

# **Videojuego de Resolución de Desafíos a través de la Programación Visual para el Fomento del Desarrollo de la Lógica de Solución de Problemas (Scrap Coder)**

Alberto Ehad García Barradas, Joel Harim Hernández Javier, Dr. Yaxkin Flores Mendoza

Escuela Superior de Cómputo I.P.N. CDMX. México.

albertogb46@gmail.com, joel.programador@gmail.com

**Resumen**—El trabajo terminal presentado consiste en el desarrollo de un videojuego de aventura y resolución de problemas en donde el jugador usará programación visual para superar desafíos, con un público objetivo de jóvenes de 12 años en adelante.

**Palabras clave** – Videojuego, Programación Visual, Lógica de Solución de Problemas, Gamificación.

## **I. INTRODUCCIÓN**

Para el siglo XXI, la UNESCO [1] considera que la resolución de problemas y la creatividad son habilidades básicas que tienen que ser desarrolladas como formación fundamental en la educación humana, y que según Henríquez y Sotomayor [2] son un elemento que emergió como un eje prioritario en el Estudio Regional Comparativo y Explicativo.

La habilidad de comprensión y resolución de problemas, según la OCDE [3], se considera dentro de lo que es el Nivel 2 en la prueba Programme for International Student Assessment (PISA por sus siglas en inglés). Dicho nivel es considerado como “... el nivel básico de conocimiento que se requiere para participar plenamente en una sociedad moderna.” [3].

En México, la prueba PISA 2018 [3] reveló que los estudiantes mexicanos, en promedio, obtuvieron un puntaje bajo en las tres áreas evaluadas: lectura, matemáticas y ciencia, comparando el nivel con el promedio mundial de la OCDE, y sólo el 1% obtuvo un desempeño en los niveles de competencias más altos, a comparación del 10% a nivel global.

La OCDE define a los estudiantes que obtuvieron el Nivel 1 en la prueba PISA como de bajo rendimiento, es decir que “pueden responder y seguir cuestiones e instrucciones sencillas, pero no pueden enfrentarse a la resolución de problemas que requieran razonamientos complejos” [3]. Así mismo la prueba PISA 2018 [3] arrojó que el 35% de los estudiantes no lograron alcanzar el nivel mínimo de competencia, es decir, están en el Nivel 1, a comparación del 13% a nivel mundial.

Los datos anteriores dejan de manifiesto que es de relevancia nacional el atender y promover el desarrollo de estas habilidades anteriormente mencionadas en la población mexicana, particularmente el análisis y resolución de problemas.

Este proyecto busca crear un videojuego, llamado Scrap Coder, que fomente en los jugadores el desarrollo de la lógica de resolución de problemas mediante acertijos que se encontrarán a lo largo del juego a través de la programación visual.

Este videojuego utilizará los supuestos teóricos de la gamificación y del aprendizaje por TICs para poder lograr su objetivo; y la programación visual, ya que es un medio propicio para introducir la programación a personas sin experiencia a través de los videojuegos, según Sáez y Cózar [4].

### **Objetivo general**

Desarrollar un videojuego 2D para la plataforma de escritorio Windows de resolución de desafíos o puzzles usando la programación visual como mecánica principal para fomentar el desarrollo de la lógica de solución de problemas en los jugadores.

### **Objetivo particular**

1. Diseñar e implementar el intérprete para la parte de programación visual.
2. Realizar el arte de los personajes, objetos, escenarios e interfaz del juego.
3. Diseñar e implementar dos niveles junto con la lógica que involucra para resolverlos.

Los productos que se esperan desarrollar son:

1. El videojuego como programa ejecutable.
2. La documentación técnica.
3. El manual de usuario.

## **II. MARCO TEÓRICO**

**Gamificación:** La Gamificación es una técnica de aprendizaje que traslada la mecánica de los juegos hacia otro ámbito, generalmente educativo y/o profesional, con el fin de conseguir mejores resultados, de acuerdo con Gaitán [5].

Hamari [6] define la gamificación como los desarrollos culturales o tecnológicos en los cuales la realidad se vuelve más lúdica y, por lo tanto, permite la acumulación de habilidades, beneficios motivacionales, creatividad, crecimiento y felicidad en general.

Adicionalmente, Zichermann y Cunningham [7] mencionan que la gamificación permite fomentar y potenciar la motivación mediante actividades recreativas, así como reforzar la conducta mediante el uso de técnicas y dinámicas propias de los juegos.

