

Videjuego educativo para ayudar a comprender los principios básicos de la programación y desarrollar la habilidad lógica en niños de educación básica

Ana Caren Ticante Hernández, Carlos Miguel Herrera Orduña, Pedro Arguijo, Roberto Ángel Meléndez Armenta, Antonio Hiram Vázquez López

Tecnológico Nacional de México / Instituto Tecnológico Superior de Misantla,
Departamento de Maestría en Sistemas Computacionales
Misantla, Veracruz, México

anacarenticante@gmail.com, herrera.orduna.carlosmiguel@gmail.com,
pedro_arguijo@excite.com, ramelendeza@itsm.edu.mx,
jahvazquezl@itsm.edu.mx

Resumen. Debido a la dificultad de aprender la lógica necesaria para la programación se diseña y desarrolla un videjuego educativo que tiene como objetivo ayudar a niños a introducirse en el mundo de la programación. El videjuego busca desarrollar la lógica de los infantes en etapa de primaria y la forma en que se realiza es dando al jugador el control de un personaje el cual se maneja mediante líneas de pseudocódigo que deben ser escritas en un campo de texto de manera correcta. El videjuego consta de 20 niveles y su dificultad es incremental. La metodología de desarrollo utilizada es Scrum, además, se utilizaron herramientas de software, como el motor gráfico Unity 3D, Asesprite, Tiled y Adobe Photoshop para el diseño y desarrollo del videjuego.

Palabras clave: videjuego, aprendizaje reforzado, Q-learning, educación, programación.

Educational Game to Help Understand Basic Topics of Programming and Develop Logical Skills in Children of Basic Education

Abstract. Due to the difficulty in learning the logic necessary for programming, an edu-game is designed and developed which the purpose to help children get started in the world of programming. The videogame seeks to develop the logic of kids in the basic education, the way it is done is giving to the player the control of a character which is handled by pseudocode lines that must be written in a text field so correct. The video game consists of 20 levels with an incremental difficulty. This was created based on the Agile Scrum methodology and software tools were used, such as the Unity 3D graphics engine, Asesprite, Tiled and Adobe Photoshop for the design and development of the videogame.

Keywords: videogame, reinforcement learning, Q-learning, education, programming

1. Introducción

Actualmente, la tecnología se ha convertido en la mano derecha de las personas, esta ayuda a eliminar obstáculos de comunicación, ha permitido avances importantes en medicina, contribuye en la efectividad de los procesos empresariales y permite implementar nuevas técnicas de enseñanza, entre muchas otras aplicaciones que mejoran la calidad de vida de la sociedad. Gran parte de estas tecnologías operan con base en código escrito en un lenguaje de programación que permite automatizar tareas tediosas y reducir el tiempo en el cual se ejecutan los procesos. La programación se encuentra en la vida cotidiana de la sociedad, por lo que actualmente, es común referirse a los jóvenes como “nativos digitales” que tienen la capacidad de diseñar, inventar y crear con la computadora [1]. Es por lo que la enseñanza de la programación es vital para el desarrollo tecnológico de la sociedad.

La psicología educativa establece que la edad apropiada para aprender a programar depende de la complejidad del código de programación, se han realizado varios estudios sobre cómo los estudiantes jóvenes utilizan la codificación de Entornos de Aprendizaje Inicial (ILE), y estos estudios proporcionan pistas sobre el tipo de programación que están realizando, hay evidencia de que algunos conceptos de programación pueden enseñarse a estudiantes de 5 a 7 años de edad mientras el lenguaje utilice conceptos accesibles para ese grupo de edad [2]. Además, debe convertirse en una actividad de aprendizaje, debe resultar divertido y motivacional, por lo que los videojuegos son una herramienta que puede aportar distintos beneficios a este proceso de aprendizaje.

Desde su aparición, los videojuegos son parte del desarrollo social y cultural de las personas y llegan a influir en su desarrollo físico, sensorial, mental y creativo. Dada su importancia, en la vida cotidiana, se han creado videojuegos educativos con el objetivo de generar estrategias y métodos de enseñanza alternativos a los tradicionales, animando a los infantes y jóvenes a obtener conocimiento con una herramienta atractiva y motivadora a través de retos y actividades que van presentando dichos videojuegos. En este artículo se describe el diseño y desarrollo de un videojuego capaz de ayudar a comprender los principios básicos de la programación para el desarrollo de la habilidad lógica en niños de educación básica.

