Asociación Colombiana de Facultades de Ingeniería

Revista Educación en Ingeniería
http://www.educacioneningenieria.org

Videojuego para el aprendizaje de lógica de programación

Leonardo Bermón Angarita, Amparo Prieto Taborda, Juan Diego Escobar Márquez & Juan David Vergara Díaz

Departamento de Informática y Computación, Universidad Nacional de Colombia, Manizales, Colombia.
lbermona@unal.edu.co, maprietot@unal.edu.co, judescobarmar@unal.edu.co, jdvergard@unal.edu.co

Resumen— Los cursos de programación de computadores abordan algunos conceptos y elementos que son difíciles de aprender por los estudiantes. Una estrategia para facilitar el aprendizaje de dichos conceptos y elementos es la utilización de videojuegos que ayuden a presentar y reforzar los conceptos de lógica y estructuras computacionales. Este artículo presenta el desarrollo de un videojuego para dispositivos móviles orientado al aprendizaje de la lógica de programación en estudiantes de básica secundaria. La validación realizada con un grupo de estudiantes mostró un amplio interés por la programación utilizando videojuegos considerándolo un recurso didáctico, divertido y con una dificultad alta. El videojuego desarrollado facilita el aprendizaje de conceptos de programación estructurada.

Palabras Clave— videojuegos, lógica computacional, programación estructurada, gamificación.

Recibido: 17 de noviembre de 2020. Revisado: 6 de marzo de 2021. Aceptado: 10 de marzo de 2021.

Game for learning programming logic

Abstract— Computer programming courses address some concepts and elements that are difficult for students to learn. One strategy to facilitate the learning of such concepts and elements is to use videogames that help present and reinforce the concepts of logic and computer structures. This paper presents the development of a videogame for mobile devices oriented to the learning of programming logic in basic secondary students. The validation carried out with a group of students showed a wide interest in programming using videogames considering the video game didactic, fun and with a high difficulty. The videogame developed facilitates the learning of structured programming concepts.

Keywords— videogames, computational logic, structured programming, gamification.

1 Introducción

Muchas iniciativas alrededor del mundo se han creado con el fin de promover el aprendizaje de la programación de computadores en cualquier etapa de la vida [1]- [2]. Sin embargo, se hace mayor énfasis en fomentar este tipo de aprendizaje en personas más jóvenes [3].

La programación de computadores se caracteriza por ser un proceso inherentemente analítico. El aspecto más importante de la programación es desarrollar un algoritmo; en otras palabras, una solución precisa y exacta a un problema que se puede ejecutar en un período de tiempo específico siguiendo pasos bien definidos. Sin embargo, lograr un algoritmo correcto implica que los desarrolladores poseen un pensamiento lógico y estructurado. Por lo tanto, programar se convierte en una tarea complicada ya que involucra procesos mentales complejos que se evidencian en el promedio obtenido por los estudiantes que cursan asignaturas relacionadas con el tema, el cual llegan a sobrepasar el 26% de repitencia [4].

Además, el análisis y diseño de una solución algorítmica en el contexto de las prácticas de programación implica identificar los objetivos de implementación, comprender los recursos disponibles para construir una solución, dividir un problema en partes más pequeñas, resolverlas y sintetizar una solución basada en los componentes más pequeños [5].

Las habilidades de programación son un punto débil ya que los estudiantes de carreras, en las cuales la programación es fundamental, llegan sin ninguna experiencia previa, lo cual dificulta su proceso de aprendizaje [6]- [7]. Por lo tanto, se evidencia la importancia de enseñar las bases, la lógica computacional desde el colegio, de igual forma que se hace con las matemáticas, la física, la química, etc.

Desde una perspectiva educativa, los videojuegos han demostrado ser una herramienta complementaria en la construcción del conocimiento sistemático de los estudiantes [8]. La sistematización a través de juegos digitales puede permitir un mejor acompañamiento de los estudiantes, verificando errores frecuentes y presentándoles recursos multimedia de una manera más atractiva en comparación con las aulas tradicionales.

La propuesta que se plantea en este artículo es el desarrollo de un videojuego para dispositivos móviles orientado al aprendizaje de la lógica computacional en estudiantes de básica secundaria.

El artículo está organizado como sigue. El primer apartado presenta la introducción. El segundo apartado describe los elementos teóricos sobre aprendizaje de lógica de programación y videojuegos aplicados al aprendizaje de lógica de programación. El tercer apartado muestra los trabajos relacionados. El cuarto apartado describe la metodología utilizada en el estudio. El quinto apartado realiza una descripción del videojuego desarrollado. El sexto apartado presenta los aspectos referentes a la validación del videojuego. El séptimo apartado realiza un análisis y discusión acerca del

videojuego desarrollado. Finalmente, se presentan las conclusiones y trabajo futuro en el último apartado.

2 Marco teórico

El marco teórico del estudio se describirá a partir de una descripción acerca del aprendizaje de la lógica de programación y sobre videojuegos aplicados en el aprendizaje de dicha lógica.

2.1 Aprendizaje de lógica de programación

Es probable que las habilidades de razonamiento algorítmico y lógico sean las más importantes y necesarias para aprender a programar [9]. Estas habilidades determinan las capacidades de los estudiantes para brindar soluciones a través de estrategias y técnicas de resolución de problemas [10].

Algunas de las habilidades útiles que la programación fomenta son el pensamiento analítico y lógico, la creatividad, la capacidad de resolución de problemas y el trabajo en equipo entre otras [5].

La programación permite a los alumnos encarar procesos de autocorrección y búsqueda de errores (depurar un programa que no funciona adecuadamente), los enfrenta a retos de resolución de problemas complejos (introduciendo al alumno en la algoritmia) o les presenta conceptos que pueden llegar a ser complejos para un alumno de primer curso de ingeniería como, por ejemplo, la recursividad [11].

Debido a esto, países como Estonia, Francia e Inglaterra han añadido a sus currículos escolares la enseñanza de la programación desde los primeros años de formación [12].

Antes de entrar a manejar un lenguaje de programación robusto, con su sintaxis y métodos propios, se debe iniciar con la base, la lógica computacional. Esta lógica se considera el pilar fundamental porque si se logra comprender, se puede aplicar a cualquier lenguaje de programación ya que estos son herramientas que permiten exteriorizar dicha lógica por medio de palabras propias de cada lenguaje [5]. Las herramientas que enseñan a programar comienzan con una aproximación en primera instancia a la lógica de programación por medio de ejercicios en los que se explican los principios básicos de la programación. Es común que esta primera aproximación sea desde la enseñanza de la programación estructurada que es uno de los paradigmas de programación [13].

En el aprendizaje de la programación no sólo basta con memorizar los conceptos y la sintaxis propia de cada lenguaje, sino que se deben desarrollar capacidades de análisis y lógica para que, junto con la inteligencia y la disciplina, el aprendiz sea capaz de manipular datos por medio de algoritmos y estructuras de datos entendiendo el proceso [5], [14].

Según [15], el conocimiento de programación incluye los siguientes pasos: (1) una buena comprensión del problema o conjunto de requisitos, (2) identificación continua de las construcciones de programación a partir de la especificación, (3) diseño de un algoritmo, (4) estrategias de resolución de problemas (*top-down*, *bottom-up*, comunicación, lluvia de ideas, aprendizaje entre pares), (5) pensamiento algorítmico, (6) estrategias para el diseño, (7) conversión de pseudocódigo a lenguaje de programación, y (8) prueba y corrección de código hasta que el programa esté libre de errores.

