DESARROLLO DE UN VIDEOJUEGO COMO HERRAMIENTA DE APRENDIZAJE LÓGICO Y DEL PROCESO BÁSICO DE LA PROGRAMACIÓN PARA NIÑOS DE 7 A 11 AÑOS

MICHAEL STIVEN SANDOVAL ESPINOSA

RAÚL HERNANDO LIBREROS VÉLEZ

UNIVERSIDAD DE SAN BUENAVENTURA SECCIONAL CALI

FACULTAD DE INGENIERÍAS

PROGRAMA DE INGENIERÍA MULTIMEDIA

SANTIAGO DE CALI

2014

DESARROLLO DE UN VIDEO-JUEGO COMO HERRAMIENTA DE APRENDIZAJE LÓGICO Y DEL PROCESO BÁSICO DE LA PROGRAMACIÓN PARA NIÑOS DE 7 A 11 AÑOS

MICHAEL STIVEN SANDOVAL ESPINOSA
RAÚL HERNANDO LIBREROS VÉLEZ

TESIS DE GRADO PARA OPTAR AL TÍTULO DE INGENIERO MULTIMEDIA
Asesor:
VICTOR PEÑEÑORI

UNIVERSIDAD DE SAN BUENAVENTURA CALI
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA MULTIMEDIA
SANTIAGO DE CALI
2014

Nota de Aceptación

_____

_____

_____

_____

_____

_____

_____

Firma del jurado

_____

Firma del jurado

Cali, 25 de Marzo de 2014

DEDICATORIA

A mis padres, Raúl Libreros y Rosalba Vélez,
A mi hermana Viviana Libreros, que físicamente no se encuentra conmigo, pero su energía me acompaña en todo momento.
A mi compañero de Tesis: Michael, por 5 años de constante lucha.

Raúl H. Libreros.

A mi madre Edna Espinosa y a mi padre John Sandoval
A mi abuelita Julia, que cuidó de nosotros en todos los procesos.
A Raúl, más que un compañero, un hermano.

Michael E. Sandoval.

AGRADECIMIENTOS

Agradecemos la ayuda de los profesores del Programa de Ingeniería Multimedia.
A Gabriel Montaño, que nos guio y apoyó cuando era tan solo una idea.

También a nuestro director de Tesis Víctor Manuel Peñeñory por su ayuda, dedicación porque nos ayudó con su experiencia hasta en los días festivos.

A nuestros familiares y conocidos, por su paciencia y colaboración en el proceso.
A Pablo Bejarano por asesorarnos en el desarrollo.
A Camila Reyes, que nos comprendió en situaciones difíciles y compartió sus conocimientos con nosotros.

Al Colegio Claret de la ciudad de Cali, que nos brindó apoyo en los procesos de desarrollo. En especial a Ibeth León.

TABLA DE CONTENIDOS

GLOSARIO……..x
LISTA DE FIGURAS.....xi
LISTA DE GRÁFICOS.....xii
LISTA DE ANEXOS.....xii
RESUMEN.....1
INTRODUCCIÓN.....2

CAPÍTULO 1.
LOS VIDEO-JUEGOS WEB COMO HERRAMIENTA DE APRENDIZAJE LÓGICO.....5

1.1 PLANTEAMIENTO DEL PROBLEMA.....5

1.1.1. Formulación del problema.....6
1.1.2. Preguntas directrices.....6
1.1.3. Delimitación del objeto de investigación.....7

CAPÍTULO 2
OBJETIVOS.....8

2.1 OBJETIVOS GENERALES.....8
2.2 OBJETIVOS ESPECÍFICOS.....8
2.3 JUSTIFICACIÓN.....8

2.3.1. El Videojuego Como Herramienta Para El Aprendizaje en Niños De 7 A 11 Años.....13
2.3.2. Principales Características Del Aprendizaje Basado En Videojuegos13
2.3.2.1 Contexto Neurológico.....14

CAPÍTULO 3
MARCO TEÓRICO.....16

3.1 LA EVOLUCION DE LOS VIDEOJUEGOS.....16

3.1.1. Tipos de Video Juegos.....19
3.1.1.1 Videojuegos Educativos.....22
3.1.1.2 Videojuegos Web.....24

3.1.2. Programación en la Educación Escolar.....24
3.1.3. Teoría Cognitiva de Jean Piaget.....25

3.1.2.1. Sensorio motor.....25
3.1.2.2. Pre-operacional. 25
3.1.2.3. Operaciones Concretas.....26
3.1.2.4. Operaciones Formales. 26

3.2 HERRAMIENTAS DIDÁCTICAS DE APRENDIZAJE.....27

3.2.2. Los Videojuegos En La Escuela.....27

3.3 TÉCNICAS DE APRENDIZAJE.....32
3.3.1. Tipos de Aprendizaje.....32

3.3.2. Teorías del Aprendizaje.....33

3.3.2.1 Teorías conductistas.....33
3.3.2.2. Teorías Cognitivas.....34
3.3.2.3. Teoría del procesamiento de la información..... 35

CAPÍTULO 4.
ESTADO DEL ARTE.....36

4.1. ANTECEDENTES.....36

4.1.1. Videojuegos Tipo Herramienta Educativa En Aprendizaje Lógico.....37

4.2. PLATAFORMAS QUE ENSEÑEN A PROGRAMAR.....39

4.2.1 ALICE.....39
4.2.2. SCRATCH.....42
4.2.3. PSeINT.....45

4.3 HCI.....49

4.3.1. Historia De La IPO.....50
4.3.2. La interfáz de usuario.....53
4.3.3. La interdisciplinariedad de la IPO.....54
4.3.4. Usabilidad.....58
4.3.5. El diseño centrado en el usuario.....58

CAPÍTULO 5
IDENTIFICACIÓN DE LA METODOLOGÍA DE ENSEÑANZA.....59

5.1. MÉTODOS DE APRENDIZAJE…….....59

5.1.1. Método de Aprendizaje por Proyectos.....59
5.1.2. Método de Aprendizaje Basado en Problemas.....60
5.1.3 Método del Aprendizaje Activo.....60

5.2. TEMAS APLICADOS EN LOS NIVELES DEL VIDEOJUEGO.....61
5.3. SELECCIÓN DE LA METODOLOGÍA DE ENSEÑANZA.....62

CAPÍTULO 6
DESARROLLO DEL PROYECTO.....64

6.1. SELECCIÓN DE LA METODOLOGÍA PARA EL DESARROLLO.....64
6.2. OTROS MODELOS DE DESARROLLO 71

6.2.1. RUP.....71
6.2.2. Secuencial.....75

CAPÍTULO 7
DESARROLLO DEL VIDEOJUEGO MEDIANTE EL MODELO ESCOGIDO.....77

7.1. ANÁLISIS DE REQUISITOS.....77
7.2. DISEÑO.....85
7.3. PROTOTIPO.....96
7.4. EVALUACIÓN.....103

7.4.1. Instrumentos de Evaluación.....103
7.4.2. Resultados Observación en Contexto General.....104
7.4.3 Análisis de la Observación Individual de Usuario.....106
7.4.4. Resultados Obtenidos Encuestas a Docentes.....108
7.4.5. Resultados Obtenidos en las Entrevistas a Usuarios.....113

7.1.5. IMPLEMENTACIÓN.....114

CONCLUSIONES.....136
REFERENCIAS.....138
ANEXOS.....148

GLOSARIO

TIC: Tecnologías de la Información y Comunicación

EAO: Enseñanza Asistida por Ordenador

ETR: Juego de Estrategia en Tiempo Real

FPS : First Person Shooter (Acción en primera persona)

GBL: Game Based Learning (Aprendizaje basado en juegos)

IA: Inteligencia artificial

MMORPG: Massive Multiplayer Online Role Playing Games (Juegos de rol multijugador masivos en línea)

MOD: Modified version of a Digital game (Versión modificada de un videojuego)

RAM: Random Access Memory (Memoria de acceso aleatorio)

RPG: Role Playing Game (Juego de rol)

SCORM: Sharable Content Object Reference Model (Modelo de referencia de objetos de contenido compartible)

SGA: Sistema de Gestión del Aprendizaje

TDA: Trastorno por Déficit de Atención

TDAH: Trastorno por Déficit de Atención con Hiperactividad

HCI: Human Computer Interaction (Interacción Hombre-Máquina)

RUP: Metodology Rational Unified Process (Metodología Proceso Unificado Racional)

ABP: Aprendizaje Basado en Problemas

LISTA DE FIGURAS

Figura 1. Learning to program with Alice 42
Figura 2. Interfaz gráfica de Scratch 44
Figura 3. Logo PSeInt 49
Figura 4. Ingeniería de la usabilidad y la accesibilidad. Modelo de proceso 64
Figura 5. Nivel 6 de Cat-Shmat 78
Figura 6. Modelo conceptual 86
Figura 7. Cuevas de la India 88
Figura 8. Paleta de colores para el logo y objetos, basadas en las imágenes 89
Figura 9. Proporción Áurea en boceto del logo 90
Figura 10. Logotipo final 91
Figura 11. Uso prohibido del logotipo. Ejemplo 1 91
Figura 12. Uso permitido del logotipo. Ejemplo 2 91
Figura 13. Uso prohibido del logotipo. Ejemplo 3 92
Figura 14. Uso prohibido del logotipo. Ejemplo 4 92
Figura 15. Caracteres de la fuente principal 93
Figura 16. Caracteres de la fuente principal empleada para parrafos 93
Figura 17. Estructura del personaje principal. 94
Figura 18. Items relacionados con el personaje principal y su entorno 95
Figura 19. Bocetos del personaje principal. Cavernícola 97
Figura 20. Bocetos de los botones de la interfaz 98
Figura 21. Bocetos opciones pantalla de inicio 99
Figura 22. Bocetos del nivel 1 100
Figura 23. Bocetos del nivel 2 y sus elementos 101
Figura 24. Bocetos del nivel 1 y sus elementos 102
Figura 25. Bocetos de botones del nivel 3 103
Figura 26. Ejemplo de la estructura de código de un juego sencillo con enchant.js 115
Figura 27. Código de clase de objetos de imágenes 117
Figura 28. Estructura de creación de las escenas de la pantalla principal, la selección de niveles y los tres niveles 118
Figura 29. Creación de objeto de clase ImageObject y adición de EventListener 118
Figura 30. Interfaz del nivel 1 119
Figura 31. Diagrama de flujo de las acciones del nivel 1 121
Figura 32. Diagrama de flujo de las acciones de los objetos de clase DraggableObject 122
Figura 33. Interfaz del nivel 2 123
Figura 34. Diagrama de flujo del nivel 2 125
Figura 35. Interfaz del nivel 3 126
Figura 36. Código de la reproducción de las acciones del nivel 3 127
Figura 37. Función de ejecución de las acciones con delay 127
Figura 38. Métodos de la clase Player de ejecución de las acciones 128