2. Trabajos relacionados

Un videojuego es un juego digital interactivo, creado con fines de entretenimiento en general y se basa en la interacción con personas o personajes ficticios.

Los videojuegos pueden clasificarse de acuerdo a su género, existen juegos de acción donde su objetivo es básicamente eliminar enemigos, de tipo arcade, son juegos de plataformas, laberintos y aventuras, también existen los videojuegos deportivos que promueven la competitividad y valores propios de los deportes, los de estrategia que hacen uso de planificación de recursos o acciones para lograr un objetivo, los de simulación que se caracterizan por permitir al jugador experimentar e investigar el funcionamiento de máquinas, fenómenos, situaciones y asumir el mando. También existen los juegos serios que están diseñados con un propósito más formativo que de entretenimiento.

Los juegos serios tienen principalmente un propósito educativo, explícito y cuidadosamente planeado, este tipo de juegos ayudan a la comprensión y aprendizaje de nuevos conceptos y habilidades. La taxonomía de los juegos serios se rige por siete modalidades [3], juegos para la salud, publicitarios, para la formación, para la ciencia y la investigación, educativos y de producción.

Los juegos serios se caracterizan por hacer uso de simulaciones principalmente para el ramo empresarial, un ejemplo es Virtual Safety and Education Platform [4], que es un juego que tiene como finalidad capacitar operarios de servicios de emergencia, operarios de maquinaria, personal de hospitales, ejercito, etc., SimCityEdu, es otro videojuego basado en las simulaciones, en el que estudiante de primaria debe crear y gestionar una ciudad, trabajando conceptos de ciencia, tecnología, matemáticas y economía [5]. Los juegos serios pueden usar cualquier tecnología de juegos y estar desarrollados para cualquier plataforma, además, están dirigidos a todo público, principalmente a profesionales, consumidores y estudiantes.

En el ámbito educativo, los juegos son una herramienta útil para la obtención de conocimiento, donde a través de retos y actividades, el estudiante es el protagonista de su propio aprendizaje. Existen videojuegos educativos para aprender muchas cosas, los más básicos enseñan a leer, sumar, restar o hasta tocar instrumentos, aprender algún idioma, o sobre temas más específicos como la biología, tal es el caso de Proyecto Kokori, este videojuego pone a los jugadores al mando de una pequeña nave capaz de introducirse en una célula para estudiarla, además de identificar y solucionar los problemas que puedan ir encontrando [6].

Incluso existen videojuegos para el área médica, por ejemplo, Immune Attack, su objetivo es salvar la vida de un paciente que sufre una grave infección bacteriológica, para ello, los jugadores tratan de encontrar todos los focos de la infección y devolver la salud al paciente, además, los jugadores acceden y ponen en práctica conocimientos sobre los procesos biológicos que permiten detectar y combatir las infecciones [7], otra aplicación de los juegos educativos, es la enseñanza de la programación.

El aprendizaje de conceptos de programación no es como cualquier otro aprendizaje, no es un proceso definido y no se trata sólo de memorización, para este propósito los videojuegos educativos pueden ser una herramienta efectiva, donde el jugador puede obtener conocimientos de programación en un ambiente educativo-virtual e interactivo [8], donde pueden aumentar la motivación para el aprendizaje y mejorar la comprensión y adquisición de nuevos conocimientos.

Existen distintos juegos enfocados a enseñar lenguajes de programación específicos, por ejemplo, Code combat, este es una es una plataforma interactiva en línea, un juego con estilo de rol enfocado a niños de más de 9 años o adultos y está pensado para que los jugadores puedan aprender a programar lenguajes de programación específicos, por ejemplo, JavaScript acompañado del lenguaje de etiquetas HTML, en Code combat el jugador toma el papel del personaje que el elija, y mediante hechizos y acciones a través de comandos de programación el juego avanza con diferentes niveles, aumentando el nivel complejidad [9].

Otro videojuego es CodinGame, este juego funciona sobre el navegador, el cual le permite al jugador practicar sus habilidades de programación combinándolas con un videojuego de disparos y aventura. En este juego se derrotan enemigos, programando comandos en la pantalla que se asemeja a una consola [10].

3. Materiales y métodos

La investigación que se reporta es de carácter cualitativo y su principal objetivo es explorar el uso de ambientes virtuales de videojuegos para el desarrollo de la lógica y de las habilidades de programación.