2.2 Videojuegos en el aprendizaje de lógica de programación

En [16] se especifican algunas de las razones por las que la programación es difícil de enseñar/aprender: (1) exige un alto nivel de abstracción, (2) necesita un buen nivel de conocimientos y técnicas prácticas de resolución de problemas, (3) requiere un estudio muy práctico e intensivo, (4) generalmente la enseñanza no se puede individualizar, debido al tamaño de las clases, (5) es mayoritariamente dinámica, pero generalmente se enseña utilizando materiales estáticos, (6) las metodologías de los profesores muchas veces no toman en consideración los estilos de aprendizaje del estudiante, (7) los lenguajes de programación tienen una sintaxis muy compleja con características definidas para su uso profesional y no con motivaciones pedagógicas.

El uso de videojuegos como herramienta de aprendizaje ofrece ventajas como las indicadas en [17] y [18]. Así: Ejercitar la imaginación, facilitar la interacción con otras personas, ayudar a mejorar la atención y el autocontrol, y después de un tiempo, el dominio de habilidades se hace notable, dando así una sensación de control y confianza. Además, por medio de estas herramientas se ven favorecidas una serie de habilidades como [19]: reflexión (al examinar el juego y ver sus mecánicas, los jugadores pueden sacar sus propias conclusiones), dinamización de la conducta y el pensamiento (se refleja en una mayor agilidad mental), capacidad de deducción, capacidad para resolver problemas (presentan una serie de actividades a realizar para continuar avanzando), memorización (se deben retener en la mente patrones, conceptos, objetos, mecánicas) y aumento de la autoestima y confianza debido al impacto emocional.

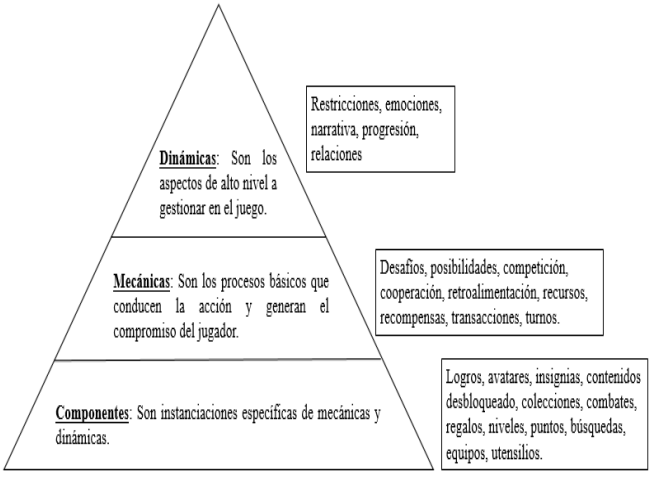

Figura 1. Elementos de un videojuego.
Fuente: Adaptado de [20].

En [20] se propone una pirámide que organiza los elementos del juego en tres categorías: dinámicas, mecánicas y componentes. Los componentes forman la base de la pirámide, las mecánicas se encuentran en el medio y las dinámicas en la parte superior (Fig. 1). Las dinámicas contienen los aspectos principales y elementos conceptuales de un juego serio. Ejemplos de elementos de este grupo son: restricciones, emociones, narrativa, progresión y relaciones. Las mecánicas

contienen el proceso básico que dirige a los usuarios a interactuar con el contenido y continuar impulsando la acción. Algunos ejemplos de mecánicas son: desafíos, retroalimentación, competencia y cooperación. Los componentes son menos abstractos que las dos primeras categorías y son herramientas que pueden emplearse para motivar al usuario en el entorno de interés. Algunos ejemplos son logros, avatares, insignias, combates, tablas de clasificación y niveles.

3 Trabajos relacionados

CoderDojo [21], La Hora del Código [22], Scratch Day [23] e incluso la iniciativa Yo puedo programar que es impulsada por el MinTIC colombiano, Microsoft y Colombia Joven, son algunas de las iniciativas más conocidas que tienen como objetivo enseñar a los jóvenes a programar. No solo para fomentar el interés en la adquisición de habilidades técnicas en el manejo de lenguajes de programación sino en desarrollar habilidades útiles que pueden ser aplicadas en la vida académica-profesional y/o personal.

La Hora del Código [22] es una iniciativa patrocinada por grandes marcas y usa juegos para ayudar a las personas a aprender a programar. Los juegos incluidos están desarrollados en Phyton, C/C++, Javascript o HTML/CSS, entre otros. Los métodos de los juegos se centran en enseñar el flujo normal de un algoritmo, variables, ciclos y demás conceptos relacionados, con la ayuda de una interfaz amigable.

CoderDojo [21] es una iniciativa que utiliza como estructura el concepto de dojo ("lugar de meditación y práctica del budismo zen y de las artes marciales tradicionales del Japón") para practicar y aprender conceptos de programación que guíen a los niños. Scratch [23] es una propuesta gratuita desarrollada en el Instituto Tecnológico de Massachusetts (MIT) que consiste en programar historias interactivas, juegos y animaciones, y compartir las creaciones con otros en una comunidad online. Es usado como base para la enseñanza de programación como herramienta para el desarrollo de juegos educativos. MIT App Inventor [24] es una iniciativa innovadora para enseñar programación por medio de la transformación del lenguaje complejo de codificación en simples bloques que se pueden arrastrar y soltar, lo que genera una construcción visual.

Alice [25] es utilizado por los profesores en las aulas de la escuela en temas que van desde las artes visuales y artes del lenguaje a los fundamentos de la programación e introducción a los cursos de Java.

Kodable [26] es una plataforma web que presenta la propuesta de enseñar a programar con solo 20 minutos a la semana. Las dinámicas son similares a las anteriores propuestas. Para los docentes permite la creación de clases e invitar a sus estudiantes.

En la Tabla 1 se presenta una comparativa de los trabajos anteriormente reseñados. En cuanto a la cobertura, la Hora del Código presenta plataformas web donde se puede acceder a los juegos, y muchos de ellos pueden ser descargados en Android e iOS, a diferencia de Scratch, Kodable y App Inventor. CoderDojo tiene limitantes de cobertura pues su dinámica de enseñanza en ciertos lugares limita el alcance de su ejercicio.

Tabla 1.
Trabajos relacionados

Trabajos relacionados	Cobertura	Dificultad	Efectividad
Hora de código	5	4	3
CoderDojo	3	4	4
Scratch	4	4	3
Kodable	4	4	4
Alice	5	4	3
MIT App Inventor	4	3	3

Fuente: Autores.

La dificultad está relacionada con qué tan difíciles son los juegos. Los trabajos relacionados comparten una dificultad similar ya que al ser productos validados pueden mejorar con el tiempo aspectos como este. Las aplicaciones realizadas con App Inventor están limitadas por su simplicidad.

En cuanto a qué tan efectiva es la iniciativa a la hora de enseñar a los niños, CoderDojo y Kodable se llevan las mejores puntuaciones, pues incluyen un tutor que facilita la enseñanza. Mientras que las demás iniciativas tienen un enfoque más de aprender jugando por su propia disposición.

4 Metodología

La metodología utilizada para la realización del videojuego consta de las siguientes fases: diseño del videojuego, desarrollo del videojuego y validación.