En nuestro caso, las técnicas recreativas serán la propia naturaleza de nuestro proyecto, la jugabilidad y las diversas mecánicas que se integrarán en el juego, como los puntos y las medallas por pasar los niveles, así como la historia que servirá como medio de cohesión para el resto de los elementos antes mencionados.

**Programación Visual:** Un lenguaje de programación visual (LPV), en computación, es cualquier lenguaje de programación que permite al usuario crear programas o algoritmos manipulando elementos visuales en lugar de expresarlos de forma textual, previniendo errores sintácticos típicos dado que la sintaxis de dichas declaraciones se encuentra implementada en las formas visuales del mismo lenguaje, como indican Budde et al. [8].

Repenning [9] nos indica que los principales obstáculos que los lenguajes visuales resuelven (o al menos, tratan de abordar) respecto a los lenguajes textuales, se pueden resumir en tres puntos importantes:

1. **Sintaxis:** Los LPVs utilizan íconos o bloques, formas y diagramas para construir los programas, lo cual reduce o incluso elimina los errores sintácticos.
2. **Semántica:** Visualmente se provee el significado de cada bloque o elemento que componen al programa.
3. **Pragmática:** En algunas implementaciones, se puede observar el comportamiento del programa o los estados de este conforme es ejecutado.

El videojuego integrará un lenguaje de programación visual para la programación del comportamiento del robot y el uso de sus respectivas acciones disponibles.

**Aprendizaje Basado en TICs:** Las Tecnologías de la Información y la Comunicación (TIC) tienen diferentes definiciones atribuidas a su concepción. Belloc [10] menciona que son resultado de los avances científicos y tecnológicos en la informática y telecomunicaciones, y que, dependiendo del contexto socio-tecnológico, son el conjunto de tecnologías que permiten el acceso, producción, tratamiento y comunicación de información.

Betancourt [11] define al aprendizaje basado en TIC como la utilización de las TIC como herramientas

didácticas y pedagógicas, teniendo en cuenta el factor motivacional que representan, en el proceso del aprendizaje y la enseñanza, haciendo este proceso más dinámico.

El aprendizaje basado en TIC facilita el aprendizaje a distancia, sin la presencia física de un profesor, y por su propia naturaleza ayuda a desarrollar habilidades de aprendizaje autónomo y favorece la lectura de comprensión, señala Betancourt [12].

Nuestro proyecto contará con un entorno de programación visual, el cual es considerado una TIC y ha sido evaluado en varios contextos; además de esto, se tendrá la habilidad de remendar o reparar el código generado por el jugador, permitiendo repetición y experimentación. Estas dos características permiten identificar el juego como una TIC.

### III. JUGABILIDAD

La jugabilidad se compone principalmente por las mecánicas dadas al jugador para el control de su personaje y el intérprete con el editor, así como el comportamiento que éste da al robot.

#### Jugador

1. **Movimiento:** El jugador es capaz de navegar dentro de los niveles controlando a Bel, la protagonista.
2. **Interactuar:** El jugador puede interactuar con botones, los cuales pueden estar conectados a una o múltiples puertas.
3. **Inspección:** El jugador puede gracias al modo de inspección, ver cómo los elementos en el juego se encuentran conectados, además de que los cables al pasar corrientes se ven iluminados por un shader personalizado.
4. **Programación visual:** Finalmente, el jugador puede acceder a un entorno de programación visual, en el cual es capaz de armar algoritmos para controlar el comportamiento del robot y así llegar a la solución del nivel.

#### El Robot

1. **Movimiento:** Una de las acciones más sencillas que el jugador puede hacer realizar al robot es la de moverse, esto puede lograrse haciendo uso de nodos de caminar y de girar.
2. **Interactuar:** Copper es capaz de interactuar con botones, para esto tiene que pasar por encima de ellos, además de esto es capaz de empujar cajas.
3. **Cargar y Guardar:** Una de las funciones a las cuales se tiene acceso con el robot es la de guardar algoritmos dentro del juego, así como poder cargarlos en el juego. Esto le permite al jugador guardar sus soluciones de los niveles, algoritmos propios que quiera guardar y además de esto, compartir el archivo generado a otros jugadores.
4. **Escanear:** El robot es capaz de escanear valores, los cuales se muestran en placas metálicas en el

piso de diferentes colores. Estas placas pueden mostrar números del 0 al 5. Cuando el robot escanea una placa, el valor obtenido es almacenado en una variable.

5. **Ingresar Valores:** El robot puede ingresar valores en paneles con botones, esto ofrece un método de extraer información almacenada previamente en el robot. Uno de los niveles implementados en el juego requiere del uso de uno de estos paneles, en donde el robot debe ingresar una serie de números ordenada de mayor a menor para abrir la puerta y terminal el nivel.