3.1. Materiales

- Hardware: computadora con requerimientos mínimos, uso mínimo de 2Gb de memoria RAM y 4 núcleos de procesador 1.2 Giga Hertz.
- Escenarios: como escenario se definen los mapas del videojuego con los que interactúa el jugador.
- Participantes: Los participantes se eligieron con base en el público objetivo, por lo que se contactó a la escuela primaria urbana "Niños Héroes" ubicada en la colonia Espaldilla de Misantla, Veracruz. Los participantes fueron estudiantes de quinto y sexto de primaria, como se puede ver en la figura 1, participaron 4 niñas y 2 niños de 10 años, 7 niñas y 11 niños de 11 años, 3 niñas y 1 niño de 12 años, contabilizando un total de 28 participantes.

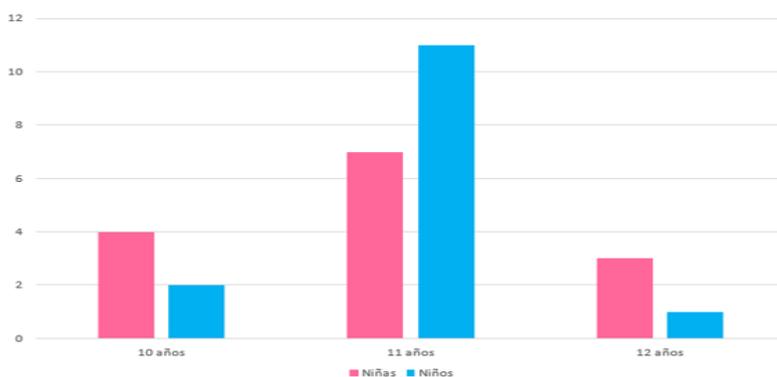


Fig. 1. Rango de edades de los participantes.

3.2. Herramientas de desarrollo

- Unity: Es un motor de videojuego multiplataforma creado por Unity Technologies. Está disponible como plataforma de desarrollo para Microsoft Windows, OS X, Linux. La plataforma tiene soporte de compilación con diferentes tipos de plataformas [11].
- TiledMap: Tiled es un editor de nivel de software libre. Admite la edición de mapas en varias proyecciones (ortogonal, isométrica, hexagonal) y también admite niveles de construcción con imágenes libremente posicionadas, giradas, escaladas o anotándolas con objetos de diversas formas [12].
- Adobe Photoshop: Adobe Photoshop es un editor de gráficos desarrollado por Adobe Systems Incorporated y utilizado principalmente para el retoque de fotografías y gráficos. Traducido al español significa “taller de fotos” y es el

líder mundial dentro del mercado de las aplicaciones de edición de imágenes en general [13].

3.3. Métodos

Metodología de desarrollo de software. Una metodología de desarrollo de software es una serie de pasos que se realizan de tal forma que permite a los desarrolladores elegir las técnicas para cada fase del proyecto y definir principalmente los participantes, sus actividades a desarrollar y una estructura de tiempo en el que deben realizarse. Seguir una metodología de desarrollo facilita la planificación, control, gestión de requisitos, operación, mantenimiento y validación de proyectos [14]. El conjunto de etapas y subetapas que sigue un proyecto, se denominan ciclo de vida del software, éste es fundamental para elegir la metodología de desarrollo a seguir ya que define las necesidades desde su concepción, los cambios o ajustes que puedan presentarse, el mantenimiento necesario, la realización de pruebas, hasta llegar a su finalización.

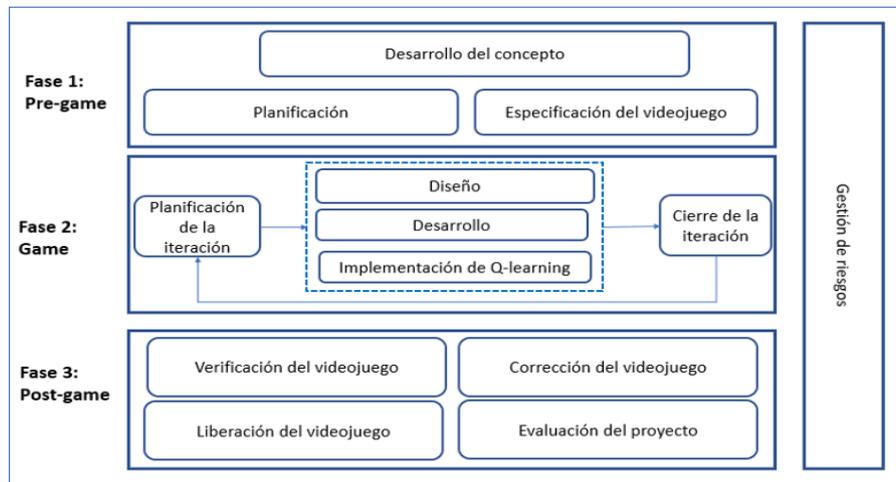


Fig. 2. Diagrama de la metodología Scrum para el desarrollo de videojuegos.