4.1 Diseño del videojuego

El diseño del videojuego se abordó desde dos perspectivas. La primera fue el diseño del videojuego desde el contexto de la gamificación. Para ello, se determinaron elementos fundamentales del videojuego como las mecánicas, dinámicas y componentes. La segunda perspectiva fue un diseño basado en la ingeniería de software donde se realizó un diseño del videojuego a partir de la definición de sus requisitos funcionales de software utilizando diagramas de casos de uso, diagramas de clases y otros diagramas de modelado de software. El diseño de software también incluyó el diseño de la interfaz gráfica de usuario.

4.2 Desarrollo del videojuego

El desarrollo del videojuego se realizó utilizando la herramienta Construct, una herramienta propietaria para la creación de videojuegos en dos dimensiones con capacidad de ser exportados a plataformas móviles y de escritorio.

Debido a las restricciones de tiempo del proyecto de desarrollo, se utilizó un enfoque basado en prototipos donde fue necesario, una vez se desarrollaba un conjunto de requisitos funcionales, realizar las respectivas pruebas y una posterior retroalimentación para mejorar el juego de forma continua.

Las características mínimas del hardware requerido para el funcionamiento del videojuego son: procesador Dual-Core 1.0 Ghz, memoria RAM de 512 MB, resolución de pantalla de

480x800 pixeles. Con respecto al sistema operativo, el videojuego requiere de Android 4.1 (*Jelly Bean*).

En cuento al desarrollo específico de los componentes software del programa, se desarrolló un conjunto de hojas de eventos utilizando la herramienta Construct. Las hojas de eventos permitieron la definición de acciones como cargar partidas, definir jugadores y dificultad, habilitar niveles, realizar movimientos del jugador y de los rivales, entre otros.

4.3 Validación

Para la validación del videojuego desarrollado se seleccionó un grupo de 10 estudiantes de un colegio privado de la ciudad de Manizales. Entre los criterios de inclusión para pertenecer al grupo estuvieron:

- La edad del estudiante debe estar entre los 11 y 15 años.
- Los estudiantes debían tener conocimiento previo en videojuegos utilizando teléfonos móviles.
- Los estudiantes no debían tener experiencia previa en programación de computadores.
- Los estudiantes manifestaron su deseo de participar voluntariamente.

El instrumento de validación utilizado fue una encuesta para valorar cinco variables: facilidad de aprendizaje de conceptos de programación estructurada, dificultad del videojuego, diversión percibida por el jugador, capacidad para finalizar el videojuego y continuidad del juego. La encuesta estaba conformada por siete ítems (tres ítems en escala Likert, dos preguntas dicotómicas en formato Si/No, una pregunta de elección múltiple y una pregunta abierta).

La validez del constructo fue evaluada por medio del indicador denominado alfa de Cronbach cuyo valor obtenido de 0,7819, el cual al ser mayor a 0,7 y estar cercano a 0,8 refleja un buen nivel de fiabilidad del instrumento.

Para la recolección de datos, los estudiantes respondieron la encuesta implementada como un formulario en la herramienta *Google Forms*. Luego, los datos fueron procesados y analizados utilizando *Microsoft Excel*.

Para el desarrollo y validación del videojuego se tuvieron en cuenta algunos aspectos éticos, como el propósito, la complejidad y la orientación.

- El propósito del juego fue facilitar el aprendizaje de estructuras básicas de programación que pueda ser utilizado por estudiantes de educación secundaria, los cuales generalmente son menores de edad. Por lo tanto, los ejercicios planteados son apropiados para dicho público objetivo. Para ello, el contexto del videojuego es el fútbol, un deporte muy popular.
- La complejidad del videojuego no es muy alta y se puede realizar en pocos minutos. Por lo tanto, su uso no ocasiona trastornos de adicción.
- El juego no está orientado a una inmersión total. Así que los jugadores no perderán la noción del tiempo y no estarán aislados de lo que les rodea.

5 Descripción del videojuego

Con la incorporación de técnicas de aprendizaje basadas en videojuegos y aprovechando la disposición actual de los jóvenes hacia las nuevas tecnologías, se espera atraerlos y motivarlos en la adquisición de conocimiento y en la solución de problemas, enlazando actividades propias del entretenimiento con los procesos de aprendizaje.

El objetivo principal del videojuego es que el jugador aprenda los conceptos básicos expuestos en el teorema del programa estructurado [27]. Este teorema sostiene que cualquier función que sea ejecutada por un computador puede ser escrita en un lenguaje de programación utilizando combinaciones de únicamente 3 estructuras lógicas (secuencia, selección e iteración):

- Secuencia: ejecución de una instrucción tras otra.
- Selección: ejecución de una de dos instrucciones (o conjuntos), según el valor de una variable booleana.
- Iteración: ejecución de una instrucción (o conjunto) mientras una variable booleana sea verdadera. Esta estructura lógica también se denomina ciclo o bucle.

Además de estas tres estructuras básicas, se presentará al jugador el concepto de subrutina. Las estructuras lógicas propuestas en el teorema más el uso de subrutinas conforman lo que se conoce como el paradigma de la programación estructurada [13].

Se eligió el modelo por prototipos como modelo de desarrollo de software debido a que si bien se conocía el problema al que se intentaba dar solución, fue necesario realizar pruebas y posterior retroalimentación para mejorar el videojuego de forma continua.

Como herramienta de desarrollo se seleccionó Construct 2 [28] por su facilidad de manejo, gracias a su modelo *drag-and-drop* que permite crear videojuegos con estética 2D de manera rápida y sin la necesidad de tener conocimientos en un lenguaje de programación específico, ya que el desarrollo se realiza por medio de hojas de eventos que utilizan condiciones o *triggers* que ejecutan las acciones deseadas.

5.1 Dinámicas, mecánicas y componentes

A continuación, se describirán las dinámicas, mecánicas y componentes del videojuego. Las dinámicas definidas para el videojuego se las listadas en los párrafos siguientes.

Narrativa. El juego tiene un enfoque orientado al deporte, específicamente el futbol, donde se simula su dinámica y sus reglas. El objetivo central del futbol, y por ende del juego, es anotar un gol o llevar el balón hasta la portería del rival. Para ello, el jugador debe planificar una solución, la cual imita la realización de un programa de computador (una serie de instrucciones que le dicen al computador qué hacer y cómo hacerlo) [29].

Progresión. Cada vez que el jugador supera un nivel satisfactoriamente, obtiene puntos por este logro y pasa al siguiente nivel. Cada nivel tiene un rango de puntos que será otorgado según el número de intentos con el cual se superó el nivel, entre menos intentos más puntuación.

Reglas o restricciones: a) El jugador no puede salir del campo, si esto ocurre debe reiniciar el nivel, b) El jugador no

puede perder el balón con un rival, si esto ocurre debe reiniciar el nivel. El rival quita el balón al coincidir en la misma posición con el jugador, c) El jugador debe cumplir con el número de movimientos necesarios para llegar a la meta. Si los movimientos no son suficientes, el jugador debe reiniciar el nivel.

Las mecánicas específicas definidas para el videojuego fueron desafíos y retroalimentación.

Desafíos: El videojuego en los primeros niveles presentará retos simples como mover el futbolista de un punto a otro por medio de la barra de ejecución para enseñarle el funcionamiento básico. A medida que el juego avanza, los retos se irán haciendo más complicados.