Figura 39. Diagrama de flujo de detección de colisiones con los obstáculos 129
Figura 40. Ejecución de la acción Forward 130
Figura 41. Ejecución de la acción Jump 131
Figura 42. Gráfico de la función parabólica $\boldsymbol{y = -x2 + 4x + 5}$ 131
Figura 43. Gráfico de la función de escalamiento de la posición en el eje Y del movimiento parabolico 132
Figura 44. Ejecución de la acción Up 133
Figura 45. Gráfico de la función de escalamiento de la posición en el eje Y del movimiento parabólico de la liana 134
Figura 46. Ejecución de la acción Bat 135

LISTA DE GRÁFICOS

Gráfico 1. Porcentaje de respuestas. Pregunta 1 109
Gráfico 2. Porcentaje de respuestas. Pregunta 2 109
Gráfico 3. Porcentaje de respuestas. Pregunta 3 110
Gráfico 4. Porcentaje de respuestas. Pregunta 4 111
Gráfico 5. Porcentaje de respuestas. Pregunta 5 111
Gráfico 6. Porcentaje de respuestas. Pregunta 6 112
Gráfico 7. Porcentaje de respuestas. Pregunta 7 113

LISTA DE ANEXOS

Anexo A. Index.html 148
Anexo B. game.js 148
Anexo C. Entrevistas a los niños 176
Anexo D. Encuestas para los docentes 180

RESUMEN

Con el fin de conocer la importancia de la tecnología educativa en la población infantil, se desarrolla la presente investigación, desde la Ingeniería Multimedia, acerca de los videojuegos como una herramienta trascendental en la educación de hoy, dado que permite desarrollar e incentivar la lógica de los niños y sus procesos de aprendizaje de una manera lúdica y vivencial, partiendo de lo anterior es fundamental tener en cuenta aspectos como la evolución del videojuego, los tipos y su uso en la web.

Cabe aclarar que otro de los objetivos propuestos busca que los niños que se encuentran entre los 7 y 11 años de edad se interesen por conocer el proceso básico de la programación a través de los videojuegos, tema de gran atracción para el infante, no obstante para lograr esto fue relevante conocer acerca de los procesos y tipos de aprendizaje de los niños, por medio de la teoría cognitiva de Piaget, autor principal para el desarrollo de la investigación.

Para dar fin a la investigación acompañada de la mayoría de áreas de la Ingeniería Multimedia y considerando el Modelo de Procesos de la Ingeniería de la Usabilidad y de la Accesibilidad, se logra crear el videojuego que promete cumplir con los objetivos anteriormente mencionados.

Palabras clave: desarrollo, videojuego educativo, navegador-web, herramienta, enseñanza-aprendizaje, lógica, Cali, Colombia, ingeniería multimedia.

INTRODUCCIÓN

Las nuevas tecnologías han traído consigo en los últimos años, la posibilidad de transformar la manera en que se llevan a cabo ciertas tareas: trabajar, experimentar entretenimiento, comunicar, enseñar, aprender; incluso, la forma de pensar. Mediante el conocimiento racional y creativo, se han encontrado soluciones a los problemas de la humanidad y a las necesidades sociales e individuales. La sociedad es consciente de lo que se utiliza a diario; sin embargo, la mayoría de las personas al escuchar el término 'tecnología' imaginan sólo productos tangibles, sin tener claro que, la parte intangible que son los programas permiten la debida manipulación de los artefactos. (Pearson, Y., 2002).

Por lo anterior, se puede evidenciar con el devenir histórico que la tecnología juega un papel importante en todos los ámbitos de la vida. El conocimiento de esto permite que la sociedad actual permita diseñar estrategias que garanticen nuestro acercamiento a la productividad. Es por esto que Colombia en 1993 decide efectuar un autoanálisis de la situación de la ciencia, la tecnología y la educación, teniendo en cuenta el avance mundial tanto en el campo político social como el de la ciencia y la tecnología. Luego logra, a partir de la ley 115 del 8 de Febrero. Artículo 23, establecer la Tecnología e Informática como un área obligatoria y primordial, formalizando una base relacionada entre la educación y la tecnología. (Llinás, R., 1994).

Desde entonces las instituciones del nivel básico y medio han intentado dotar el área con infraestructura y personal humano adecuados, para generar interés en ellos por pensar, crear y desarrollar soluciones adaptadas a las dificultades del entorno que los rodea. Sin embargo, es evidente que no se tienen en cuenta estrategias que involucran las características del aprendizaje mediante los juegos lúdicos, que aunque en principio fueron asociadas al ocio, ahora se revalidan por

permitir una enseñanza eficaz, en algunas circunstancias, superior a la del presente esquema educacional. (Gros, B., & Aguayos, J., 2004).
Por otra parte, una de las áreas cognitivas más importantes en la etapa de crecimiento de los niños es el pensamiento lógico, puesto que el mundo exige desempeñarse en procesos de razonamiento superior. Además, el debido desarrollo de dichas estructuras cognitivas concurre al éxito en las etapas educativas posteriores. (Leiva, M., 2006).

Para complementar lo anterior esta tesis es constituida por siete Capítulos, en el primero se realiza el planteamiento del problema por medio de la explicación de las dificultades que presentan los niños a la hora de utilizar las nuevas tecnologías y recalcando el valor de la programación en los más chicos.

En el segundo Capítulo se justifica el proyecto desde la necesidad de proponer métodos de enseñanza diferentes e incentivar el aprendizaje del proceso básico de la programación, para ello, se hace una mirada del videojuego como herramienta de aprendizaje en niños de 7 a 11 años, con las principales características de dicho aprendizaje y un análisis de manera general del contexto neurológico a la hora de utilizar estas herramientas.

En el tercer Capítulo se hace un repaso a la evolución de los videojuegos, se proporciona una descripción de los tipos de videojuegos, se profundiza en el tipo educativo y tipo web. También, se tienen en cuenta la programación en la educación escolar y la Teoría Cognitiva de Jean Piaget. Una mirada a las técnicas de aprendizaje para confirmar las teorías.

En el cuarto Capítulo se reseñan los videojuegos similares y las plataformas que enseñan a programar. Para la realización del Investiga la disciplina que evalúa el diseño e implementación de sistemas informáticos interactivos para uso humano.

En el quinto Capítulo se revisan los tipos de aprendizaje y se elige el más apropiado para la investigación.

En el sexto Capítulo se revisan los modelos de desarrollo y se selecciona el indicado para este videojuego.

Para finalizar, en el séptimo Capítulo se concluye con las etapas pertenecientes al desarrollo del producto en cuestión: los análisis de requisitos, el diseño, prototipado, implementación y evaluación.

CAPÍTULO 1.
LOS VIDEO-JUEGOS WEB COMO HERRAMIENTA DE APRENDIZAJE LÓGICO

1.1 PLANTEAMIENTO DEL PROBLEMA

Nuestra sociedad ha atravesado y atraviesa grandes cambios. Diariamente surgen nuevas formas de comunicación y de adquirir información, se han creado nuevas Tecnologías de la Información y Comunicación (TIC), con el fin de mejorar la calidad de vida en todos los estratos socioeconómicos y de brindar herramientas necesarias para que las personas puedan enfrentar los retos de una sociedad cada vez más globalizada y competitiva.

Muchos niños, en especial sectores de estratos 3,4,5,6, ya han tenido contacto con algún tipo de tecnología, pese a esto muchos colegios se resisten a incorporar a las TIC dentro de su currículo como recurso y medios de aprendizaje. Los computadores de la actualidad se sustentan en lógica matemática basada en un sistema binario, el cual se implementa sobre dispositivos electrónicos que permiten el desarrollo de algoritmos. Ligado a esto nace la necesidad de implementar un método para comunicarnos con los ordenadores, y así fue como surgieron los lenguajes de programación que hoy en día, han hecho posible la comunicación con el computador. (Mahoney, 1988).

Teniendo en cuenta lo anterior, el propósito del presente proyecto, busca promover en niños de 7 a 11 años de edad, su atracción por la tecnología, con el fin de incentivar su pensamiento lógico y razonamiento, dado que hoy en día la tecnología cumple un rol relevante en la educación y es aplicada en diversas áreas básicas como las matemáticas, lengua castellana, biología, geografía y lengua extranjera. (Sangesh, 2010).

Es importante mencionar que la asignatura “tecnología”, hace parte del currículo de algunos colegios y tiene como objetivo enseñarle al niño a manejar herramientas básicas computacionales, es aquí donde prevalece la problemática principal al formar al educando como usuario de un programa y no como un desarrollador, haciendo que vean la computación como una herramienta de consumo pero no como una de creación. Además no cuentan con una aplicación en línea dedicada a sus necesidades que sea utilizada tanto en su hogar como en el salón de clase. Por tal razón es conveniente evitar los modelos de enseñanza tradicionales, brindando una herramienta web para el aprendizaje lógico y del proceso básico de la programación a los niños de la era digital de hoy. (Jenkins, 2002).

1.1.1. Formulación del problema

¿Puede crearse un video-juego web para incentivar el aprendizaje del proceso básico de la programación y la lógica matemática en niños de 7 a 11 años de edad?

1.1.2. Preguntas directrices

- ¿Qué tipo de video-juego será el adecuado para desarrollar el razonamiento lógico matemático en niños de 7 a 11 años?

- ¿Cuál es el nivel de pensamiento lógico y razonamiento que se logran a través de los video-juegos?

- ¿Servirán los video-juegos para mejorar el razonamiento lógico en niños de 7 a 11 años?

1.1.3. Delimitación del objeto de investigación

• OBJETO DE ESTUDIO

Incentivar la motivación del aprendizaje en el proceso básico de la programación y la lógica.

• CAMPO DE ACCIÓN

Desarrollo de un video-juego

CAPÍTULO 2
OBJETIVOS

2.1 OBJETIVOS GENERALES

- Desarrollar un videojuego web que facilite el aprendizaje del proceso básico de la programación y la lógica matemática en niños de 7 a 11 años de edad.

2.2 OBJETIVOS ESPECÍFICOS

- Identificar un método o metodología de enseñanza en la lógica y el proceso básico de la programación para niños de edades entre 7 y 11 años.
- Desarrollar un videojuego especializado que enseñe de manera indicada el proceso básico de la programación a niños de edades entre 7 y 11 años.
- Obtener y Analizar resultados obtenidos en pruebas del videojuego final en los niños para cada característica del prototipo (diseño, interactividad, motivación, método de enseñanza y su eficacia).

2.3 JUSTIFICACIÓN

Resulta relevante conocer la importancia de la programación, sus diferentes acepciones y su contexto histórico, con el fin de comprender lo esencial de su enseñanza en niños y niñas entre los 7 y 11 años de edad.