Se eligió utilizar la metodología de desarrollo ágil Scrum, porque demuestra un desarrollo que se adapta dinámicamente a las necesidades y requerimientos de la creación de un videojuego, además, se pueden realizar tareas simultáneamente, permite flexibilidad, mitigación de riesgos y adaptación a los cambios a los que se pueda someter en todo su ciclo de vida. Scrum permite trabajar en una serie de iteraciones en equipo.

El proceso de Scrum para este desarrollo se define en 3 fases secuenciales: pre-game, game y postgame, la fase de pre-game se realiza en una única iteración que comprende la definición del concepto del juego y la planificación de este, la fase game comprende a la implementación del videojuego y se trabaja de forma iterativa e incremental, la última fase tiene como objetivo evaluar y ajustar distintos aspectos del juego, eliminar errores detectados y, por último, entregar la versión final [15]. En la figura 3, se muestra un diagrama de la metodología aplicada en el videojuego.

Algoritmo de aprendizaje. El aprendizaje por refuerzo es un área del aprendizaje automático, el cual consiste en que un aprendiz recibe información de lo que es apropiado y con respecto a esto, realice una acción, en el caso de que la acción sea correcta, el aprendiz recibe algún tipo de valoración o premio, ante circunstancias erróneas, no se le notifica cual es el fallo, sino que, se informa que el comportamiento no fue apropiado y cuanto error se ha cometido. Este trabajo se centra en la técnica Q-learning, una forma de aprendizaje por refuerzo.

El algoritmo de Q-learning basado en el aprendizaje por refuerzo y utilizado en el aprendizaje automático, cuya ocupación es determinar qué acciones se deben escoger en un entorno dado, con el fin de maximizar alguna noción de "recompensa" o premio acumulado, esta recompensa representa la "calidad" de una acción tomada en un estado, en este caso, la recompensa representa la calidad de las soluciones introducidas por el jugador en el videojuego, esto con el objetivo de validar las soluciones. La introducción de esta técnica permite, evaluar la calidad de conocimiento obtenido y aportar una recompensa que pueda indicarle al usuario que ha llevado a cabo una acción correcta. En la figura 4, se muestra la representación del algoritmo de aprendizaje por refuerzo, implementado en el videojuego.

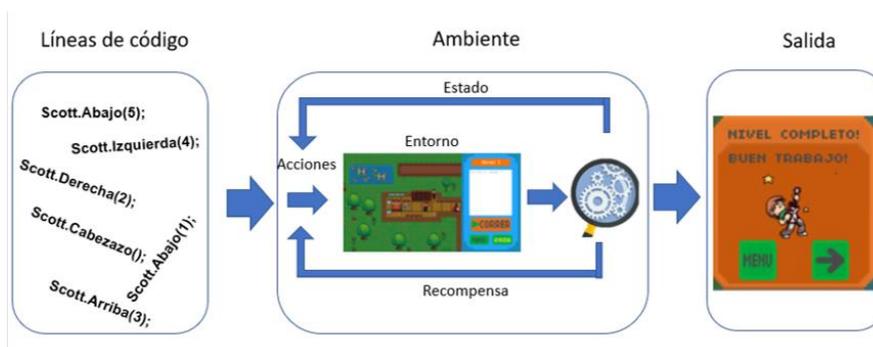


Fig. 3. Representación del algoritmo Q-learning en el videojuego.

4. Resultados

El resultado final consistió en el diseño y desarrollo de un videojuego, se dividió en 3 principales fases, pregame, game y posgame, de acuerdo con la metodología de desarrollo planteada.

4.1. Pre-game

La primera fase consistió en el planteamiento del concepto, la planificación y las especificaciones del videojuego.

Título. El título del videojuego es "Code Adventure".