Retroalimentación. Al finalizar cada nivel del videojuego se muestra una corta explicación de la relación entre el objetivo del nivel y los conceptos de la lógica de programación.

Se definió una colección de componentes para el videojuego:

- Equipos: Existen dos equipos en el juego. Cada equipo tiene cierto número de jugadores, dependiendo de la dificultad del nivel.
- Jugador: El jugador es quien lleva o va tras la pelota.
- Pelota: Es el objeto que el jugador debe conducir hasta el arco rival para anotar un gol.
- Cancha: Lugar donde se desarrolla el juego, en forma de rectángulo y con marcas que indican los límites del campo.
- Arco: Lugar a donde se debe conducir el balón.
- Avatar: Color del uniforme del equipo seleccionado a partir de un conjunto predeterminado.

En el primer nivel se aplicará la lógica secuencial por medio de movimientos consecutivos que el jugador deberá seleccionar para cumplir con el objetivo. Más adelante, se introduce el concepto de sub-rutinas, el cual tendrá unos niveles dedicados hasta lograr su entendimiento. En los niveles más avanzados, el concepto de ciclo será introducido, mientras se mantienen los conceptos anteriores y en los niveles finales, el jugador deberá aplicar todo lo aprendido para superar los retos que se plantean.

El jugador cuenta con un número limitado de intentos que se denominan créditos. Se pierde crédito por cada intento fallido; se gana crédito por cada nivel superado. Existen 3 dificultades que son:

- Fácil. Inicia el juego con un número de créditos de 7. Superar un nivel otorga 1 crédito, perderlo reduce 1 crédito. Se puede simular cuantas veces se considere el movimiento de los jugadores contrarios.
- Normal. Inicia el juego con un número de créditos de 5. Superar un nivel otorga 1 crédito, perderlo reduce 1 crédito. Se puede simular el movimiento de los jugadores contrarios.
- Alta. Inicia el juego con un número de créditos de 3. Superar un nivel otorga 1 crédito, perderlo reduce 1 crédito. No se puede simular el movimiento de los jugadores contrarios.

5.2 Requerimientos funcionales basados en casos de uso

En la Fig. 2 se presenta un diagrama de casos de uso UML de alto nivel que especifica los requerimientos funcionales definidos para el videojuego desarrollado.

Los casos de uso de alto nivel definidos para el videojuego fueron:

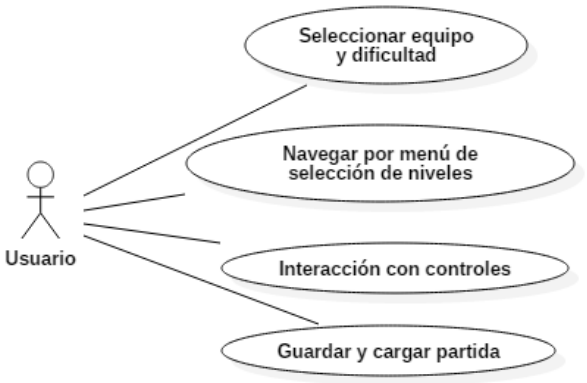

Figura 2. Casos de uso de alto nivel del videojuego.
Fuente: Los autores.

- Selección de equipo y dificultad: El usuario debe poder seleccionar un tipo de jugador (color del uniforme entre 3 diferentes personajes) y la dificultad que estará presente durante toda la partida o hasta que decida comenzar una nueva.
- Navegar por menú de selección de niveles: En este caso de uso, el usuario podrá desplazarse horizontalmente por el menú de selección de niveles. Un nivel estará disponible si el usuario ya superó el nivel inmediatamente anterior. El usuario debe poder entrar a los diferentes niveles del juego (los que se encuentren habilitados).
- Interacción con controles: Este caso de uso hace referencia a la interacción del usuario con los diferentes controles del juego. El usuario debe poder interactuar con los botones en pantalla (por ejemplo, botones de movimientos o menú desplegable). Los botones de movimiento (arriba, abajo, izquierda, derecha) al presionarlos, se irán agregando a la línea de ejecución correspondiente (principal, la de la subrutina o la de los ciclos). Para las subrutinas y los ciclos deben haberse tocado los botones que activan su funcionalidad previamente.
- Guardar y cargar partida: Este caso de uso hace referencia a la posibilidad que tiene el usuario para guardar la partida que está jugando (botón presente en la pantalla del menú de selección de niveles) y de cargar la partida al iniciar el juego, si la había guardado previamente.

5.3 Interfaz gráfica del videojuego

Al iniciar el videojuego surgen dos botones para seleccionar las opciones: Cargar la última partida guardada o iniciar/reiniciar la partida. Esta última opción lleva al menú de selección de equipo y dificultad (Fig. 3). En esta opción, se selecciona el equipo de fútbol (1) y el nivel de dificultad

correspondiente (2), el cual puede ser fácil, normal o difícil. Una vez se haya seleccionado el equipo y la dificultad, un botón identificado con un visto bueno (3) llevará al menú de selección de niveles.

Figura 3. Selección de equipo y dificultad.

En el menú de selección de niveles (Fig. 4) se muestra la cantidad de créditos disponibles (1). Se muestra la cantidad de simulaciones permitidas por nivel de los movimientos rivales (2). Estas simulaciones sólo están permitidas cuando la dificultad seleccionada sea fácil o media, y se activarán en los niveles en los que haya rivales. Además, muestra los niveles (3) a los cuales el usuario puede acceder (en color) y a los que aún no puede (en gris). También se presentan unos botones que permiten seleccionar el nivel de dificultad (4). Un botón (5) permite guardar la partida hasta ese punto para continuar más adelante.

Figura 4. Selección de niveles.

Al inicio de una nueva partida y de cada una de las fases del juego, se mostrará un tutorial, el cual sirve para familiarizar al usuario con los controles y mecánicas del juego (Fig. 5).

Figura 5. Tutorial del videojuego.

En la Fig. 6 se presenta un ejemplo de un nivel del videojuego. Se muestra el nivel actual en el que se encuentra el usuario (1), el personaje que será movido por el usuario (2), el objetivo (3) como un punto brillante, al cual debe llegar el personaje valiéndose de los movimientos proporcionados por el usuario. Muestra los créditos disponibles del usuario (4), los cuales se irán incrementando cada vez que se supere un nivel y se perderán cuando: no se llegue al objetivo, el personaje salga del campo de juego o un rival le quite el balón al personaje que es movido por el usuario (aplica para los niveles en los cuales haya rivales). También cuenta con botones hacia izquierda, derecha, arriba y abajo (5), que el usuario presionará para planificar una ruta que se ejecutará. Una barra de ejecución principal (6) muestra los movimientos que el personaje realizará cuando el usuario ejecute el nivel. Los botones de borrado y de ejecución (7) permiten eliminar un paso específico del flujo de pasos planificado y la ejecución de los movimientos, respectivamente.