A pesar de que la programación es vista como un tema nuevo, a mediados del siglo XIX los primeros lenguajes de programación nacieron de la idea de Charles Babagge. Residía en lo que él denominaba la maquina analítica, pero que por faltas de herramientas no pudo construirse hasta mediados del siglo XX. El primer algoritmo fue creado entre 1842 y 1843 por Ada Lovelace, la cual es considerada

como la primera programadora de la historia, pues realizó programas para aquella supuesta máquina de Babagge, en tarjetas perforadas [1].

En la actualidad, la noción de programación se encuentra muy asociada a la creación de aplicaciones informáticas y videojuegos; es el proceso por el cual una persona desarrolla un programa valiéndose de una herramienta que le permita escribir el código (el cual puede estar en uno o varios lenguajes, tales como C++, Java y Python) y de otra que sea capaz de "traducirlo" a lo que se conoce como lenguaje de máquina, el cual puede ser entendido por un microprocesador [1]. Programar se refiere en este caso a la elaboración de programas para la resolución de problemas mediante ordenadores. En un plano más técnico, la programación se realiza mediante el uso de algoritmos, que son secuencias finitas, ordenadas y no ambiguas de instrucciones que deben seguirse para resolver un problema. Algunas de ellas pueden agruparse y recibir un nombre para poder ser invocadas con facilidad, tantas veces como sea necesario. Del mismo modo que los seres humanos necesitan respirar constantemente, una aplicación informática necesita conocer la posición del puntero del ratón a cada momento, así como su actividad (si se ha hecho clic y con qué botón, si se ha soltado o si se mantiene presionado, etcétera).[2]

Atendiendo a lo anterior la programación está en todo, puesto que los ordenadores intervienen en casi la totalidad de las actividades que realiza el ser humano, y es lo que hace funcionar a los ordenadores. Por esta razón es importante aprender a programar, porque es el presente y el futuro de la

---

[1] [Web], Definición de Programación. [Consultado el 13 de Abril de 2013]. Disponible en:

http://definicion.de/programacion/

[2] CODE.ORG [Video]. Starring; Bill Gates, Mark Zuckerberg, will.i.am, Chris Bosh, Jack Dorsey, Tony Hsie, Drew Houston, Gabe Newell, Ruchi Sanghvi, Elena Silenok, Vanessa Hurst, and Hadi Partovi: *What most schools don't teach.*Realización: Lesley Chilcott, Hadh and Ali Partovi. USA: Org.org, Febrero de 2013. [En línea]. [Consultado el 15 de Marzo de 2013]. Disponible en:
https://www.youtube.com/watch?feature=player_embedded&v=nKIu9yen5nc

humanidad. En un mundo donde las nuevas tecnologías y la información son cada vez más omnipresentes y aumentan su importancia en nuestras vidas diarias, es curioso que en el sistema educativo de muchos países no se enseñe a programar computadoras, como mínimo, una enseñanza básica. La mayoría de los trabajos de hoy en día están relacionados con el sector tecnológico, lo que sería importante preparar a los niños desde la más temprana edad para lo que el mundo necesita.[3]

Según algunos de los personajes más influyentes en este sector(Bill Gates, Steve Jobs, Marck Zuckberg, entre otros) hacen énfasis que aprender a programar no es nada diferente a aprender a tocar un instrumento o practicar un deporte. Para ellos, al empezar puede ser tedioso, pero se vuelve fácil al dedicar tiempo y constancia. La programación facilita el aprendizaje de los niños en conceptos informáticos como condicionales y eventos, además de ganar experiencia en conceptos matemáticos como sistemas coordenados, variables y números al azar.[4]

Aprender el concepto básico de la programación acerca a los niños a la resolución de problemas, más allá del manejo de algoritmos como la mayoría de la sociedad cree. Allí se combinan muchas de las habilidades de aprendizaje del siglo XXI, la cuales serán críticas para el éxito en el futuro: pensamiento creativo, claridad en la comunicación, análisis sistemático, colaboración efectiva, diseño reiterativo y aprendizaje continuo**.** Necesariamente no se van a convertir en programadores de alto nivel, pero sí les garantiza a pensar con un orden sistemático. (Squire, 2013)

---

[3] JOLIE, O'Dell, Why your 8-year-old should be coding. Venturebeat Magazine Online. [En línea].[Consultado el 13 de Abril de 2013]. Disponible en: http://venturebeat.com/2013/04/12/why-your-8-year-old-should-be-coding/

[4] Disponible en: http://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code.html

La propuesta desde la Ingeniería Multimedia en Colombia, planteada por los autores, tiene también la necesidad de mejorar la percepción de los videojuegos por parte de los niños al mostrarlos como objetos de cambio en la sociedad, asimismo, podrán reflexionar y realizar una auto capacitación en la creación de dichos videojuegos, mezclando el área educativa y social.

La importancia de este trabajo de investigación radica en que representa un antecedente a futuras investigaciones que puedan elaborarse al respecto. Desde el punto de vista tecnológico, educativo y social, los resultados de esta investigación representan un material didáctico importante. Así como también representa un material que ayuda a fomentar el desarrollo de este tipo de videojuegos. Es necesario enfatizar el aprendizaje creativo, ya que el alumno, al interactuar libremente con el objeto que desea conocer, vuelve propio el aprendizaje, se vuelve algo significativo para él.

La principal condición para formarse con principios y teorías es empleando el aprendizaje experimental, el cual se basa, como su nombre lo indica, en analizar las experiencias propias de cada individuo (D. W. Johnson & Johnson, 1975). El aprendizaje se da en un ciclo que parte, en primer lugar, de una experiencia personal concreta. Con base en ello se observa, examina y reflexiona esta experiencia para formular conceptos abstractos, reglas, principios, etcétera, lo cual lleva al alumno a crear una teoría personal que tiene que probarse en nuevas situaciones, lo cual nos lleva otra vez al principio de este ciclo.[5]

Esta metodología resulta muy efectiva para crear aprendizaje significativo y es compatible con el método científico hipotético deductivo. Esto lo demuestran los estudios sobre cómo las personas pueden dominar mejor ciertas habilidades: los

[5] Disponible en: http://www.academia.edu/1365883/Videojuegos_y_Educacion._Explorando_aprendizajes_entre_adolescentes

alumnos recuerdan el 10% de lo que leen, 20% de lo que escuchan, 30% si están viendo imágenes mientras se les explica, 50% si ven a alguien haciendo algo mientras lo explica, pero casi el 90% cuando ellos mismos realizan una simulación (Federation of American Scientists, 2006, p. 15).

Otro mecanismo que es indispensable a la hora de enseñar es la diversión. Lo último que se quiere es que los estudiantes se dispersen porque dejan de absorber conocimientos. Si se obligan a tareas repetitivas no reciben suficientes estímulos. Centran su atención de nuevo si los estímulos cambian (Gardner, 2004, p. 44). El juego es algo que cambia radicalmente los estímulos que se presentan a los alumnos y los están cambiando cada instante. El juego quizás es el elemento de desarrollo humano más importante, ya que aprendemos a vivir del juego, y tristemente el juego se considera únicamente como un fenómeno de la niñez que no aporta ningún valor al mundo adulto (Next Generation Forum, 1999, p. 14). El juego permite que las experiencias nuevas y antiguas se confabulen sin problema. Hay que fomentar que los alumnos (o cualquier persona) vayan más allá de los límites cognitivos que ellos mismos se han impuesto por medio de una experiencia vivencial que adquieren por medio de un elemento lúdico [5].

Diversos autores se encuentran de acuerdo con iniciar a programar desde la Educación Básica. Como ejemplo está Gary Stager, reconocido educador y consultor, que lleva más de veinte años en busca de apoyar y concientizar a los docentes, frente a su rol en esta era del computador. Él expresa que "la habilidad de visualizar caminos de razonamiento divergentes, anticipar errores y evaluar rápidamente escenarios mentales, es resultado directo de las clases de programación"; y agrega, "este don me sirve hoy en mi vida cotidiana cuando tengo que navegar un sistema de correo de voz o sacar mi carro de un estacionamiento". (Stager, 2003).

2.3.1. El Videojuego Como Herramienta Para El Aprendizaje en Niños De 7 A 11 Años

El juego es una actividad orientada a la diversión de los participantes, y que en ocasiones se utiliza como herramienta en el proceso de aprendizaje.

Al tratar este epígrafe nos basamos en que el uso de los juegos como herramientas de aprendizaje no es nuevo, ya que en Mesopotamia alrededor del 3500 a. C. se utilizaba un juego para enseñar factores económicos a niños de 11 años.

Begoña Gros (2009) manifiesta que el juego hace parte del hombre, y tiene diferentes acepciones desde su progreso. Así, a finales del siglo XIX, el juego estaba directamente ligado al entretenimiento, pero con la Escuela Nueva, diversos autores, como Piaget y Vygotsky, por mencionar algunos, defendieron su importancia como metodología de enseñanza, herramienta para aprender y concebir estrategias descubriendo y fortaleciendo habilidades.

2.3.2. Principales Características Del Aprendizaje Basado En Videojuegos

Como pilar principal los videojuegos tiene esa chispa que envuelve a los jugadores. Ellos incluyen la más completa integración de estímulos en todos los sentidos, y es por esto que terminan siendo adictivos. Puede ser tan interesante de lograr que sus jugadore entren en estado de flujo (Csíkszentmihályi,1996), donde olviden su entorno y centrarse sólo en la tarea que realiza. Por aquél estado, los jugadores exploran todas sus habilidades para alcanzar sus objetivos.

Algunos de los factores que detienen a los jugadores son la jugabilidad, la velocidad, los gráficos, el tipo de juego o su guión [6].

No obstante, el comportamiento de los usuarios cuando juegan depende de las propias aspiraciones. Así pues, la importancia cuando se obtiene la recompensa puede ser mayor o menor, cada usuario. En múltiples situaciones se ven los jugadores entretenidos con un guión línea o en situaciones de alto proceso cognitivo donde saquen a flote sus habilidades estratégicas. La cultura y el género son algunos de los factores que influyen en la motivación a la hora de jugar [7].

2.3.2.1 Contexto Neurológico

Las 5 características de los videojuegos son:

Conflicto, Control, Cierre, Artificio, La Competencia. Los juegos en sí, la mayoría, involucran el aprendizaje activo, la interacción entre personas, incentiva el trabajo cooperativo. Juegos de aprendizaje básicos proporcionan versatilidad para más de un estilo de aprendizaje, y también pueden afectar las habilidades cognitivas y psicomotoras. Además, mientras que el aprendizaje a través del juego puede ser muy eficaz, pueden convertirse en una distracción, haciendo que se centran demasiado en el juego y no en el aprendizaje.[8]

Otras de las características importantes de los videojuegos es que permiten que se incremente la motivación en áreas como las matemáticas y ciencias. También sirven para entrenamiento en el desarrollo del pensamiento reflexivo, para mejorar las habilidades de pilotos de avión, reducir el porcentaje de errores

---

[6] Educar en clave. Disponible en: http://xa.yimg.com/kq/groups/19326480/1493324637/name/Libro

[7] *http//www.yoyogames.com*

http://www.experientia.com/playful/creating-the-user-experience-of-aned

[8] http://centrodeartigos.com/articulos-noticias-consejos/article_130356.html

en el pensamiento racional, conseguir un mayor tiempo de reacción ante situaciones de contingencia, simular situaciones reales para enfrentamientos, y mucho más [9].