Género. "Code Adventure" se considera un juego de estrategia ya que se caracteriza por manipular personajes para lograr objetivos haciendo uso de la inteligencia y la

planificación. También contiene, aunque a menor medida, características de los juegos de rol, ya que se interactúa con el personaje y se complementa con una historia. Por otro lado, el género puede considerarse como educativo, ya que está dirigido a aportar conocimientos y habilidades al jugador durante el transcurso del juego.

Arquitectura. El videojuego puede ser jugado desde cualquier sistema operativo sobre plataforma PC.

Mecánica. En esta fase también se definió la mecánica del videojuego, en la figura 5 se muestra una vista general de la pantalla de juego del nivel 1, misma que es similar para los demás niveles, únicamente cambiando el mapa, la posición y existencia de enemigos y obstáculos.

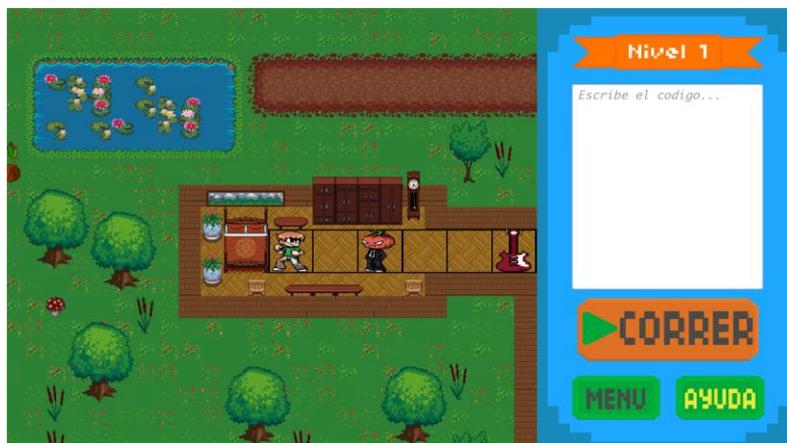


Fig. 4. Nivel 1 del videojuego.

Para completar el nivel el personaje principal debe pasar por cada uno de los obstáculos que se le presenten en el mapa, éstos pueden ser enemigos o rocas que deberá de destruir, también debe conseguir la guitarra de cada nivel, la función de la guitarra es orientar al jugador hacia el final del mapa y como un incentivo que motive al usuario a concluir todos los niveles del videojuego. En la figura 6 se muestran todas las guitarras diseñadas.

El jugador debe de introducir una serie de comandos que se traducirán en acciones que el personaje realizará. En la pantalla del juego existe un campo de texto donde el usuario debe escribir el código, que será verificado. Las líneas introducidas en el campo de texto permitidas son 5, hay cuatro acciones relacionadas con movimiento, al ejecutarse éstas permiten el desplazamiento del personaje hacia la izquierda, derecha, arriba o abajo, dependiendo de la acción que se trate, el jugador debe introducir el número de pasos que dará el personaje hacia la dirección indicada, el juego permite un número de pasos entre 1 y 20.

El personaje también puede realizar una acción de ataque, éste se realiza hacia donde se encuentre orientado el personaje, en la tabla 1, se muestra la sintaxis de las acciones posibles, donde “#” corresponde al número de pasos.

Tabla 1. Acciones posibles.

Acción	Sintaxis
Moverse a la derecha.	Scott.Derecha(#);
Moverse a la izquierda.	Scott.Izquierda(#);
Moverse hacia abajo.	Scott.Abajo(#);
Moverse hacia arriba.	Scott.Arriba(#);
Ataque	Scott.Cabezazo();



Fig. 5. Objetos recompensa.

Después de que el jugador escribe líneas de código en el panel de texto, debe de ejecutarlo, a continuación, el código se analiza para evaluar cada una de las líneas introducidas y encontrar posibles errores de sintaxis, si las líneas de código introducidas son correctas estas proceden a interpretarse, traduciéndose en acciones que realiza el personaje del juego.

Cámara. Para el caso de “Code Adventure” la plataforma de desarrollo es en 2D, lo recomendado para este tipo de juegos es una cámara fija, su posición, distancia focal y campo de visión no son cambiantes, por lo que, además, permite consumir menos recursos.

Periféricos. Ya que el videojuego es para PC, los periféricos fundamentales para jugar son teclado y ratón. El teclado permite capturar las instrucciones dadas por el jugador y el ratón permite desplazarse por las diferentes pantallas y niveles, así como poner en marcha las líneas de código escritas, haciendo clic en los botones que se encuentran en el videojuego.