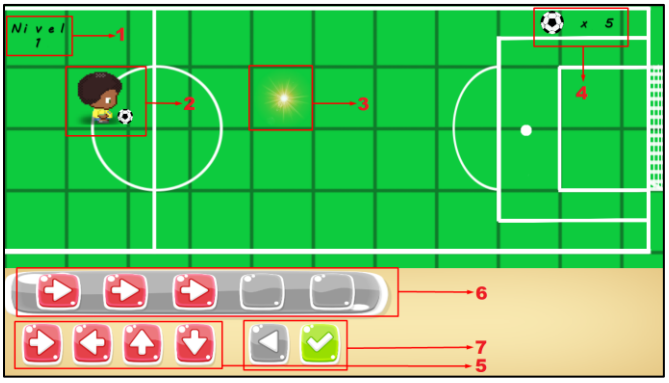

Figura 6. Ejemplo de nivel del videojuego.

Una vez el personaje alcanza al objetivo (marca brillante) o marca un gol, se despliega un mensaje (Fig. 7) que contiene los botones para: volver a jugar el nivel (1) y no se perderán ni ganarán créditos al volver a jugar un nivel ya superado; terminar el nivel y llevar al usuario al menú de selección de niveles y guardado de la partida (2); y continuar con el siguiente nivel (3).

Figura 7. Ejemplo de alcanzar un objetivo.

Al finalizar cada fase del juego se presenta (Fig. 8) una breve explicación que relaciona lo que se hizo en los diferentes niveles de la fase con la lógica de programación (1) y un botón que lleva al nivel siguiente.

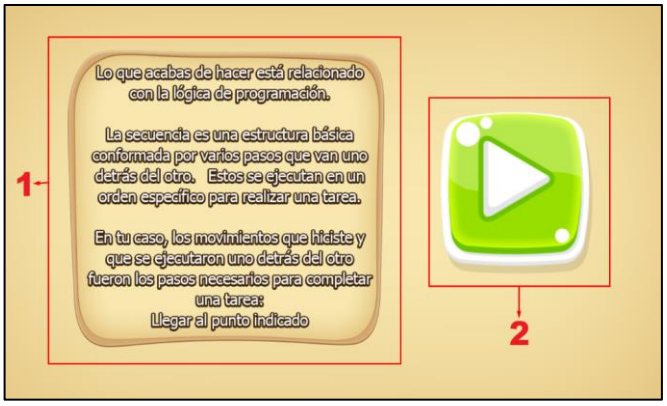

Figura 8. Explicación de la estructura de programación usada en el nivel.

Algunos niveles tienen rivales que el personaje controlado por el usuario debe evadir (Fig. 9). Un personaje rival debe ser evadido por el personaje controlado por el usuario. Si el usuario toca al personaje rival (1), se mostrarán los movimientos que éste hará durante el nivel (2). El botón (3) despliega un menú lateral.

Figura 9. Nivel del videojuego que incluye un rival.

En la Fig. 10 se presenta el menú lateral desplegado, el cual contiene los siguientes elementos:

- Botón que activa las subrutinas (1), al tocarlo (cuando se encuentre en color rojo), los botones que controlan los movimientos del jugador se agregarán a la barra de ejecución de subrutinas (2) cuando el usuario los presione. Esta barra de ejecución de subrutinas muestra los movimientos que el usuario ha introducido en la composición de la subrutina correspondiente.
- Botones de borrado y de ejecución (3). El botón de borrado elimina el último movimiento agregado por el usuario a la barra de ejecución de subrutinas. El botón de ejecución agrega la subrutina creada a la barra de ejecución principal del videojuego.
- El botón de simulación (4) se activará en los niveles en los que haya personajes rivales y la dificultad sea fácil o media. En la dificultad fácil se pueden ejecutar las simulaciones las veces que se considere necesario, mientras que, en dificultad normal sólo se podrá ejecutar 2 veces por nivel. Las simulaciones permiten mostrar en pantalla los movimientos previstos del rival, de tal manera que el usuario pueda planificar sus movimientos respectivos esquivando los del rival o rivales presentados.

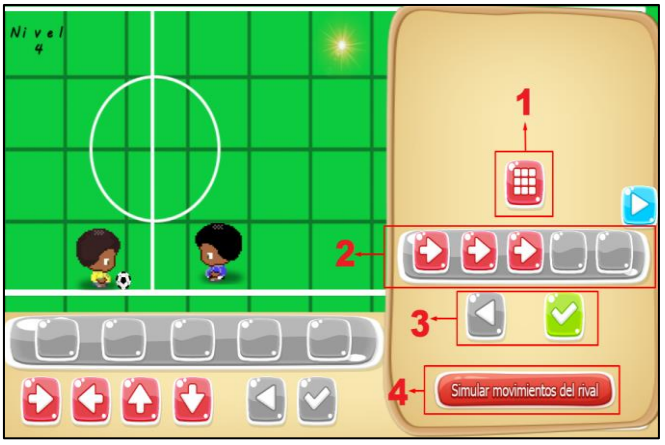

Figura 10. Menú lateral y subrutinas.

En la Fig. 11 se presenta el menú lateral desplegado correspondiente a ciclos. Presenta un botón que activa los ciclos (1), al tocarlo y estar en rojo, los botones que controlan los

movimientos se agregarán a la barra de ejecución de ciclos. Un botón aumenta en 1 el contador de ciclo (2). Un contador de ciclo (3) indica cuántas veces se repetirán los movimientos que el usuario agregó a la barra de ejecución de ciclos. La barra de ejecución de ciclos (4) muestra los movimientos que el usuario ha introducido al ciclo. Los botones de borrado y de ejecución (5) eliminan el último movimiento agregado por el usuario a la barra de ejecución de ciclos y agregan el ciclo a la barra de ejecución principal, respectivamente.

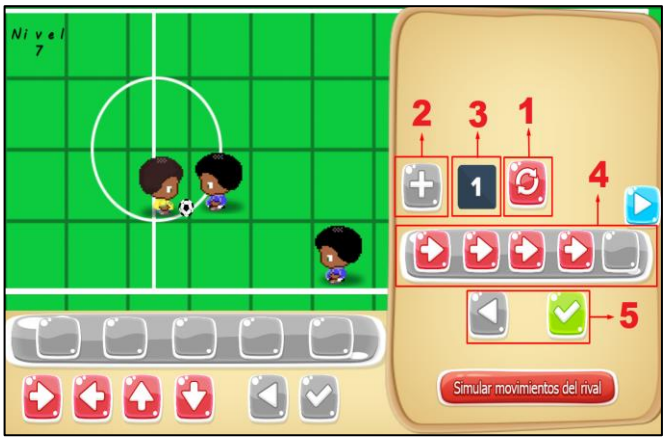

Figura 11. Menú lateral y ciclos.

En los niveles finales (Fig. 12) se introducen los conceptos aprendidos en los niveles anteriores (subrutinas y ciclos) y pueden surgir más rivales con sus respectivos movimientos que obstaculizan los movimientos del personaje.

Figura 12. Niveles finales.

6 Validación

La validación consistió una sesión de utilización del videojuego por parte de un grupo de 10 estudiantes. Inicialmente, se realizó una corta presentación que incluyó una explicación básica de los objetivos de la sesión y de las instrucciones correspondientes para interactuar con el videojuego. Luego, los estudiantes utilizaron el videojuego durante 15 minutos. Posteriormente, cada participante de la prueba diligenció una encuesta, la cual estuvo conformada por las variables y preguntas relacionadas en la Tabla 2.