[9] http://campus.usal.es/~teoriaeducacion/rev_numero_02/n2_art_etxeberria.htm

CAPÍTULO 3
MARCO TEÓRICO

3.1 LA EVOLUCIÓN DE LOS VIDEOJUEGOS

El juego siempre ha sido un elemento de entretenimiento, como lo mencionamos anteriormente, y está entre las herramienta más utilizadas en los tiempos de ocio de las personas, pese a esto debe de expresarse que se considera indispensable para las primeras etapas del niño por su gran capacidad de estimulación, lo que hace que se desarrolle con más facilidad, el aprendizaje de las cosas. Por lo anterior, algunos pedagogos justificaron la educación basada en juegos, por la ya mencionada motivación de parte de los niños hacia ellos.

En la actualidad, como expresa (Gros,1998), no debemos olvidar que los juegos de mesa también han tenido su evolución con respecto al tiempo que ha pasado, así encontramos juegos variados, en los que el usuario es el protagonista, poniendo de su parte el ingenio y conocimiento, para que de alguna forma puedan derrotar el rival o ayudar a su compañero. Los videojuegos se pueden considerar diferentes puesto que presentan nuevas tecnologías, en donde se interactúa con los elementos de un programa.[10]

El término Video Games (videojuegos) aparece por primera vez como título en un número de Reader´s Guide to Periodicals en el período marzo de 1973 a febrero de 1974. Desde 1970 ya existían publicaciones de artículos sobre los videojuegos con los títulos “Electronic Games” y “Computer Graphics”. Actualmente las publicaciones hechas aparecen con el nombre de Video Games, Videogames, Computer Games (Juegos de computadoras) o Electronic Games (juegos electrónicos), aunque el término “juego electrónico” no es muy especifico

[10] http://www.ecured.cu/index.php/Videojuego

debido a que incluye cualquier juego que tenga un componente electrónico y aunque no sea propiamente lo que la mayoría de la gente denomina videojuego, dicho término puede referirse a un videojuego y algunas veces aparece en las definiciones de los mismos. [11]

La historia de los videojuegos está dividida en cuatro períodos de aproximadamente diez años cada uno[12]

• **Primer periodo (1965-1975): Los hijos del Atari.**

Nacen las empresas creadoras de videojuegos y además las mentes maestras. Nolan Bushnell lanzó su primer proyecto "Computer Space", juego que consistía en simular una batalla entre una nave espacial y un platillo volador, para colmo, el videojuego fue un fracaso económicamente porque solo vendió 2000 ejemplares. Luego, Nolan Bushnell crea su propia empresa llamada "Atari y con la que crea "Pong", que consiste en un juego de ping-pong basado en el principio elemental de acción y reacción. [13]

• **Segundo Periodo (1975-1985): La batalla por el mercado doméstico.**

En 1976 Bushnell vende Atari a Warner. En este periodo se termina la era de los inventores geniales y excéntricos, para darles paso a los administradores y especialistas en ventas. Para este año ya existían 20 empresas dedicadas a la fabricación de videojuegos domésticos y es así como un ingeniero japonés crea

---

[11] Jugar Videojuegos tiene grandes beneficios: Especialistas. Monterey, 2012. [Consultado el 15 Marzo de 2013].
http://www.milenio.com/cdb/doc/noticias2011/215cf5be355aeb0b371d1f895d2247fc

[12] *Historia de los videojuegos*. http://www.historia-videojuegos.galeon.com.

[13] http://www.tizimin.uady.mx/tesis/tesis_final_jose.pdf

“Pac-Man”, dicha creación forma parte importante en la evolución del videojuego. Otro avance importante es el hecho de que el protagonista del juego se vuelve una figura simpática con el cual el jugador se puede identificar [13].

Se desarrollan videojuegos domésticos coincidiendo con la aparición de las computadoras personales. En este periodo las empresas dudan en desarrollar videojuegos a partir de una computadora o sacar al mercado una consola específica[13].

• Tercer Periodo (1985-1995): La era de los gigantes Nintendo y Sega.

Tras un período de recesión que duró tres años llega Nintendo y sega que permite que la venta de videojuegos crezca y alcance nuevas cimas. Miyamoto desarrolla para Nintendo el producto Mario Bros, un juego cuya principal característica era el movimiento complejo de apariencia natural. Las dos características principales fueron la concentración de empresas y la especialización del hardware. En este tiempo solo existían cuatro clasificaciones de juego: acción/aventura, acción/arcade, simulación y deportes. En 1985 Nintendo saca al mercado su primer hardware (consola de videojuegos) [13].

• Cuarto Período: En 1998 llegan el Multimedia Interactivo y los Juegos de Red.

Se aumentan la capacidad de memoria de dispositivos y la velocidad de procesamiento con el objetivo de tener una mejor calidad en el videojuego (sonido más real, guiones más complejos y la aceleración de procesamiento de imágenes). Los videojuegos dan paso al multimedia interactivo. Se desarrolla el software de visualización (middleware) que permite a los videojuegos presentar imágenes en 3D con apariencia casi real.

En el 2000 salen al mercado de forma sucesiva tres grandes consolas capaces de procesar imágenes 3D: GameCube de Nintendo, Playstation II de Sony y el Xbox de Microsoft el cual se trata de una computadora que permite conectarse en red para jugar en línea. Sega se dedica únicamente a desarrollar videojuegos, se empiezan a comercializar los videojuegos en DVD o CD.[14]

3.1.1. Tipos de Video Juegos

La evolución de los videojuegos ha sido de forma acelerada y se han ido creando nuevos tipos o clases de videojuegos, desde el ya mencionado Pong, que se podría decir que es un juego deportivo y de habilidad, hasta los nuevos juegos de estrategia o aventura.[15]

Los videojuegos se pueden clasificar de la siguiente forma:

- **Aventuras**

En los videojuegos de aventura se desarrolla una historia en la cual el jugador tiene que ir encontrando pistas con las que descubrirá misterios y avanzará en la historia. El videojuego depende muchas veces de pruebas de inteligencia o resolución de puzzles para avanzar. En este tipo de videojuegos los guiones y diálogos deben estar muy bien logrados para que la historia mantenga la atención del usuario. Ejemplos de este tipo son: Broken Sword, The Longest Journey, Monkey Island, entre otros [13].

---

[14] *Historia de los videojuegos*. http://www.historia-videojuegos.galeon.com.

[15] http://www.tizimin.uady.mx/tesis/tesis_final_jose.pdf

- **Arcade**

Los videojuegos arcade fueron los primeros que aparecieron en el mercado, respondiendo a una estructura fundamentada en actividades de mucha destreza que permiten al usuario recorrer distintas pantallas en diferentes niveles. La rapidez en este tipo de videojuegos es el elemento más importante, incluso más que la propia estrategia del juego. En un principio estos videojuegos arcade se desarrollaban en un entorno de 2 dimensiones, siendo los típicos juegos de plataformas donde la finalidad del juego consiste en atravesar pantallas. Actualmente se desarrollan en escenarios de 3 dimensiones donde el jugador debe realizar misiones, alcanzar objetivos, entre otros. Los videojuegos de arcade engloban a los de acción, que desgraciadamente suelen ser los mas violentos y sangrientos, lo cual promueve conductas inapropiadas. Por ejemplo: Duke Nukem, Rayman, Prince of Persia, entre otros [13].

- **Deportivos**

Recrean diversos deportes, tomando como patrón el desenvolvimiento del juego en su escenario real. Para jugar en ellos se requiere habilidad, rapidez y precisión. Cuantos más jugadores participen, más divertido es el juego. Una de las facetas más importantes de estos videojuegos es la jugabilidad (término empleado en el área de videojuegos para indicar el nivel de dificultad o sencillez que representa jugar). El usuario debe tener control total sobre el jugador o jugadores que maneje. Además de la jugabilidad, se han mejorado mucho los gráficos, digitalizando jugadores y recreándolos de una manera muy real. Por ejemplo: FIFA 2004, Virtual Tennis, NBA 2005, entre otros [13].

- **Estrategia**

Los videojuegos estratégicos se llaman así por su arte al coordinar acciones y de actuar para conseguir una finalidad concreta. Estos videojuegos ofrecen al usuario la posibilidad de aumentar su capacidad de reflexión, para así conseguir lo más rápidamente posible y con menor esfuerzo el objetivo propuesto. La mayoría de los videojuegos de estrategia permiten manejar más de un personaje hasta tropas de decenas de soldados. Por esto, al igual que los videojuegos de aventura, hasta hace muy poco se utilizaba entorno de dos dimensiones (2D), para que estos juegos no requirieran tener un ordenador o consola potente para jugar. Ejemplos: Ground Control, Command & Conquer, StrongHold, entre otros[13].

- **Rol**

En este tipo de videojuegos el jugador maneja un personaje, el cual se ha podido crear con cualidades concretas, que va evolucionando durante el juego según las decisiones o caminos que tome el usuario. Conllevan muchas horas de juego, al igual que los de estrategia [13].

Suelen ser videojuegos en los que el objetivo no es único, sino que hay varios objetivos que se entrelazan. Dentro de los videojuegos de rol se encuentran los videojuegos online o mundos persistentes. En este mundo se puede interactuar con otros personajes que son otros usuarios que están también conectados a este mundo. Al igual que los videojuegos de rol individuales, los personajes van evolucionando. Ejemplos: Diablo, Baldur´s Gate, Neverwinter Nights, entre otros [13].

- **Simulación**

Los videojuegos de simulación sumergen al usuario en un mundo donde se simula algún tipo de acción como por ejemplo pilotar un avión, conducir un coche de carreras, etc. Cada vez se intenta dar más realismo a este tipo de juegos, pero

eso conlleva a que los usuarios menos veteranos necesiten más tiempo de aprendizaje por la mayor complejidad de manejo de este tipo de videojuegos. Ejemplos: Flight Simulator 2004, Xpand Rally, entre otros [13.]

3.1.1.1 Videojuegos Educativos

Muchos videojuegos favorecen el desarrollo de determinadas destrezas que ayudan al desarrollo intelectual de la persona. Es decir, que la inteligencia, contrariamente a lo que se ha podido afirmar en alguna ocasión, no parece deteriorarse al utilizar este tipo de entretenimiento digital. Más bien al contrario, parece que favorecen el desarrollo de determinados aspectos de ella, en especial los de carácter espacial. Incluso hay quienes afirman que los jugadores suelen ser individuos más inteligentes, aunque no parece existir ninguna relación entre ambos aspectos. En todo caso, se apunta que los individuos con mayor inteligencia se verán más atraídos por este tipo de entretenimiento. (Huizinga, 1996)[16].

