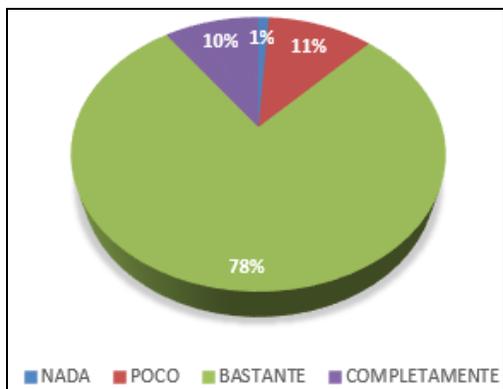


Fig. 9 Resultado de encuesta – Pregunta 2.

En la Fig. 10, se observa que al 82% de los estudiantes de medicina encuestados le resulta atractivo las aplicaciones que se pueden desarrollar en App Inventor: Completamente (21%) y bastante (61%).

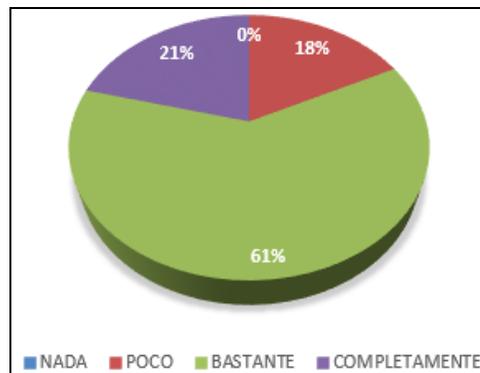


Fig. 10 Resultado de encuesta – Pregunta 3.

En la Fig. 11, se observa que el 93% de los estudiantes de medicina encuestados opina que desarrollar aplicaciones en App Inventor promueve la creatividad y pensamiento algorítmico: Completamente (21%) y bastante (61%).

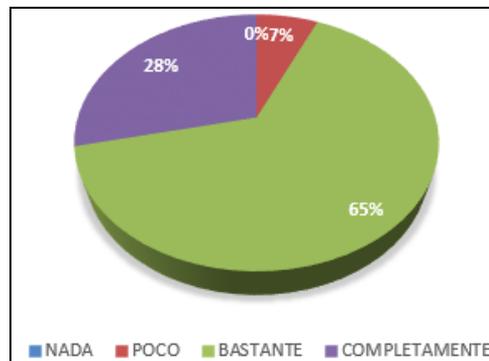


Fig. 11 Resultado de encuesta – Pregunta 4.

En la quinta pregunta, las respuestas más comunes y representativas del total de estudiantes encuestados fueron: “El poder programar visualmente de manera fácil sin complicación”, “Facilidad de aprender a programar mediante bloques”, “Fácil de utilizar y muy divertido”, “La mayor complejidad a comparación de Scratch”, “Poseer una aplicación creada por mí en mi celular”, “Poder crear una aplicación según tu imaginación y creatividad”, “Crear aplicaciones creativas que te ayuden a divertirte o en los estudios”, “Crear mi propio juego”, “Desarrollar aplicaciones que nos ayuden como complemento a nuestros estudios”, “La diversidad de aplicaciones que se pueden realizar”, “Me gusta porque puedo personalizar mis ideas y crear aplicaciones que me puedan ser útiles y perfeccionarlas”, “Aprender a crear aplicaciones ya que antes del curso no tenía conocimiento de cómo elaborar uno”.

V. CONCLUSIONES

La introducción a la programación impartida en el curso de informática básica de la Escuela Profesional de Medicina de la Universidad Nacional de San Agustín resultó fácil y divertido para los estudiantes, los cuales pudieron entender los conceptos básicos de programación y desarrollar de forma creativa sus propios juegos y aplicaciones móviles afines a su carrera profesional, fortaleciendo su pensamiento computacional y la resolución de problemas.

El aprendizaje de conceptos introductorios a la programación en estudiantes principiantes de primer año de las instituciones de educación superior puede desarrollarse de manera sencilla, comprensiva y entretenida utilizando herramientas lúdicas.

El uso de herramientas de programación visuales basadas en bloques como Scratch y App Inventor permite reducir la complejidad de comprensión de sintaxis y facilita el aprendizaje del uso de variables, instrucciones secuenciales, condicionales y repetitivas.

El videojuego Lightbot permitió captar la atención de los estudiantes y explicar de manera fácil los conceptos de programación que son difíciles de entender como algoritmos, funciones, reutilización de código y modularización de programas.

App Inventor permite desarrollar aplicaciones móviles creativas, captando el interés y motivación de los jóvenes estudiantes por aprender a programar sus propios juegos y aplicaciones para sus teléfonos celulares, favoreciendo su aprendizaje activo y constructivo.

Cuando se aprende a programar, aprende estrategias importantes para resolver problemas, diseñar proyectos y comunicar ideas, las cuales son habilidades importantes en la sociedad actual [30].