Tabla 2
Encuesta de validación

Id	Variable	Pregunta	Escala de respuesta	Respuestas obtenidas
1	No aplica	¿En qué grado estás?	Quinto, Sexto, Séptimo, Octavo o Noveno	Quinto: 10%, Sexto: 0%, Séptimo: 80%, Octavo: 0%, Noveno: 10%
2	Aprendizaje	¿Aprendí conceptos de programación usando el videojuego?	1: Muy poco, 2: Insuficiente, 3: Neutral, 4: Suficiente, 5: Mucho	1: 0%, 2: 0%, 3: 10%, 4: 30%, 5: 60%
3	Dificultad	¿Qué dificultad tiene el videojuego?	1: Muy fácil, 2: Fácil, 3: Neutral, 4: Difícil, 5: Muy difícil.	1: 0%, 2: 0%, 3: 40%, 4: 40%, 5: 20%
4	Diversión	¿Qué tan divertido es el videojuego?	1: Nada divertido, 2: No es divertido, 3: Neutral, 4: Divertido, 5: Muy divertido	1: 0%, 2: 0%, 3: 40%, 4: 40%, 5: 20%
5	Capacidad de finalizar	¿Podrías llegar al final del juego?	Si o No	Si: 70%, No: 30%
6	Continuidad	¿Me gustaría volver a usar el videojuego?	Si o No	Si: 90%, No: 10%
7	No aplica	¿Qué piensas que falta en el juego?	Pregunta abierta	Pregunta abierta

Fuente: Autores.

El diseño general de la encuesta se basó teniendo en cuenta una versión ligera adaptada de los estudios [30]- [31] sobre usabilidad de los videojuegos.

Con respecto a las respuestas recibidas, en la Tabla 2 se presentan los resultados obtenidos en la última columna.

Con relación a la pregunta 1, ocho de los niños se encuentran en séptimo grado, mientras que los otros dos se encuentran en quinto y noveno.

Las respuestas de la pregunta 2 muestran que el 30% de los niños calificaron con 5 su aprendizaje y el 30% lo evaluaron con 4. Cabe destacar que el 90% de los niños considera que hubo algún aprendizaje al utilizar el videojuego. El 10% lo evaluó con un 3, la cual es una respuesta que no se inclina por ninguno de los dos extremos de la respuesta. La amplia aceptación de esta pregunta implica que el videojuego ayudó a los estudiantes a aprender los conceptos básicos de la programación.

En cuanto a la dificultad, el 20% indicó que el videojuego era difícil, mientras que el 40% lo consideró difícil pero no al extremo, mientras que el 40% restante opinó que el videojuego tenía una dificultad moderada. Hay que resaltar que el 60% de los niños consideraron el juego difícil y ningún niño reportó que el juego fuera fácil en algún sentido, lo que conlleva a una futura revisión en la dificultad ya que las pruebas todos la ejecutaron en la dificultad “Fácil”.

En cuanto a la diversión, el 20% de los niños consideró el juego muy divertido, mientras que el 40% lo consideró divertido, pero no al extremo, y el 40% restante consideró que el videojuego guardaba un equilibrio entre muy divertido y

nada divertido. Se resalta que el 40% lo consideró equilibrado, lo que se podría mejorar aplicando algunos conceptos adicionales de gamificación.

Sobre si podrían llegar al final del videojuego, el 70% de los niños manifestaron estar seguros de poder llegar al final de este, mientras que el 30% restante consideró no poder hacerlo. Estas respuestas se consideran interesantes, pues si bien el juego debe ser difícil y representar un reto para los niños, también se debe buscar un equilibrio para que no genere desmotivación. Esto es un concepto extraído del modelo propuesto por [32], donde expresa que, si bien, generar un sentimiento de pérdida puede motivar al jugador, se debe balancear con conceptos positivos, mas no es obligatorio.

Con respecto a si volverían a jugar, al 90% de los niños le gustaría volver a usar el juego y el 10% considera que no lo volvería a usar. Esto muestra una alta aceptación de volver a jugar por parte de los encuestados.

Por lo tanto, el resultado global obtenidos por las cinco variables evaluadas permite concluir que el videojuego desarrollado facilita el aprendizaje de los conceptos de la programación estructurada de una manera divertida y con un grado de dificultad alto.

En cuanto a la última pregunta, las opiniones por parte de los niños sirven como insumo para el mejoramiento del videojuego. Se pueden observar dos opiniones que hacen énfasis en jugar con otras personas (“más jugadores”), lo cual incluye uno de los ítems importantes en gamificación, pues jugar con otras personas genera rivalidad y motivación por superar a los demás. Otra respuesta interesante fue “que sea online”. Mientras que las opiniones “mejores gráficos” y “más niveles” se enfocan en lo visual y en la durabilidad, aspectos que se consideran indispensables ya que el videojuego debe ser atractivo visualmente para el jugador y además debe brindarle una experiencia de juego que sea satisfactoria, en cuanto a duración y cantidad de niveles. Sin embargo, la opinión “más claridad en el sentido de explicar” se refiere a mejorar la usabilidad y la inmersión al juego, temas como la inclusión de tutoriales e introducción básica en los conceptos pueden ser mejorados con más niveles que permitan que los conceptos se puedan explicar de forma paulatina y con un ritmo menos apresurado.

7 Análisis y discusión

El videojuego desarrollado se puede considerar como un videojuego serio ya que “tiene como objetivo usar las ventajas que proporcionan los videojuegos, pero cuyo objetivo fundamental no es el entretenimiento sino el aprendizaje” [33]. Por lo tanto, el videojuego implementa aspectos basados en gamificación.

El videojuego permite a los estudiantes observar y analizar el entorno del juego, asimilar y retener información, realizar razonamientos inductivos y deductivos, y construir y aplicar estrategias cognitivas para solucionar problemas aplicando técnicas de programación estructurada. Además, el videojuego fomenta el desarrollo de la creatividad de los estudiantes al definir posibles soluciones aplicadas en un contexto lúdico de los deportes.

Algunas de las características del videojuego desarrollado que ayudan al aprendizaje de los estudiantes, refuerzan sus conocimientos previos y mejoran sus habilidades son: los diferentes niveles de dificultad abordados de manera progresiva, la posibilidad de detección de errores (encuentros con los rivales, salidas del campo de juego, no alcanzar un objetivo), la planeación y ejecución paso a paso de las soluciones creadas y la retroalimentación inmediata. Otras características tenidas en cuenta durante el desarrollo fueron la narrativa del videojuego orientada al fútbol (uno de los deportes más populares) y los mecanismos de puntuación definidos basados en créditos.

En el contexto de la educación en programación, la mayoría de los juegos se evalúan utilizando comentarios subjetivos recopilados a través de cuestionarios de los estudiantes después de las sesiones de juego [34]. Sin embargo, los estudiantes evaluaron el videojuego en su conjunto y no evaluaron elementos específicos del videojuego.

El diseño de las soluciones a los problemas, planteados en los diferentes niveles del videojuego, está basado en aspectos visuales gráficos y una programación visual táctil que enfatiza en el pensamiento analítico mediante el uso mínimo de la sintaxis de programación.

La retroalimentación visual obtenida en forma de ejecución animada del movimiento del jugador de fútbol o su encuentro con rivales permite a los estudiantes establecer una conexión entre la causa y el efecto de sus acciones. Los estudiantes amplían su conocimiento paso a paso ganando confianza con cada diseño de solución que los acerca a una implementación correcta.