Desde el punto de vista educativo los videojuegos, permiten una enseñanza eficaz, en bastantes ocasiones mucho mejor que el actual sistema escolar. Algunos de los factores que fomentan la motivación son intrínsecos a los videojuegos[17].

Podemos mencionar los siguientes:

- Carácter lúdico de los aprendizajes.
- Dificultad creciente y progresiva de las habilidades, pero adaptada al ritmo de cada uno, posibilidad de repetir y corregir los errores.

---

[16] http://issuu.com/generacionesinteractivas/docs/educar_hijos_interactivos_chile/68

[17] http://www.irabia.org/departamentos/nntt/proyectos/futura/futura06/Analisis_educativo.pdf

- Recompensa inmediata, que además responde a un plan predeterminado y conocido.
- Reconocimiento social de los logros adquiridos, inscripción personalizada de las puntuaciones alcanzadas o los niveles superados.
- Estimulación simultánea a múltiples niveles: visual, auditivo, etcétera.
- Identificación con héroes o personajes que fomentan la imitación.

Como bien sabemos, habitualmente, ni la escuela ni la familia, agentes educativos por excelencia, proponen aprendizajes que conlleven estas características, o al menos no de una manera tan perseverante e intensa.
Greenfield (2000) investigó el aprendizaje producido en niños y niñas de 12 a 16 años que utilizaron videojuegos de aventuras. Las principales conclusiones obtenidas fueron que[16]:

- Aumentaban las estrategias de lectura visual de imágenes y de lectura del espacio tridimensional.
- Ayudaban a trabajar el aprendizaje por observación y la verificación de hipótesis.
- Facilitaban la comprensión de simulaciones científicas.
- Incrementaban las estrategias para recibir y procesar información recibida de varias fuentes simultáneamente (procesamiento en paralelo)

McFarlane (2002) reconoce que los videojuegos allanan la adquisición y el desarrollo de ciertas estrategias fundamentales para el aprendizaje: la resolución de problemas, el aprendizaje de secuencias, el razonamiento deductivo y la memorización. También, simplifican la realización de trabajos en grupo de tipo cooperativo o en colaboración y el aprendizaje basado en la resolución de tareas[16].

3.1.1.2 Videojuegos Web

Miles de personas de todas las edades, niveles culturales o socioeconómicos se encontrarán en este preciso instante jugando en Red; algunos separados sólo por las divisiones que delimitan los espacios de cada equipo, en los locales destinados a esta nueva modalidad lúdica, o alejados miles de kilómetros, incluso separados por los más diversos mares u océanos, desde la habitación de su hogar. No en vano, cada vez existen más servidores de Internet dedicados exclusivamente a satisfacer esta modalidad del ocio moderno.

En el fondo, se trata de una innovación en el ámbito de los videojuegos de "toda la vida", en los que el objetivo era enfrentarse y derrotar a la máquina. Ahora, los jugadores, solos o formando equipos, se enfrentan entre sí, compitiendo con seres de su inteligencia, habilidad o suerte (Bringuier, 1977).

3.1.2. Programación en la Educación Escolar

Entonces, se puede evidenciar que la programación hace parte de la vida cotidiana y presenta diferentes beneficios en asignaturas escolares como la matemática y la computación, al comprometer al estudiante a solucionar problemas de manera experimental y repetitiva. Así mismo, la programación es reconocida como una estrategia válida para desarrollar el enfoque educativo STE (Ciencia, Tecnología, Ingeniería y Matemáticas), el cual tiene como fin capacitar a docentes y estudiantes en estas áreas, para obtener una adaptación en los tiempos actuales, haciendo referencia a la globalización e innovación.

Cabe mencionar que hay quienes sustentan que hoy gran parte de los artefactos que se involucran en la cotidianidad están programados, desde los semáforos, hasta los ascensores que facilitan la movilidad en edificaciones. Actualmente, los computadores son fundamentales en la sociedad

contemporánea, por esta razón, no basta con saber su uso básico, sino que resulta necesario obtener conocimientos de programación, para entender cómo funcionan las máquinas y los diferentes sistemas que hacen parte del diario vivir.

El Consorcio de Habilidades Indispensables para el Siglo XXI, identifica que otras de las habilidades que serán indispensables para el triunfo de los educandos en un futuro son: “pensar creativamente, comunicar claramente, analizar sistemáticamente, colaborar efectivamente, diseñar iterativamente y aprender continuamente”. Además de desarrollar las capacidades intelectuales de orden superior, los cuales son “análisis, síntesis, conceptualización, manejo de información, pensamiento sistémico, investigación y metacognición”. (Stager, Gary s.a)

3.1.3. Teoría Cognitiva de Jean Piaget

Teniendo en cuenta que la investigación está siendo dirigida a niños, conviene examinar sus comportamientos y habilidades cognitivas según su edad y/o etapa a la que pertenecen. Para esto, se llevará a cabo una indagación de Jean Piaget, psicólogo suizo e investigador de diversos estudios basados en la psicología infantil, además, autor de la “Teoría Cognitiva”.(Piaget, J. 1952)

Para continuar, se debe aclarar que dicha teoría, se encuentra dividida en cuatro estadios, los cuales definen el desarrollo cognitivo de los niños y adolescentes.

Sensoriomotor: Estadio al cual pertenecen niños entre los 0 a los 2 años de edad. Se desarrollan diversas habilidades motrices y mentales.

Pre-operacional. Abarca el período que va desde los 2 hasta los 6 años, se reconoce como la etapa del pensamiento y el lenguaje.

Operaciones Concretas. Comprende edades entre los 7 y 11 años. Es aquí donde los procesos de razonamiento se tornan lógicos y son aplicados a la vida cotidiana, consiguiendo resolver problemas reales o concretos.

Operaciones Formales. Va desde los 12 años en adelante, en este período, el adolescente razona de forma hipotética y sin necesidad de pruebas materiales.

Ya conociendo los cuatro estadios, formulados por Jean Piaget, es debido mencionar que los niños y niñas pertenecientes a la investigación, se encuentran entre los 7 y 11 años de edad, por tal motivo, se realizará una investigación más explícita del tercer estadio: "Etapa de operaciones concretas". Además, comprende una relación más directa, frente al proyecto a realizar. (Brienguier, J. 1977)

Entonces, continuando con lo anterior se procederá a la explicación de la que según Piaget es la tercera fase en la vida de un niño y a la que denominó fase de **Operaciones Concretas**, que comprende desde los siete hasta los doce años de edad.

En esta fase el niño debe ser capaz de manejar correctamente la información concreta y debe ser capaz de llevar a cabo acciones interiorizadas. El niño ve el proceso completo, es capaz de plantear la reversibilidad, es decir, justificar una respuesta dada al observar que, una transformación acontecida en un momento determinado, vuelve a su origen. (Evans, R. 1973)

3.2 HERRAMIENTAS DIDÁCTICAS DE APRENDIZAJE

Implementar el juego para el niño es muy importante, ya que es una herramienta indispensable para su desarrollo, tanto físico, cognitivo, psicológico y social.

"El niño que no juega es un niño al que le pasa algo".

La investigación de la relación que existe entre juego, desarrollo infantil y aprendizaje ha generado:

- Un nuevo concepto de infancia. "Desarrollo y aprendizaje a través de una vivencia feliz y placentera".
- Un nuevo concepto de la educación. "Metodología educativa centrada en el juego como base de los aprendizajes". [18]

3.2.2. Los Videojuegos En La Escuela

Además de los videojuegos, recientemente la industria multimedia ha empezado a diseñar los denominados juegos educativos. Donde su diferencia radica en la intencionalidad: en los videojuegos la intención es entretener, divertir, pero los juegos educativos se diseñan para que el niño aprenda a través de una actividad aparentemente lúdica, la conjunción puede suponer la aceptación y comprensión de términos de difícil asimilación para determinadas edades [19].

Es importante tener en cuenta que los videojuegos representan micro - estados educativos, ya que cada uno sigue unas reglas y unos principios particulares. Representan un proceso que Marx llamó la "creatividad del

---

[18] UNIVERSIDAD ESPECIALIZADA DE LAS AMERICA. Modulo II. Las tecnologías Analógicas en la docencia Superior. Lorena Montenegro

[19] http://www.dialogosfelafacs.net/wp-content/uploads/2011/10/80.pdf

capitalismo", que conduce a diseños cada vez mejores para el buen aprendizaje y, de hecho, al buen aprendizaje de cosas desafiantes. Se trata de encontrar "la teoría del aprendizaje humano incorporada a los buenos videojuegos" (Gee, 2004) desde el campo de la teoría cognitiva, la que se encarga de estudiar cómo aprenden más y mejor los seres humanos [20].

El "Informe sobre el uso de juegos en educación" (Freitas 2007) sostiene que para que exista aprendizaje, los juegos han de tener relación con los resultados del aprendizaje, y al mismo tiempo han de ser relevantes para contextos de práctica del mundo real. El hecho de jugar se entiende como un momento lúdico, sin valor pedagógico, hecho que dificulta la implantación del aprendizaje basado en los videojuegos debido a su concepción social meramente basada en el entretenimiento y no analizando el tipo de juego en concreto, lo que puede aportar una u otra herramienta. Hay que valorar que los videojuegos pueden ser potentes herramientas de: simulaciones reales, gestión de recursos, conectividad, red de acceso a fuentes de información, desarrollo de actividades multitarea… En definitiva, "reduce a mecanismos esenciales la descripción de realidades muy complejas" (Gros, 2008)[21].

La mayoría de los juegos educativos, se centran en proponer al niño la realización de una serie de actividades que coinciden en la mayoría de los productos: laberintos, juegos de emparejamiento, de dibujo y actividades de reconocimiento de letras, colores, números, formas y notas musicales. En cualquier caso, no ofrecen grandes diferencias respecto a programas puramente educativos y, en cierta forma, falsean con su presentación la intención ya que no es jugar sino aprender de forma bastante rutinaria con un entorno, eso sí, más atractivo. No hay que negar la utilidad de este tipo de productos, pero hay que

[20] http://ticeducac1.wikispaces.com/Los+videojuegos+en+educaci%C3%B3n

[21] http://dialnet.unirioja.es/descarga/articulo/3719704.pdf

insistir en que se pueden trabajar numerosos aspectos curriculares a nivel de contenidos, procedimientos y valores utilizando videojuegos.

El uso de los videojuegos como un material informático más en la escuela, supone una aproximación por parte del profesorado, que hasta el momento no ha visto la potencialidad de este producto o simplemente, lo considera excesivamente complejo. En este sentido, el profesor o profesora que utiliza videojuegos debe replantearse su propio papel dentro del aula ya que, en muchos casos, se le escapará el control del videojuego en sí mismo ya que no es extraño que los estudiantes estén mucho más capacitados que los profesores en el dominio técnico del programa. Por ello, su incidencia no está en el juego sino en su uso, su análisis y utilización para adquirir unos objetivos educativos concretos[22].