#### IV. PROGRAMACIÓN VISUAL E INTÉRPRETE

Para la programación visual se implementó un lenguaje visual que utiliza formas visuales en forma de nodos. Cada nodo se puede conectar de acuerdo a la categoría del nodo:

1. **Condiciones:** Indicados por formas octogonales.
2. **Instrucciones:** Conexiones de forma vertical con una muesca en la parte superior e inferior.
3. **Valores y Variables:** Tienen la forma parecida a piezas de rompecabezas.
4. **Arreglos:** Piezas de rompecabezas con varias muescas a lo largo del cuerpo del nodo.

Para los gráficos de los nodos se utilizaron varios conjuntos de *SpriteShapes* junto con varios *Sprites* simples, que componen los diversos estados para los diversos efectos gráficos utilizados en los nodos; así también permiten que los nodos cambien en tamaño de forma libre dependiendo del contenido de los nodos internos.

Así también la interfaz junto con todos los controles utilizados, son implementaciones personalizadas.

Sobre el intérprete, este componente utiliza la jerarquía creada por el mismo usuario al arrastrar y soltar nodos. Los nodos, al poder conectarse solo en lugares específicos de acuerdo a su categoría, permite que la jerarquía se pueda utilizar como un árbol sintáctico con una tokenización implícita y con un análisis léxico muy ligero, dejando la generación de código intermedio como una simple llamada a métodos ya predefinidos.

El intérprete tiene 3 fases principales:

**Analizador:** Se realiza un análisis muy breve del árbol de nodos, en el cual se verifica lo siguiente:

1. Existe un inicio.
2. Existe un fin.
3. El inicio está conectado con el fin.
4. Cada contenedor, de cada nodo, está lleno con al menos un nodo.
5. Los campos de textos tengan un input no vacío y válido.

**Ejecutor:** Si la fase anterior devolvió un resultado positivo, el ejecutor procede a la ejecución del primer nodo

y posteriormente ejecuta el comportamiento individual de cada instrucción.

Internamente utiliza una pila donde se van colocando las instrucciones que están siendo ejecutadas, y cada instrucción se comunica con el ejecutor para colocar en la pila más instrucciones que la instrucción en sí requiere que se ejecuten para que termine su ejecución.

El comportamiento individual de cada nodo, que son derivaciones de la superclase *InterpreterElement*, define la naturaleza y las reglas de la gramática del lenguaje visual, indicando la sintaxis, que partes de la instrucción se ejecuta primero, cómo se evalúan las expresiones y dónde se asignan, entre otras cosas.

Estos comportamientos individuales también son los encargados de comunicarse con la clase encargada de guardar y gestionar los valores de las variables y los arreglos, como se verá más adelante.

**RobotController:** Esta fase se ejecuta cada vez que se tiene una instrucción que requiera una acción por parte del robot. La clase *RobotController* recibe las acciones mandadas por el ejecutor e internamente utiliza una enumeración que indica las diversas acciones que se pueden realizar y selecciona el valor del enum adecuado de acuerdo a la acción recibida.

El estado del robot, en conjunto con la dirección en la que está viendo, es usado también para determinar la animación que debe de ejecutarse. Para controlar esto se hizo uso de un árbol de estados.

Finalmente, hablando sobre los tipos de datos con el que cuenta el lenguaje visual, las variables pueden representar variables numéricas con los límites de enteros de 32 bits con signo. Y los arreglos pueden contener hasta 20 elementos numéricos con los mismos límites que las variables.

#### V. RESULTADOS

Se desarrolló de manera exitosa un videojuego de resolución de problemas conformado por dos niveles divididos en múltiples escenarios. Se implementó un lenguaje de programación visual cuyo editor es una de las características principales del videojuego.

Los nodos usados son explicados e integrados progresivamente con el avance del jugador, evitando así el que ver por primera vez el editor sea abrumador para un jugador sin experiencia

El potencial y las capacidades del lenguaje implementado en este videojuego se ve reflejado en algunos algoritmos que se pueden implementar en este, como por ejemplo:

**Serie de fibonacci:** Esta serie es una sucesión infinita de números naturales que comienza con los números 0 y 1 y se pueden generar con un algoritmo iterativo.

**Ordenamiento por burbuja:** El algoritmo de burbuja permite el ordenamiento de elementos dentro de una estructura de datos de forma iterativa y con una complejidad  $O(n^2)$ .