Dificultad. Para determinar la dificultad de un videojuego se requiere considerar aspectos como el diseño de los escenarios, la historia, el modo y el ritmo de juego. Para el caso de “Code Adventure”, se determinó que la dificultad sea incremental, que los primeros niveles se resuelvan con pocas líneas de código y al avanzar se requieran de más líneas de código para resolver niveles más complejos. Por lo que la dificultad en este juego se compone por fácil, medio y difícil.

4.2. Game

En esta fase, se diseñaron los elementos visuales, así como las funcionalidades del videojuego.

Diseño. Se diseñaron los botones, íconos y demás elementos que componen las interfaces, como recursos que sirven para mostrar información al usuario para que logre identificar los diferentes niveles del videojuego, así como paneles contenedores de los botones, textos y demás elementos de la interfaz.



Fig. 6. Distribución de contenido.



Fig. 7. Mapa general de los niveles del videojuego.

Se diseñaron interfaces para la navegación dentro del videojuego, la primera corresponde al menú principal donde se pueden seleccionar las opciones de continuar, seleccionar un nivel en específico información sobre los desarrolladores y la opción de salir. Se diseñó una interfaz para que el usuario pueda seleccionar un nivel específico para jugar. También, se diseñó la interfaz que muestra la información de los desarrolladores y un botón para regresar al menú principal. Para la pantalla de juego principal se distribuyó el contenido como se puede observar en la figura 7, tomando como base una resolución de 1280x720 píxeles.

Desarrollo. En el proceso de desarrollo se añadió la funcionalidad al videojuego, desde poder navegar entre las interfaces del videojuego a través de los botones, se añadió un botón para mostrar información sobre la sintaxis del videojuego. Se añadió una función al botón de “Correr”, cuyo objetivo es, notificar que el usuario ha introducido líneas de

código, para que sean analizadas, también se desarrollaron todas las animaciones necesarias para el videojuego. Así, si el usuario introdujo las líneas de código correctas, se pueda verificar a través de movimientos realizados por el personaje al recorrer el mapa de acuerdo con las acciones introducidas.

Se diseñaron 20 niveles, los primeros doce se desarrollan dentro de la casa de Scott y los ocho restantes se ambientan en jardines y áreas verdes. Para el diseño de cada uno de estos niveles se usó la herramienta Tiled que permite construir mapas utilizando los tiles descargados previamente. Posteriormente, dichos mapas se exportaron a la plataforma de desarrollo del videojuego. En la figura 8 se muestra el mapa final que compone los 20 niveles.

4.3. Postgame

La última fase comprende la evaluación y verificación del software. Las pruebas de usabilidad son uno de los métodos más comunes para medir un videojuego. La usabilidad es un atributo de calidad que evalúa la calidad de experiencia del usuario e identifica la facilidad de uso de un sistema; en una prueba de usabilidad el videojuego se presenta a un grupo de usuarios objetivo, es decir, al público al que va dirigido principalmente el juego.

Lo que se busca es identificar posibles errores que los desarrolladores hayan pasado por alto, no es necesario presentar el producto final, basta con un prototipo o una versión avanzada del software, ya que se recomienda que dicha prueba se aplique al menos tres semanas antes de la entrega final [16].

La validación del videojuego se llevó a cabo mediante una prueba de usabilidad, la cual se desarrolló a partir de la implementación del prototipo del videojuego que incluye dos niveles de cada grado de dificultad dando un total de 6 niveles. Este prototipo fue aplicado a estudiantes de quinto y sexto de primaria, dichos participantes están descritos con más detalle en la sección 3.1. La manera en la que se llevó a cabo la prueba fue la siguiente, se dio una introducción a los estudiantes acerca de la importancia de su participación en esta actividad y de los objetivos del videojuego, además, se resolvieron todas las dudas existentes. Cabe mencionar que se contaba con escasas 3 PC y ya que el tiempo asignado para esta actividad fue limitado, debido a las diversas actividades programadas para ese día escolar, se optó por agrupar a los estudiantes en parejas de manera aleatoria y se dio la instrucción de que se alternarían para resolver los niveles por sí mismos, así, los estudiantes pudieron interactuar con cada grado de dificultad del videojuego en un lapso más corto.