REFERENCIAS

- [1] M. Prensky, "Digital Natives, Digital Immigrants Part 1", *On the Horizon*, vol. 9, no. 5, pp. 1-6, 2001.
- [2] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman and Y. Kafai, "Scratch: Programming for All", *Communications of the ACM*, vol. 52, no. 11, pp. 60-67, 2009.
- [3] J. Díaz, C. Banchoff, C. Queiruga and E. Martín, "Experiencias de la Facultad de Informática en la Enseñanza de Programación en Escuelas con Software Libre", in *Congreso Iberoamericano de Ciencia, Tecnología, Innovación y Educación*, 2014.
- [4] P. Lozano, B. Guerrero and W. Gordillo, "Scratch y Makey Makey: herramientas para fomentar habilidades del pensamiento de orden superior", *Redes de Ingeniería*, vol. 7, no. 1, pp. 16-23, 2016.
- [5] C. López-Escribano and R. Sánchez-Montoya, "Scratch y Necesidades Educativas Especiales: Programación para todos", *RED. Revista de Educación a Distancia*, no. 34, pp. 1-14, 2012.
- [6] K. Brennan, "Learning computing through creating and connecting", *Computer*, vol. 46, no. 9, pp. 52- 59, 2013.
- [7] A. Miños, "Primer curso de programación en enseñanza media: la experiencia con scratch", *Didasc@lia: Didáctica y Educación*, vol. 5, no. 1, 2014.
- [8] H.Y. Wang, I. Huang and G.J. Hwang, "Effects of an Integrated Scratch and Project-based Learning Approach on the Learning Achievements of Gifted Students in Computer Courses", in *2014 IIAI 3rd International Conference on Advanced Applied Informatics*, IEEE, pp. 382-387, 2014
- [9] Scratch. <https://scratch.mit.edu/>
- [10] Lightbot. <http://lightbot.com/>
- [11] App Inventor. <http://appinventor.mit.edu/explore/>
- [12] Universidad Nacional de San Agustín. <http://www.unsa.edu.pe/>
- [13] J. Valverde, M. R. Fernández and M. C. Garrido, "El pensamiento computacional y las nuevas ecologías del aprendizaje", *RED. Revista de Educación a Distancia*, no. 46, 2015.
- [14] M. Zapata-Ros, "Pensamiento computacional: Una nueva alfabetización digital", *RED. Revista de Educación a Distancia*, no. 46, 2015.
- [15] D. Sánchez, ¿Cuál es la importancia de la 'alfabetización digital' en el siglo XXI?, <http://www.vertice.org/blog/la-importancia-la-alfabetizacion-digital-siglo-xxi/>, Revisado en Febrero del 2017.
- [16] X. Basogain, M. A. Olabe and J. C. Olabe, "Pensamiento Computacional a través de la Programación: Paradigma de Aprendizaje", *RED. Revista de Educación a Distancia*, no. 46, 2015.
- [17] N. Charlier, M. Ott, B. Remmele and N. Whitton, "Not just for children: game-based learning for older adults", In *6th European Conference on Games Based Learning*, Cork, Ireland, pp. 102-108, 2012.
- [18] aulaPlaneta innovamos para una educación mejor, "Ventajas del aprendizaje basado en juegos o Game-Based Learning (GBL) [Infografía]", <http://www.aulaplaneta.com/2015/07/21/recursos-tic/ventajas-del-aprendizaje-basado-en-juegos-o-game-based-learning-gbl/>, Revisado en Febrero del 2018.
- [19] J. C. Cortizo, F. Carrero, B. Monsalve, A. Velasco, L. I. Díaz and J. Pérez, "Gamificación y Docencia: Lo que la Universidad tiene que aprender de los Videojuegos", in *VIII Jornadas Internacionales de Innovación*, Universidad Europea de Madrid, 2011.
- [20] E. Serna, M. Mauricio, T. San Miguel and J. Megías, "Experiencia de gamificación en Docencia Universitaria: aprendizaje activo y entretenido", in *In-Red 2016. II Congreso nacional de innovación educativa y docencia en red*, Editorial Universitat Politècnica de València, 2016.
- [21] J. Delgado, J. Güell, J. García, M. Conde and V. Casado, "Aprendizaje de la programación en el Citilab", *Revista CTS*, vol. 8, no. 23, pp. 123-133, 2013.
- [22] X. Basogain, M. A. Olabe, J. C. Olabe, I. Maiz and C. Castaño, "Mathematics Education through Programming Languages", in *21st Annual World Congress on Learning Disabilities*, pp. 553-559, 2012.
- [23] J. C. Olabe, M. A. Olabe, X. Basogain and C. Castaño, "Programming and robotics with Scratch in primary education", *Education in a Technological World: Communicating current and Emerging Research and Technological Efforts*, pp. 356-363, 2011.
- [24] N. Monjelat and P. S. San Martín, "Programar con Scratch en contextos educativos: ¿Asimilar directrices o co-construir Tecnologías para la Inclusión Social?", *Praxis educativa*, vol. 20, no. 1, pp. 61-71, 2016.
- [25] D. Yaroslavski, Lightbot: How does Lightbot teach programming?, http://lightbot.com/Lightbot_HowDoesLightbotTeachProgramming.pdf, Revisado en Febrero del 2017.
- [26] A. Mathrani, S. Christian and A. Ponder-sutton, "PlayIT: Game Based Learning Approach for Teaching Programming Concepts", *Educational Technology & Society*, vol. 19, no. 2, pp. 5-17, 2016.
- [27] J. González, V. Paparoni and L. Vallejos, "Encendiendo las luces del conocimiento con lightbot 1.0.: la construcción del conocimiento en la clase de computación", In *XII Congreso de Tecnología en Educación y Educación en Tecnología*, 2017.
- [28] K. Roy, "App inventor for android: report from a summer camp", In *Proc. of the 43rd ACM technical symposium on Computer Science Education*, pp. 283-288, New York, USA, 2012.
- [29] Página oficial de appinventor.org, <http://www.appinventor.org/>, Revisado en Enero del 2018.
- [30] M. Resnick, "Learn to code, code to learn", *EdSurge*, May, 2013.