## VI. CONCLUSIONES

En nuestro marco teórico contemplamos 3 puntos en los cuales se fundamenta el diseño del videojuego

Gamificación: se ve reflejada de manera explícita, ya que no se trata sólo de usar mecánicas propias de videojuegos en un ámbito educativo, sino que el proyecto es un videojuego en su totalidad.

Programación visual se basa en el permitir al usuario crear algoritmos manipulando elementos visuales, buscando quitar la sintaxis, semántica y pragmática fuera de las preocupaciones del usuario, haciéndolo más accesible y sencillo para personas sin experiencia en programación, la programación visual es una de las principales mecánicas del juego, enseñando al jugador desde el primer nivel lo más básico para que pueda comenzar a hacer algoritmos sencillos.

Finalmente, el Aprendizaje basado en TICs se encuentra en el juego al usarse como una herramienta didáctica, la cual permite al usuario aprender de manera autónoma, además de que es capaz de representar escenarios de la vida real funcionando como un laboratorio virtual.

Con relación al fomento de la lógica de solución de problemas en niños mexicanos, confiamos en las bases teóricas anteriores y en el trabajo de múltiples autores, los cuales nos sirvieron de guía para la inclusión y diseño de las diversas mecánicas implementadas; aunque una prueba de campo es necesaria para tener un mayor nivel de certeza.

## VII. REFERENCIAS

[1] UNESCO, E2030: Education and Skills for the 21st Century, 2017. [Online] Recuperado el 26 de marzo de 2021, de <http://www.unesco.org/new/fileadmin/MULTIMEDIA/FIELD/Santiago/pdf/Habilidades-SXXI-Buenos-Aires-Eng.pdf>

[2] Henríquez, C., Sotomayor, C., Avanzar en las habilidades básicas del siglo XXI, 2020. [Online] Recuperado el 26 de marzo de 2021, de <https://es.unesco.org/news/avanzar-habilidades-basicas-del-siglo-xxi>

[3] OCDE, Programa para la evaluación internacional de alumnos (PISA), PISA 2018 - Resultados, 2018.

[Online] Recuperado el 2 de abril de 2021 en [https://www.oecd.org/pisa/publications/PISA2018\\_CN\\_MEX\\_Spanish.pdf](https://www.oecd.org/pisa/publications/PISA2018_CN_MEX_Spanish.pdf)

[4] Sáez, J., Cózar, R., Programación visual por bloques en Educación Primaria: Aprendiendo y creando contenidos, Revista Complutense de Educación, 2015.

[5] Gaitán, V., Gamificación: el aprendizaje divertido, 2013. [Online] Recuperado el 12 de septiembre de 2021 en <https://www.educativa.com/blog-articulos/gamificacion-el-aprendizaje-divertido/>.

[6] Hamari, J., Gamification, 2019. [Online] Recuperado el 2 de abril de 2021 en <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781405165518.wbeos1321>

[7] Zichermann, G., Cunningham, C., Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps Sebastopol. California. O'Reilly Media, 2011.

[8] Budde, R., Jost, B., Ketterl, M., Leimbach, T., Graphical Programming Environments for Educational Robots: Open Roberta - Yet another One?. IEEE International Symposium on Multimedia 2014, 2014. [Online] Recuperado el 15 de septiembre de 2021 en [https://www.researchgate.net/publication/269397166\\_Graphical\\_Programming\\_Environments\\_for\\_Educational\\_Robots\\_Open\\_Roberta\\_-\\_Yet\\_Another\\_One](https://www.researchgate.net/publication/269397166_Graphical_Programming_Environments_for_Educational_Robots_Open_Roberta_-_Yet_Another_One)

[9] Repenning, A., Moving Beyond Syntax: Lessons from 20 Years of Blocks Programming in AgentSheets. University of Colorado, 2017. [Online] Recuperado el 14 de septiembre de 2021 en [http://ksiresearchorg.ipage.com/vlss/journal/VLSS2017/vlss17paper\\_10.pdf](http://ksiresearchorg.ipage.com/vlss/journal/VLSS2017/vlss17paper_10.pdf).

[10] Belloch, O., LAS TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN (T.I.C.). Universidad de Valencia. [Online] Recuperado el 4 de octubre de 2021 en <https://www.uv.es/~bellochc/pdf/pwtic1.pdf>

[11] Betancourt, J., Estrategias didácticas innovadoras: Recursos para maestros y alumnos del siglo 21. [Online] Recuperado el 4 de abril de 2021 en <https://estrategiasdidacticassite.files.wordpress.com/2017/03/libro.pdf>

[12] Meerbaum-Salant, O., Armoni, M. y Ben-Ari, M., Learning computer science concepts with scratch. Computer Science Education, 23(3), 239-264, 2013