Si planteamos introducir los videojuegos en el ámbito educativo, hemos de añadir la influencia del entorno de uso. El videojuego introducido en la escuela se trasforma, ya no es un programa para jugar sino que el juego tiene una intencionalidad educativa. Utilizaremos el juego para desarrollar unas determinadas habilidades o procedimientos, para motivar a los alumnos y/o para enseñar un contenido curricular específico [23].

Ventajas de los videojuegos:

Permiten aprender diferentes tipos de habilidades y estrategias.

---

[22] http://edutec.rediris.es/Revelec2/Revelec12/gros.html

[23] https://www.academia.edu/1967866/LOS_JUEGOS_ELECTRONICOS_COMO_RECURSO_DIDACTICO_PARA_EL_APRENDIZAJE_EN_FORMACION_CIVICA_Y_ETICA_EN_CUARTO_GRADO_DE_EDUCACION_PRIMARIA

Ayudan a dinamizar las relaciones entre los niños del grupo, no sólo desde el punto de vista de la socialización sino también en la propia dinámica de aprendizaje.

Permiten introducir el análisis de valores y conductas a partir de la reflexión de los contenidos de los propios juegos. No todos los videojuegos son válidos como herramientas didácticas, deben de contener una serie de elementos que faciliten y mejoren las técnicas curriculares utilizadas normalmente para desarrollar los contenidos que el docente quiere impartir.

Empleando videojuegos se incrustan competencias para el uso de programas y entornos multimedia, se moviliza con el paso de la información y se administran afluencia de recursos mientras se despliegan estrategias de organización, diseño y planificación. A través de ellos los niños y niñas se introducen en todo un nuevo sistema de símbolos complejos. Hay un nuevo tipo de lenguaje audiovisual en este campo que ellos, gracias a su corta edad, y por ello con una capacidad muy alta de comprender nuevos lenguajes, son capaces de adquirir de una manera rápida y dinámica (Gros, 2008). Forman parte de un nuevo lenguaje, y que desarrollan una serie de competencias, divididas en cuatro grandes grupos:

Competencias instrumentales para gestionar entornos multimedia Son programas de diseño avanzado en entornos multimedia en el que se aglutinan elementos de tipo:

Textual, verbal, icónico, procesamiento de tareas, retroalimentación, comunicación, interacción, intencionalidad, planificación, control de situaciones. Aprendemos a leer, a escribir, a hablar, a escuchar y a ver en función de los avances tecnológicos que se producen en el campo de la comunicación [24].

[24] http://ticeducac1.wikispaces.com/Los+videojuegos+en+educaci%C3%B3n

- Competencias en la gestión de recursos

Se trata de cómo interpretan y utilizan la información que les proporciona el juego sobre los recursos disponibles de cada civilización y como los gestionan conforme a las reglas establecidas del videojuego, para ello deben dominar estas cuatro competencias básicas [23]:

- Gestión de la información
- Gestión de los recursos digitales
- Gestión y desarrollo de estrategias de diseño y planificación
- Gestión de la información y de las variables del juego

- Competencias para la comunicación:

La utilización del videojuego en el aula como recurso comporta distintos niveles de comunicación, por un lado la relacionada con los medios electrónicos y por otro lado la comunicación verbal y escrita. Favorece el debate ya que todos los alumnos han participado en una experiencia individual y grupal al mismo tiempo. La facultad de este juego en concreto permite la conexión entre todos los alumnos en una serie de equipos que pueden o no, trabajar juntas para la consecución de los objetivos propuestos y para ello se facilita discusiones-negociaciones donde los participantes, a modo de foro discuten sus alianzas y sus próximos pasos. [25]

---

[25] www.ub.es/personal/videoju.htm Etxeberria, F. “Videojuegos y educación”.(Última visita, 3 de junio de 2005)

- Competencias para la crítica reflexiva

Al favorecer el diálogo entre los alumnos a partir de la competencia anterior se forma la crítica entre los participantes. La creación de estrategias conjuntas o no, el avance en un determinado campo o investigación. Todas las decisiones tienen consecuencias directas en el videojuego que deben ser valoradas en su justa medida; en función del tipo de decisiones tomadas así serán las acciones. Dar competencias para ejercer la crítica reflexiva en los medios favorece la resistencia a la manipulación que dichos medios puedan ejercer creando ciudadanos competentes para construir la vida social.[26]

3.3 TÉCNICAS DE APRENDIZAJE

El aprendizaje es el proceso a través del cual se adquieren o se modifican ciertas conductas, actitudes, habilidades, destrezas, conocimientos, o valores como resultado del estudio, la experiencia, la instrucción, el razonamiento y la observación.

3.3.1. Tipos de Aprendizaje

Existen muchos tipos de aprendizajes los más comunes citados por la pedagogía moderna son:

Aprendizaje receptivo.
Aprendizaje por descubrimiento.
Aprendizaje repetitivo.

---

[26] Buckingham, D. (2002). Crecer en la era de los medios electrónicos. Madrid: Morata

Aprendizaje significativo
Aprendizaje observacional
Aprendizaje latente.

3.3.2. Teorías del Aprendizaje

El aprendizaje y las teorías que tratan los procesos de adquisición de conocimiento han tenido durante este último siglo un enorme desarrollo debido fundamentalmente a los avances de la psicología y de las teorías instruccionales, que han tratado de sistematizar los mecanismos asociados a los procesos mentales que hacen posible el aprendizaje. Existen diversas teorías del aprendizaje, cada una de ellas analiza desde una perspectiva particular el proceso.

Algunas de las más difundidas son:

3.3.2.1. Teorías conductistas.

Condicionamiento clásico: Pávlov, a principios del siglo XX, propuso un tipo de aprendizaje en el cual un estímulo neutro (tipo de estímulo que antes del condicionamiento, no genera en forma natural la respuesta que nos interesa) genera una respuesta después de que se asocia con un estímulo que provoca de forma natural esa respuesta.

Conductismo. Desde la perspectiva conductista, formulada por B.F. Skinner (Condicionamiento operante) hacia mediados del siglo XX y que arranca de los estudios psicológicos de Pavlov sobre Condicionamiento clásico y de los trabajos de Thorndike (Condicionamiento instrumental) sobre el esfuerzo, intenta explicar el aprendizaje a partir de unas leyes y mecanismos comunes para todos los individuos.

Reforzamiento. B.F. Skinner, propuso para el aprendizaje repetitivo un tipo de reforzamiento, mediante el cual un estímulo aumentaba la probabilidad de que se repita un determinado comportamiento anterior. Desde la perspectiva de Skinner, existen diversos reforzadores que actúan en todos los seres humanos de forma variada para inducir a la repetitividad de un comportamiento deseado.

3.3.2.2. Teorías Cognitivas:

Aprendizaje por descubrimiento. J. Bruner, atribuye una gran importancia a la actividad directa de los estudiantes sobre la realidad.

Aprendizaje significativo (D. Ausubel, J. Novak) postulan que el aprendizaje debe ser significativo, no memorístico, y para ello los nuevos conocimientos deben relacionarse con los saberes previos que posea el aprendiz.

Cognitivismo. La psicología cognitivista (Merrill, Gagné…), está basada en las teorías del procesamiento de la información y recoge también algunas ideas conductistas (refuerzo, análisis de tareas) y del aprendizaje significativo que aparece en la década de los sesenta y pretende dar una explicación más detallada de los procesos de aprendizaje.

Constructivismo. Jean Piaget, propone que para lograr el aprendizaje es necesario un desfase óptimo entre los esquemas que el alumno ya posee y el nuevo conocimiento que se propone. “Cuando el objeto de conocimiento está alejado de los esquemas que dispone el sujeto, este no podrá atribuirle significación alguna y el proceso de enseñanza/aprendizaje será incapaz de desembocar”.

Socio-constructivismo. Se basa en las ideas de Vigotski, que considera también los aprendizajes como un proceso personal de construcción de nuevos conocimientos a partir de los saberes previos (actividad instrumental), pero

inseparable de la situación en la que se produce. El aprendizaje es un proceso que está íntimamente relacionado con la sociedad[27].

3.3.2.3. Teoría del procesamiento de la información.

La teoría del procesamiento de la información, influida por los estudios cibernéticos de los años cincuenta y sesenta, presenta una explicación sobre los procesos internos que se producen durante el aprendizaje.

Conectivismo. Pertenece a la era digital, ha sido desarrollada por George Siemens que se ha basado en el análisis de las limitaciones del conductismo, el cognitivismo y el constructivismo, para explicar el efecto que la tecnología ha tenido sobre la manera en que actualmente vivimos, nos comunicamos y aprendemos. [28]

[27] http://www.slideshare.net/paula1887/collection-teorias-del-aprendizaje

[28] http://www.s2is.org/Issues/v1/n1/papers/paper9.pdf

CAPÍTULO 4.
ESTADO DEL ARTE

Un videojuego web es un programa informático interactivo destinado al entretenimiento que puede funcionar en diversos dispositivos: ordenadores, consolas, teléfonos móviles, etcétera; integra audio y vídeo, y permite disfrutar de experiencias que, en muchos casos, sería muy difícil de vivir en la realidad.

Como apunta Ortega (2001), su estructura narrativa es muy variada. Así, encontramos argumentos basados en la apología, la parábola, la alegoría, la crónica, los relatos de viaje, los cuentos clásicos, los mitos, los relatos oníricos, los ritos iniciáticos o los juegos de rol. Aunque hemos dicho que su origen es lúdico, hoy en día se han ampliado y sobrepasado los límites del entretenimiento, porque se han abierto posibilidades de aplicación en el ámbito educativo [29].

4.1. ANTECEDENTES

Hoy en Colombia se ha presentado una visible evolución frente al ámbito tecnológico relacionado con el educativo. Un planteamiento común en la educación de programación es enseñar primero los conceptos básicos de un lenguaje de programación, y ofrecer una guía a los estudiantes hacia las estrategias eficaces para el proceso de programación. Por lo tanto, el aprendizaje de los conceptos básicos a menudo forma la base para el desarrollo de habilidades más avanzadas.

[29] http://www.irabia.org/departamentos/nntt/proyectos/futura/futura06/Analisis_educativo.pdf

4.1.1. Videojuegos Tipo Herramienta Educativa En Aprendizaje Lógico

La lógica es muy importante, ya que permite resolver incluso problemas a los que nunca se ha enfrentado el ser humano utilizando solamente su inteligencia y apoyándose de algunos conocimientos acumulados, se pueden obtener nuevos inventos e innovaciones a las ya existentes o simplemente utilizados en los mismos [30].