Se diseñó una encuesta de 10 preguntas para medir la satisfacción de los estudiantes ante el videojuego, la cual fue aplicada después de que todos interactuaran con él. Para dicha encuesta se utilizaron preguntas con un lenguaje sencillo para que los estudiantes pudieran comprenderlas fácilmente, además, se incluyeron respuestas de opción múltiple por la misma razón y para que la información pudiera ser plasmada estadísticamente, las cuales pueden observarse en la tabla 2.

Para obtener datos de la encuesta, se utilizó una escala en la que a cada posible respuesta se le asigna un valor, por ejemplo: no=1, tal vez=2, si=3, dichos valores se encuentran asignados en la tabla 2. En algunas de las preguntas se pedía asignar una calificación del 1 al 5, porque lo que no fue necesario realizar una conversión.

Tabla 2. Prueba de usabilidad.

NUM	PREGUNTAS	POSIBLES RESPUESTAS				
1	¿Te gustó el juego? Califícalo del 1 al 5.	1	2	3	4	5
2	Del 1 al 5, ¿Qué tan fácil es de jugar?	1	2	3	4	5
3	¿Es divertido?	1. Poco	2. Regular	3. Mucho		
4	¿Te gustó el minijuego?	1 Poco	2.Regular	3.Mucho		
5	Del 1 al 5, ¿Qué tanto te agradó el personaje?	1	2	3	4	5
6	¿Volverías a jugarlo?	1 No	2.Tal vez	3. Si		
7	¿Lo recomendarías con tus amigos?	1. No	2. Tal vez	3. Si		
8	¿Aprendiste algo nuevo?	1. No	2. Tal vez	3. Si		
9	¿Encuentras interesante el mundo de la programación después de jugar este juego?	1. Poco	2. Regular	3. Mucho		
10	¿Calificación final del videojuego?	1	2	3	4	5

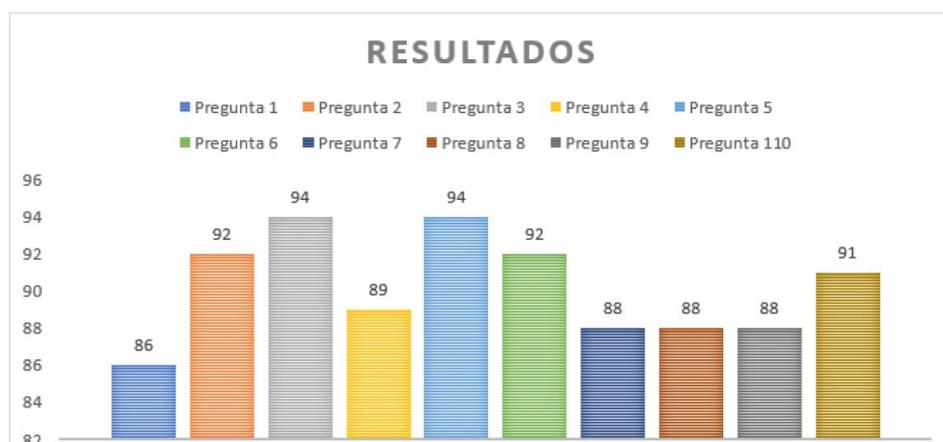


Fig. 8. Porcentajes obtenidos.

Una vez teniendo solamente valores ordinales, se obtuvieron los promedios de cada una de las preguntas, los cuales se muestran en la figura 9.

En conclusión, se desarrolló un videojuego 100% funcional, con un total de 20 niveles con distintos grados de dificultad, para la evaluación del software obtenido, se realizaron las pruebas pertinentes, se visitó a una escuela primaria de la ciudad de Misantla, Veracruz, donde 28 estudiantes de dicha institución educativa interactuaron con el software y posteriormente realizaron una encuesta de satisfacción.

Una vez realizadas las validaciones y cada una de las pruebas, se identifica que el juego satisface con las acciones verificadas anteriormente, brindando confiabilidad y funcionalidad en cada uno de los niveles del juego.

Se evidencia el correcto funcionamiento de los elementos del videojuego, dando un visto bueno en cada una de las pruebas realizadas y confirmando que el juego cumple con cada uno de los objetivos esperados.