La cantidad de elementos que tiene el videojuego es pequeña, lo cual podría parecer que los niveles del videojuego fueran simples. Empero, los resultados de la encuesta mostraron que los estudiantes evaluaron los niveles jugados como difíciles. Lo que evidencia que el entendimiento de las estructuras de programación utilizadas no es fácil de lograr y aplicar en pocos minutos de juego y que se requiere de experiencia para asimilar los conceptos de programación presentados. Los resultados obtenidos están acordes con los estudios de [35] donde se manifiesta que la utilización de videojuegos serios por parte de estudiantes hace que el aprendizaje se vuelva placentero, con más oportunidades para que el estudiante supere el miedo a aprender la programación de computadores y desmitifique su dificultad.

La validación realizada por medio de la encuesta a los estudiantes arrojó unos resultados muy interesantes. Los estudiantes evaluados manifestaron que aprendieron utilizando el videojuego y lo consideraron divertido. Aunque debido a limitaciones de tiempo, los estudiantes no terminaron todos los niveles, ellos manifestaron que podrían finalizarlo con éxito y les gustaría volver a jugarlo.

El aprendizaje de los temas básicos de programación puede conllevar dificultades al tratar de entender conceptos cognitivos complejos (pensamiento lógico, resolución de problemas y entendimiento de conceptos abstractos). Por consiguiente, el videojuego corrobora los planteamientos de [36] y [37] de que los videojuegos facilitan un número de condiciones que promueven el aprendizaje como: la

motivación del estudiante (mediante la lúdica y diversión), aprendizaje activo (por medio de retos donde se aplican conceptos de programación estructurada) y simulación (con esquemas de un partido de fútbol que simulan las estructuras de programación).

La validación realizada mostró además que el nivel de dificultad fue considerado alto. Este resultado es justificado debido a que el videojuego desarrollado afronta dos grandes problemas que presentan los estudiantes en programación: dificultades para resolver problemas y separar los problemas en pasos en la programación [38], y en estudiantes nuevos en la programación (programadores novatos), tener habilidades de resolución de problemas como requisito previo. Por lo tanto, conocer y aplicar con éxito y sin experiencia previa de estructuras de programación puede plantear dificultades a neófitos y particularmente a estudiantes de grados menores de secundaria.

Los trabajos relacionados en el tercer apartado de este artículo se evaluaron teniendo en cuenta tres parámetros: cobertura, dificultad y efectividad. La cobertura del videojuego desarrollado será bastante amplia, una vez pueda ser descargado desde la tienda online correspondiente. Con respecto a su dificultad, el videojuego propuesto tiene un alto nivel de dificultad, lo cual puede ser problemático cuando se comienza a utilizar el videojuego. Sin embargo, esto puede tomarse como un reto que motive a los jugadores y con la experiencia que vayan ganando con el tiempo, puedan desenvolverse mejor en el cumplimiento de sus objetivos. En cuanto a la efectividad, el videojuego propuesto tiene un enfoque de aprender jugando que facilita el aprendizaje de los conceptos de programación presentados.

Por lo tanto, la evaluación de dichos aspectos para el videojuego realizado por el equipo desarrollador fue de 4 para cobertura y efectividad, y 5 para dificultad.

8 Conclusiones y trabajo futuro

Este artículo presentó un videojuego orientado a dispositivos móviles para el aprendizaje de la lógica de programación. El videojuego desarrollado plantea una serie de niveles basados en tres estructuras computacionales: secuencia, condición e iteración en un contexto deportivo de un partido de fútbol.

Este estudio muestra las posibilidades didácticas que puede tener un videojuego exponiendo sus ventajas para entender o reforzar conceptos básicos de la programación estructurada.

El uso de juegos serios brinda a los estudiantes la posibilidad de hacer que la programación sea divertida incluso en contextos académicos.

El videojuego desarrollado combina elementos de los videojuegos de deporte (simulando acciones que pueden ocurrir en un partido de futbol, en concreto esquivar rivales, trasladar la pelota y marcar goles), videojuegos de estrategia (administrando recursos escasos, para el caso, créditos y previendo los comportamientos de los rivales) y juegos de lógica (planteando posibles soluciones a partir de componentes de programación).

Como trabajo futuro se requiere una validación más compleja del videojuego donde se evalúe la implementación y aprendizaje de los diferentes conceptos de programación incluidos en el videojuego desarrollado. Además, se pueden agregar más conceptos básicos de programación como constantes y variables y temas más avanzados como *arrays* y recursividad hasta llegar a la programación orientada a objetos.

Agradecimientos

Este trabajo ha sido desarrollado en el marco del proyecto “35587-Desarrollo de un videojuego orientado al aprendizaje de soluciones lógicas computacionales en estudiantes de básica-secundaria”, financiado por la Universidad Nacional de Colombia Sede Manizales y desarrollado por el Grupo de Aplicaciones y Herramientas Web del Departamento de Informática y Computación.

Referencias

[1] T. Mitamura, Y. Suzuki y T. Oohori, «Serious games for learning programming languages,» de *2012 IEEE international conference on systems, man, and cybernetics (SMC)*, 2012.

[2] Á. Serrano-Laguna, J. Torrente, B. M. Iglesias y B. Fernández-Manjón, «Building a scalable game engine to teach computer science languages,» *EEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 10, nº 4, pp. 253-261, 2015.

[3] G. Petri, C. G. von Wangenheim y A. F. Borgatto, «Quality of games for teaching software engineering: an analysis of empirical evidences of digital and non-digital games,» de *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)*, 2017.

[4] R. T. Pereira, A. C. Torres, J. C. C. Mora, J. J. Toledo, H. O. Erazo y C. Colunge, «Un cambio de paradigma en la enseñanza de fundamentos de programación en ingeniería de sistemas,» *Revista Educación en Ingeniería*, vol. 4, nº 7, pp. 120-128, 2009.

[5] H. Tsalapatas, «Programming Games for Logical Thinking,» *EAI Endorsed Transactions on Serious Games*, vol. 1, nº 1, pp. 1-7, 2013.

[6] N. Prokofyeva, M. Uhanova, S. Katalnikova, K. Synytsya y A. Jurenoks, «Introductory programming training of first year students,» *Procedia Computer Science*, vol. 104, pp. 286-293, 2017.

[7] B. Özmen y A. Altun, «Undergraduate students' experiences in programming: difficulties and obstacles,» *Turkish Online Journal of Qualitative Inquiry*, vol. 5, nº 3, pp. 1-27, 2014.

[8] F. Lewandowski y A. Soares, «Desenvolvimento de jogos educacionais apoiados por um agente tutor pedagógico,» *Nuevas Ideas en Informática Educativa*, vol. 4, pp. 7-14, 2008.

[9] V. Roadrangka, «The construction and validation of Group Assessment of Logical Thinking (GALT),» de *In Paper Presented at the National Association for Research in Science Teaching Annual Meeting.*, 1983.

[10] E. F. Iepsen, M. Bercht y E. Reategui, «Detection and assistance to students who show frustration in learning of algorithms,» de *2013 IEEE Frontiers in Education Conference (FIE)*, 2013.

[11] M. Othman y Z. Nurzaid, «Online Collaboration for Programming: Assessing Students' Cognitive Abilities,» *Turkish Online Journal of Distance Education*, vol. 16, nº 4, pp. 84-97, 2015.

[12] S. Grover y R. Pea, «Computational thinking in K–12: A review of the state of the field,» *Educational researcher*, vol. 42, nº 1, pp. 38-43, 2013.