Considera (Gonzales, Walfredo, 1987). “Se plasma la Inteligencia Lógica Matemática para el desarrollo de la creatividad informática, ya que actualmente deben aprender a manejar la computadoras dando punto de partida a estos conocimientos, a partir de los cinco años de edad”.

Considerando que sí el alumno aprende lógica matemática no tendrá problemas para aprender ciencias exacta y será capaz de programar computadoras, ya que un programa de computadora no es otra cosa que una secuencia de pasos lógicos que la persona establece para resolver un problema determinado[31].

Esta investigación, es de carácter científico porque consta de criterios, que son comprobados y verificados por los investigadores. Esto sirve para enseñar a pensar y razonar científicamente a los alumnos.

Como así lo indica en el Diccionario de Psicología “Las expectativas del profesor son determinantes para hacer predicciones sobre lo que un alumno puede llegar aprender” (Rosewthar, 1998).

---

[30] http://conociendolasmatematicas.files.wordpress.com/2012/11/la-matematica.pdf

[31] http://repositorio.utc.edu.ec/bitstream/27000/233/1/T-UTC-0259.pdf

El niño, de acuerdo a las experiencias adquiridas irá construyendo sus conocimientos, por lo cual, es conveniente proporcionar al niño de material, de vivencias, por lo que hemos desarrollado este videojuego web el cual llama poderosamente la atención a los niños y despierta su curiosidad, lo cual a través de manipuleo, indagación y experiencia le llevará a seleccionar, asimilar e interpretar un aprendizaje.

Según considera (LEMUS, Luis, 2001, Pág. 65) en el Libro LOOS Sigrid. Juegos y actividades para el desarrollo físico y psíquico del niño, edición Narcea, Estados Unidos, Tercera Edición, 2007. "La inteligencia humana no tiene límites, es casi infinito nuestro potencial, diferentes textos y expositores plantean el poder que tiene la visión, los sueños, las ganas de hacer las cosas, la actitud mental positiva" [33].

En el Libro Inteligencias Múltiples y estimulación temprana de Gonzale, Walfredo, 1987, Pág. 123; "Se plasma la inteligencia lógica matemática para el desarrollo de la creatividad informática ya que actualmente todos deben aprender a manejar las computadoras, dando punto de partida a estos conocimientos, a partir de los 5 años de edad".

Confirmando la necesidad de fortalecer algunas nociones lógico matemáticas en los niños hemos desarrollado un videojuego web, el cual constituye una herramienta adecuada para cubrir las necesidades de cada uno de los infantes y lograr un optimo desarrollo de la inteligencia .

Conociendo que el juego es una de las actividades más importantes y un medio por el cual el infante aprende a conocer su entorno, hemos creado esta herramienta web, la cual se enfoca al desarrollo de la inteligencia Lógica, siendo de gran ayuda para desarrollar de mejor manera la labor educativa y que el aprendizaje en los niños/as tenga un significado profundo en su vida.

4.2. PLATAFORMAS QUE ENSEÑEN A PROGRAMAR

4.2.1. ALICE

El proyecto Alice es una iniciativa de varias universidades encabezado por la Universidad de Carnegie Mellon, en la que existe un Alice Team, conformado por profesores, personal y estudiantes[32].

Alice se define como un entorno innovador de programación 3D que hace que sea fácil crear una animación para contar una historia, crear un juego interactivo, o un video para compartir en la web.

Alice es una herramienta de enseñanza de libre disposición, es decir, es gratuita y disponible en el Internet para todo el mundo con mínimos requisitos de hardware y software, capaz de ser ejecutada en cualquier sistema operativo. Esta herramienta ha sido diseñada para ser la primera exposición de un estudiante de programación orientada a objetos.

Invita a los estudiantes a aprender los conceptos fundamentales de programación en el contexto de la creación de películas animadas y juegos de vídeo de manera fácil. En Alice, objetos 3-D (por ejemplo, personas, animales y vehículos), pueblan un mundo virtual y los estudiantes crean un programa para animar los objetos.

Para crear un programa en la interfaz interactiva de Alicia, los estudiantes sólo deben de arrastrar y soltar bloques gráficos. Cada bloque corresponde según las instrucciones empleadas, a las declaraciones estándar en un lenguaje de programación orientado a objetos, tales como Java, C + + y C #. De esta forma el estudiante no requiere inicialmente conocer todas las reglas de un lenguaje de

[32] http://www.postgradoeinvestigacion.uadec.mx/CienciaCierta/CC29/8.html

programación para realizar programas, y sí podrá ir comprendiendo el funcionamiento y comportamiento de los diferentes elementos disponibles para cada objeto que haya integrado a su mundo virtual.

Alice permite ver de inmediato cómo se ejecutan sus programas de animación, lo que les ayuda a comprender fácilmente la relación entre las declaraciones de la programación y el comportamiento de los objetos en su animación. El lograr la ejecución de un programa es una de las primeras barreras al utilizar la programación, aquí sucede casi instantáneamente, ya que puede ejecutar el programa elaborado en cualquier momento, sin riesgos de que no compile o peor aún que le aparezcan mensajes de error complicados de interpretar y corregir.

Mediante la manipulación de los objetos en su mundo virtual, los estudiantes adquieren experiencia con todas las construcciones de programación que típicamente se enseñan en un curso introductorio a la programación. Aquí nos referimos a que de forma gráfica y simultánea el estudiante puede ver las diferentes estructuras de programación, tales como decisiones, secuencias de instrucciones, ciclos, pausas, mensajes, etc.

La creación de un programa con Alice resulta muy fácil de obtener para cualquier persona, gracias al entorno visual que presenta. En primer lugar al activar el programa, el usuario, debe seleccionar un ambiente para desarrollar el programa, como arena, hierba, espacio o nieve entre otros.

Una vez que se ha elegido y definido el mundo virtual, se proporciona acceso a los elementos de la interface gráfica de Alice. En dicho mundo virtual se puede agregar cualquier cantidad de objetos tridimensionales disponibles en una amplia biblioteca, clasificados temáticamente. Cada objeto en Alice posee un conjunto de

acciones (moverse, girar, cambiar de color, etc.) a través de las cuales puede interactuar dentro del mundo donde se utiliza[33].

La interface gráfica se divide en cinco áreas:

- Ventana del mundo. En ella se visualiza el mundo que se va construyendo con los objetos que se hayan integrado.
- Ventana de objetos. En ella se muestra en forma jerárquica o de árbol, los objetos incluidos. La raíz corresponde al mundo y dentro de ella se ligan todos los objetos añadidos.
- En cualquier momento debe estar seleccionado un objeto o mundo.
- Ventana de detalles. En esta ventana se visualizan en tres categorías las características del objeto seleccionado: propiedades, métodos y funciones. Cabe mencionar que dependiendo de cada objeto dichas propiedades, métodos y funciones son específicas.
- Ventana de programa. Corresponde a la visualización de los métodos disponibles en un mundo virtual. A través de la técnica de arrastrar y soltar aquí se puede detallar el “programa”, es decir, la secuencias de acciones que deberán realizar cuando se ejecute el programa.
- Ventana de eventos. En esta se agregan nuevos eventos, que pueden definir a través de la ventana de programas.

Consideremos el primer ejemplo al que llamaremos “Rutina de patinaje”. La secuencia para programarlo es la siguiente:

- Seleccionar plantilla de mundo de nieve (snow).
- Agregar patinadora usando el objeto “ice Skater”.
- Seleccionar el objeto “ice Skater” y luego ubicarse en la ventana de detalles.

[33] http://www.postgradoeinvestigacion.uadec.mx/CienciaCierta/CC29/8.html

- Selecciona la categoría “Properties” y arrastra a la ventana de programa “pose3”.
- Ahora seleccionar la categoría “Methods” y arrastrar a la ventana de programa “iceSkater.move”. Se requiere elegir la dirección “Forward” y la distancia “1 meter”.
- Arrastrar a la ventana de programa “iceSkater.turn”. Se requiere elegir la dirección “Left” y la cantidad “2 revolutions”.
- Arrastrar a la ventana de programa “iceSkater say”. Se requiere elegir otro mensaje “Other” y escribir “Hola”.
- Ejecuta con el botón de “Play”.

Finalmente se guarda el mundo construido con el nombre “Rutina de patinaje”.

Figura 1. Learning to program with Alice

4.2.2. SCRATCH

Scratch es un entorno de aprendizaje de lenguaje de programación, que permite a los principiantes o personas sin sentido de la programación, obtener resultados sin tener que hacer nada de nada o aprender a escribir de manera sintácticamente correcta.

Scratch está escrito en *Squeak* (una implementación libre de*Smalltalk-80*), a partir de la versión 2.0 el código es reescrito en *actionscript* (*Adobe Flash*). Es un entorno de programación que facilita el aprendizaje autónomo.

Fue desarrollado por "el grupo permanente *Kindergarten*" en el Media Lab del MIT (Instituto Tecnológico de Massachusetts) por un equipo dirigido pJor Mitchel Resnick, apareció por primera vez en el año 2007.

Scratch se puede instalar y redistribuir gratuitamente en cualquier ordenador con*Windows*, *Mac OS X* o *Linux*.[4]El logo de *Scratch* es un gato de color naranja.

El programa *Scratch* se encuentra en el software de las laptops XO del Plan CeibalenUruguay. Podemos utilizar este para programar, tal como dice su lema: programar, jugar y crear. En *Scratch* todos los objetos, gráficos, sonidos y secuencias de comandos pueden ser fácilmente importados a un nuevo programa y combinados en maneras permitiendo a los principiantes conseguir resultados rápidos y estar motivados para intentar más.

El rol del educador cambia al utilizar dicho tipo de software, ya que no se limita a enseñar contenidos, sino más bien a incentivar la exploración y el descubrimiento por parte de los alumnos, haciéndoles notar las diversas estrategias que pueden utilizar para llegar a alcanzar distintos aprendizajes.

Como ejemplo está “Scratch”, el cual es un proyecto de código abierto, libre y gratuito que tiene una interfaz con notables avances en diseño, logrando que la programación sea más llamativa y accesible, para aquellos que se enfrenten a programar por primera vez. Dicho proyecto, busca promover el desarrollo de habilidades del siglo XXI, además de incentivar las capacidades intelectuales y el pensamiento algorítmico de los estudiantes de básica primaria.

Scratch, fue utilizado por un grupo de docentes de Informática, Matemáticas y Ciencias Naturales en escuelas vulnerables de Cali Colombia, con el fin de lograr

los objetivos anteriormente mencionados. Se debe especificar que al programar con Scratch, los educandos pueden realizar actividades que capturan su atención como crear historias interactivas, animaciones, creaciones artísticas, entre otras, e ir programando de una forma sencilla y divertida. Por tal razón, hubo un gran avance y productividad con el aprendizaje de los niños participantes del proyecto, y luego de realizar una retroalimentación entre docentes y estudiantes acerca de este, se observó que presentó beneficios en su proceso de enseñanza-aprendizaje, por esto, se llevó a cabo en grupos más amplios.