5. Discusión

Para tener una perspectiva de “Code Adventure” en comparación a lo que ofrecen otros videojuegos educativos, se revisaron las diferentes propuestas de videojuegos educativos, principalmente, aquellos que tienen como objetivo ayudar a personas a aprender a programar, se analizaron factores como: plataforma, público objetivo, dificultad, tópicos de programación que tratan y sus costos. En la tabla 3 se muestra una comparación entre algunos de los principales videojuegos analizados.

Tabla 3. Comparación entre videojuegos para aprender a programar.

Videojuego	Plataforma	Lenguajes que enseña	Público objetivo	Costo
Code Monkey	PC WEB	CoffeeScript	de 10 a 12 años	29 dólares anuales
CodeCombat	PC WEB	Javascript Python.	de 10 a 17 años	39.99 dólares
Tynker	PC WEB Android	Python, HTML/CSS, JavaScript	de 7 a 9 años	\$20-\$180 dólares
CheckiO	PC WEB	Python JavaScript	de 17 a 18 años	24.99 dólares anuales
Code adventure	Pc Escritorio	Pseudocódigo	De 7 a 12 años	Gratis

Analizando la comparación presentada en la tabla 3, Code Adventure se diferencia en el tipo de plataforma, ya que, al ser una aplicación instalable offline no requiere de una conexión a internet, esto puede traer beneficios, ya que escuelas y hogares que no cuenten con este servicio pueden utilizar el videojuego.

En cuanto a los lenguajes que enseñan todos los videojuegos comparados se enfocan en enseñar lenguajes de programación específicos, Code Adventure tiene un enfoque

completamente distinto puesto que, se basa en enseñar la lógica computacional necesaria para comprender los principios básicos de la programación y así aplicar este método de razonamiento a cualquier lenguaje de programación. Por otro lado, los costos de los videojuegos analizados comprenden desde los 20 hasta los 180 dólares, costos que pueden no ser accesibles para algunos usuarios, por el contrario “Code Adventure” al ser una plataforma desarrollada en una institución educativa se realizó sin fines de lucro.

Referencias

1. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y.: Scratch: Programming for all. *Commun. Acm*, 52(11), 60–67 (2009)
2. Duncan, C., Bell, T., Tanimoto, S.: Should your 8-year-old learn coding? In: *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*, pp. 60–69 (2014)
3. Gros Salvat, B.: Certezas e interrogantes acerca del uso de los videojuegos para el aprendizaje. *Revista Internacional de Comunicación Audiovisual, Publicidad y Literatura* 1(7), 251–264 (2009)
4. VStep, RescueSim-Virtual Emergency Response Training, www.rescuesim.com, Último acceso: 2018/12/03
5. Hao, J., Mislevy, R.J.: The Evidence Trace File: A Data Structure for Virtual Performance Assessments Informed by Data Analytics and Evidence-Centered Design (2018)
6. Mellado, M., Roa, J., Baez, M., Carpinelli, J., Garretón, V., Mercovich, E., Szwarcberg, M.: Kokori, set de herramientas TIC gratuito para la enseñanza y aprendizaje en biología celular (1), 1–16 (2018)
7. Stegman, M.: Immune Attack players perform better on a test of cellular immunology and self-confidence than their classmates who play a control video game. *Faraday discussions* 169, pp. 403–423 (2014)
8. Balerdi, F.E.: Videojuegos y educación. *Comunicar. Revista científica iberoamericana de comunicación y educación* 10, pp. 171–180 (1998)
9. CodeCombat, Acerca de, <https://codecombat.com/about>, Último acceso: 2019/03/08
10. CodinGame, Acerca de, <https://www.codingame.com/about/team>, Último acceso: 2019/03/02
11. Unity 3D, Documentation, <https://docs.unity3d.com/es/current/Manual/> Último acceso: 2019/03/08
12. MapEditor, About Tiled, <https://doc.mapeditor.org/en/stable/manual/introduction/>, Último acceso: 2019/03/02
13. Adobe, Introducción a PhotoShop <https://helpx.adobe.com/mx/photoshop/get-started.html>, Último acceso: 2019/03/02
14. Nader, J.R.: Metodología de Desarrollo de Software: MBM (Metodología Basada en Modelos). *Ingeniare* 16, pp.113–127 (2014)
15. Musil, J., Schweda, A., Winkler, D., Biffel, S.: Improving video game development: Facilitating heterogeneous team collaboration through flexible software (2010)
16. D. E. de Videojuegos. Libro Blanco del Desarrollo Español de Videojuegos 2016. Desarrollo Español de Videojuegos (2016)