[13] J. Moreno, «Digital competition game to improve programming skills,» *Journal of Educational Technology & Society*, vol. 15, nº 3, pp. 288-297, 2012.

[14] D. Vrajitoru y P. Toprac, «Games Programming in Computer Science

Education,,» de *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS) (*, 2016.

[15] L. E. Winslow, «Programming pedagogy – A psychological overview,» *ACM Sigcse Bulletin*, vol. 28, nº 3, pp. 17-22, 1996.

[16] A. Gomes y A. J. Mendes, «An environment to improve programming education,» de *In Proceedings of the 2007 international conference on Computer systems and technologies*, 2007.

[17] B. R. Gifford, «The learning society: Serious play,» *Chronicle of Higher Education, 1991*, 1991.

[18] Á. Pérez García, «El aprendizaje con videojuegos: experiencias y buenas prácticas realizadas en las aulas españolas,» *Revista de Investigación Educativa*, vol. 17, pp. 135-156, 2014.

[19] E. M. Rodrigo y M. L. Carmen, Jóvenes interactivos: Nuevos modos de comunicarse, Netbiblo, 2011.

[20] K. Werbach y D. Hunter, For the win: How game thinking can revolutionize your business, Wharton digital press, 2012.

[21] I. Sheridan, D. Goggin y L. O'Sullivan, «Exploration of learning gained through CoderDojo Coding Activities,» 2016.

[22] J. Du, H. Wimmer y R. Rada, «" Hour of Code": Can It Change Students' Attitudes Toward Programming?,» *Journal of Information Technology Education: Innovations in Practice*, vol. 15, pp. 52-73, 2016.

[23] Schneider, G.S.; Martins de Souza, E. M.; Ferreira Gomes, L., «Ensino e Aprendizado de Lógica Através do Scratch,» de *V Seminário Nacional de Inclusão Digital*, 2018.

[24] E. W. Patton, M. Tissenbaum y F. Harunani, MIT app inventor: Objectives, design, and development, Singapore: Springer, 2019, pp. 31-49.

[25] J. M. Costa y G. L. Miranda, «Relation between Alice software and programming learning: A systematic review of the literature and meta-analysis,» *Relation between Alice software and programming learning: A systematic review of the literature and meta-analysis, British Journal of Educational Technology*, vol. 48, nº 6, pp. 1464-1474, 2017.

[26] S. Pila, F. Aladé, K. J. Sheehan, A. R. Lauricella y E. A. Wartella, «Learning to code via tablet applications: An evaluation of Daisy the Dinosaur and Kodable as learning tools for young children,» *Computers & Education*, vol. 128, pp. 52-62, 2019.

[27] C. Böhm y G. Jacopini, «Flow diagrams, Turing machines and languages with only two formation rules,» *Communications of the ACM*, vol. 9, nº 5, pp. 366-371, 1966.

[28] A. Subagio, Learning Construct 2, Birmingham, Reino Unido: Packt Publishing Ltd, 2014.

[29] S. M. Freund, S. L. Sebok y M. E. Vermaat, «Discovering Computers: Technology in World of Computers, Mobile Device, and The Internet,» Cengage Learning, Boston USA, 2014.

[30] H. M. Johnsen, M. Fossum y P. F. A. a. S. Å. Vivekananda-Schmidt, «Teaching clinical reasoning and decision-making skills to nursing students: Design, development, and usability evaluation of a serious game,» *International journal of medical informatics*, vol. 94, pp. 39-48, 2016.

[31] J. E. N. Lino, M. A. Paludo, F. V. Binder, S. Reinehr y A. Malucelli, «A serious game to assist software project managers training,» de *IEEE Frontiers in Education Conference (FIE)*, 2015.

[32] Y. K. Chou, Actionable gamification: Beyond points, badges, and leaderboards, Packt Publishing Ltd. , 2019.

[33] B. G. Salvat, «El uso de los videojuegos para la formación universitaria y corporativa,» *Comunicación y pedagogía: Nuevas tecnologías y recursos didácticos*, vol. 239, pp. 14-18, 2009.

[34] G. Petri y W. G. C., «How games for computing education are evaluated? A systematic literature review,» *Computers & Education*, vol. 107, pp. 68-90, 2007.

[35] A. L. dos Santos, R. D. A. Mauricio, E. Figueiredo y M. Dayrell, «Game Elements for Learning Programming: A Mapping Study,» de *CSEDU (2)*, 2018.

[36] C. Johnson, M. McGill, D. Bouchard, M. K. Bradshaw, V. A. Bucheli, L. D. Merkle, M. Scott, Z. Sweedyk, J. Velázquez-Iturbide, X. Ziphing y M. Zhang, «Game development for computer science education,» de *Proceedings of the 2016 ITiCSE Wo*, 2016.

[37] M. Zušek y J. Rugelj, «Learning programming with serious games,» *EAI Endorsed Transactions on Serious Games*, vol. 13, nº 1.

[38] P. H. Tan, C. Y. Ting y S. W. Ling, «Learning difficulties in programming courses: Undergraduates' perspective and perception,» de *Kota Kinabalu*, Malaysia, 2009.

Leonardo Bermón Angarita es Ph. D. en Ingeniería Informática de la Universidad Carlos III de Madrid, España (2010), Magister en Informática (2001) e Ingeniero de Sistemas de la Universidad Industrial de Santander (UIS) de Bucaramanga, Colombia (2001). Actualmente es Profesor Asociado del Departamento de Informática y Computación de la Universidad Nacional de Colombia – Sede Manizales. Sus principales áreas de investigación son Ingeniería de software, Desarrollo Web y Gestión del Conocimiento. Pertenece al Grupo de Investigación en Aplicaciones y Herramientas Web.
ORCID: 0000-0002-6034-0483.

Amparo Prieto Taborda es Magíster en Administración de la Universidad Nacional de Colombia – Sede Manizales, Colombia e Ingeniera de Sistemas de la Universidad Autónoma de Manizales, Colombia. Actualmente es Profesora Asistente del Departamento de Informática y Computación de la Universidad Nacional – Sede Manizales. Sus principales áreas de investigación son la Planeación de sistemas de información e Impacto social de las TIC. Pertenece al Grupo de Investigación en Aplicaciones y Herramientas Web.

Juan Diego Escobar Márquez es Administrador de Sistemas Informáticos de la Universidad Nacional de Colombia – Sede Manizales, Colombia (2017). Con experiencia en gestión y desarrollo de proyectos de base tecnológica y enfocados a los sistemas de información. Áreas de interés en desarrollo de software, especialmente orientado a la web, e implementación de tecnologías como automatización de procesos con RPA y BPMS, inteligencia artificial y desarrollo de soluciones en plataformas cloud. Actualmente trabaja en ARUS Tecnología + Información + Conocimiento en Medellín, Antioquía

Juan David Vergara Díaz es Administrador de Sistemas Informáticos de la Universidad Nacional de Colombia - Sede Manizales, Colombia (2018). Sus principales áreas de interés son el desarrollo de software y la automatización de procesos por medio RPA. Su experiencia abarca la automatización de procesos para el sector bancario y el desarrollo de soluciones informáticas para el sector de las aerolíneas. Actualmente trabaja en la compañía MasGlobal Consulting como desarrollador de software para un cliente que forma parte de la lista Fortune Global 500.