A continuación se presentará una imagen representativa de la interfaz gráfica de Scratch

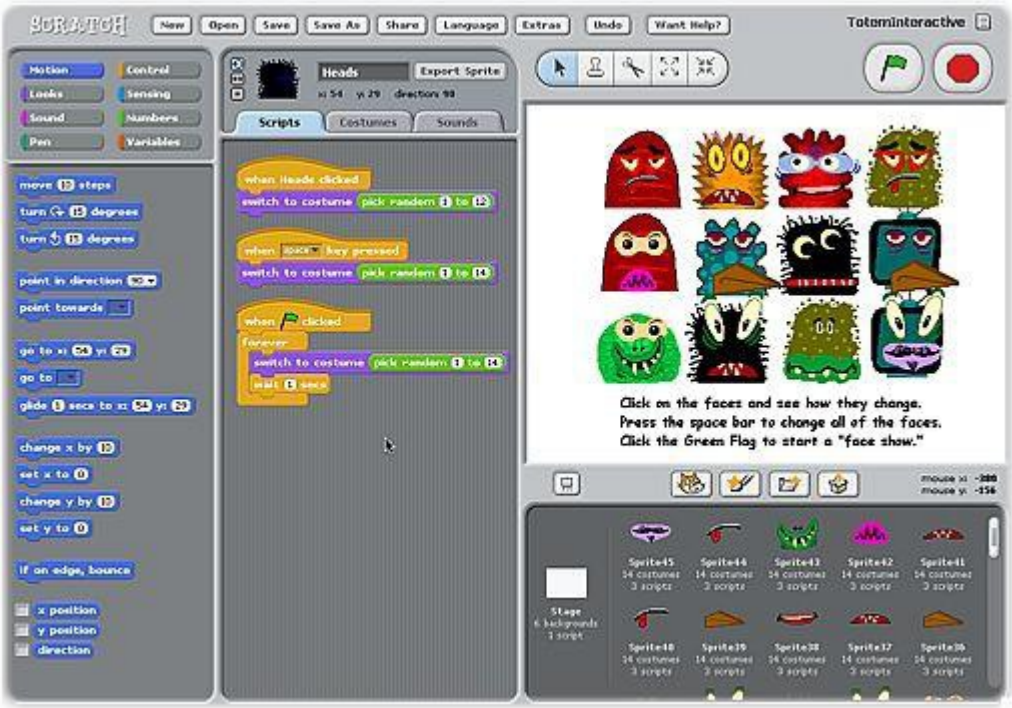

Figura 2. Interfaz gráfica de Scratch

Teniendo en cuenta lo anterior, prevalece la creación de un videojuego para niños, asumiendo los logros alcanzados con el programa “Scratch” elaborado por un grupo de investigadores del Lifelong Kindergarten Group del Laboratorio de Medios del MIT, bajo la dirección del Dr. Mitchael Resnick, dado que luego de una

detallada observación e investigación se concluyó que los niños de primero y segundo grado demuestran interés por algunos artefactos tecnológicos, especialmente por los computadores y las consolas para videojuegos. (Harbey B. 2010)

4.2.3. PSeINT

PSeInt es un programa creado en la Facultad de Ingeniería de la UNL (Universidad del Litoral) como ayuda para aprender lógica de programación: mediante el uso de pseudo-código, permite aprender de una forma muy sencilla lo fundamental de un algoritmo de computación. El pseudo-código en el que se basa **PSeInt** es un pseudo-lenguaje que, mediante el uso de 4 premisas básicas: sintaxis sencilla, manejo de las estructuras básicas de control, el uso de solo 3 tipos de datos básicos y los arreglos de las estructuras de datos- permite dominar los conceptos más básicos de la programación[34].

PSeInt es un intérprete de pseudocódigo. El pseudocódigo debería servir para enseñar los conceptos, la lógica básica, la abstracción necesaria para empezar a programar, sin tener que preocuparse por un lenguaje y sus particularidades. PSeInt ofrece un editor con coloreado de sintaxis, autocompletado, ayuda rápida en pantalla, la posibilidad de visualizar el algoritmo como diagrama de flujo, de convertirlo luego a código C++, etc, y lo más importante: la posibilidad de que la computadora interprete y ejecute el algoritmo en caso de ser válido, o señale los errores en caso contrario. Siendo lo segundo es mucho más difícil e interesante que lo primero.

PSeInt está pensado para asistir a los estudiantes que se inician en la construcción de programas o algoritmos computacionales. El pseudocódigo se

[34] http://grupodetrabajo101.blogspot.com/2010_09_01_archive.html

suele utilizar como primer contacto para introducir conceptos básicos como el uso de estructuras de control, expresiones, variables, etc, sin tener que lidiar con las particularidades de la sintaxis de un lenguaje real. Este software pretende facilitarle al principiante la tarea de escribir algoritmos en este pseudolenguaje presentando un conjunto de ayudas y asistencias, y brindarle además algunas herramientas adicionales que le ayuden a encontrar errores y comprender la lógica de los algoritmos[35].

Caracteristicas y Funcionalidades de PSeInt:

- ✓ Presenta herramientas de edición para escribir algoritmos en pseudocódigo en español
- ✓ Autocompletado
- ✓ Ayudas Emergentes
- ✓ Plantillas de Comandos
- ✓ Coloreado de Sintaxis
- ✓ Indentado Inteligente
- ✓ Permite generar y editar el diagrama de flujo del algoritmo
- ✓ Permite la edición simultánea de múltiple algoritmos
- ✓ El lenguaje pseudocódigo utilizado es configurable
- ✓ Ofrece perfiles de configuración predefinidos para numerosas instituciones
- ✓ Puede interpretar (ejecutar) los algoritmos escritos
- ✓ Permite ejecutar el algoritmo paso a paso controlando la velocidad e inspeccionando variables y expresiones
- ✓ Puede confeccionar automáticamente una tabla de prueba de escritorio
- ✓ Determina y marca claramente errores de sintaxis (mientras escribe) y en tiempo de ejecución
- ✓ Permite convertir el algoritmo de pseudocodigo a código C++

[35] http://pseint.sourceforge.net/index.php?page=features.php

- ✓ Ofrece un sistema de ayuda integrado acerca del pseudocódigo y el uso del programa

Incluye un conjunto de ejemplos de diferentes niveles de dificultad

- ✓ Es multiplataforma (probado en Microsoft Windows, GNU/Linux y Mac OS X)
- ✓ Es totalmente libre y gratuito (licencia GPL)

Reglas generales para elaboración de Pseudocódigo con PSeint

La redacción de c/u de los pasos de Pseudocódigos con el software PSeint debe cumplir “sintaxis especificas”, entre las cuales se pueden mencionar las siguientes:

· A cada pseudocódigo se le identifica como un “Proceso”, por lo cual, se debe reemplazar “sin_titulo” por un nombre apropiado.
· Las acciones simples como Escribir y Mostrar, deben finalizar con (;)
· Cualquier variable debe ser declarada antes de ser utilizada, de lo contrario se desencadenara un error de “variable no inicializada”
· Debe tener mucho cuidado con la redacción de los identificadores de las variables (nombre asignado), porque PSeint es “sensible al texto”, diferencia entre Mayúsculas y minúsculas.
· Para redactar las expresiones matemáticas, debe tomar muy en cuenta la “jerarquía de operadores”, para obtener los resultados esperados.

Pruebas y depuración del Pseudocódigo

Una vez ha creado el Pseudocódigo del Algoritmo, PSeint permite “ejecutar” la secuencia de pasos, con el fin de comprobar si se obtienen la información esperada/requerida. Se debe guardar el archivo del pseudocódigo y luego presionar el botón Ejecutar… del Panel de Ejecución.

La ejecución se muestra en una “ventana de comandos” y se activa un marco de “Resultados” en la parte inferior del Editor de Pseudocódigos.

PSeint diferencia con los siguientes “colores” la ejecución correcta/incorrecta de los diferentes pasos, asi:

+ Color azul, indica la ejecución del paso inicial y final del pseudocódigo.

+ Color amarillo: datos que usuario digita gracias a una acción “Leer”

+ Color verde: cadenas de salida de datos que se muestran al usuario, gracias a la acción “Escribir”

+ Color rojo: avisa de errores de sintaxis o durante ejecucion. Identifica el # de línea y una descripción del error.

En el “marco de Resultados” se confirman las respuestas anteriores.

Guía* # 01*: Introducción al Software DFD y PSeint

Introducción a la Programación

Se puede utilizar el botón “Ejecutar paso a paso…” para ejecutar instrucción por instrucción del pseudocódigo, con el fin de “hacer un seguimiento de los cambios en el valor de las variables” (Elaborar la Traza de Ejecución del Pseudocódigo[36].

[36] http://www.udb.edu.sv/udb/archivo/guia/informatica-tecnologico/introduccion-a-la-programacion/2013/i/guia-1.pdf

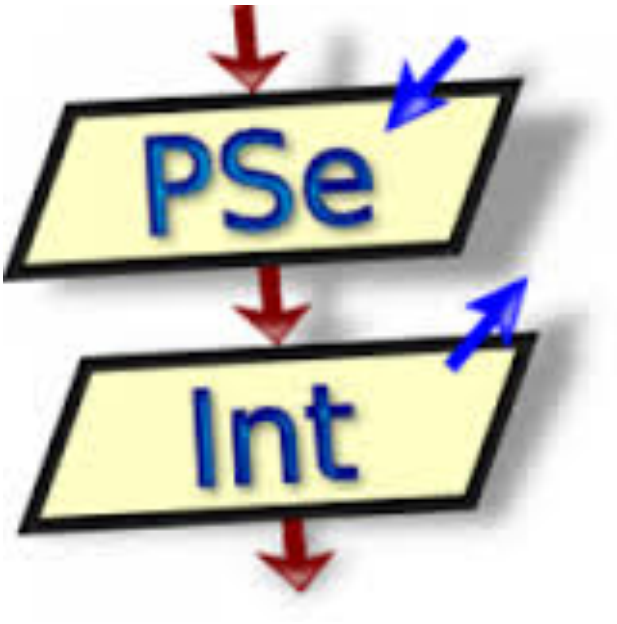

Figura 3. Logo PSeInt

4.3 HCI (Human Computer Interaction)

La Association for Computing Machinery define la interacción persona-ordenador como "la disciplina que estudia el diseño, evaluación e implementación de sistemas informáticos interactivos para uso humano y con el estudio de los principales fenómenos que los rodean", mientras que otros definen HCI como el estudio, la planificación y diseño de la interacción entre personas y ordenadores y a menudo considerado como la intersección de las ciencias de la computación, ciencias de la conducta, el diseño y varios otros campos de estudio. Interacción entre usuarios y equipos se produce en la interfaz de usuario, que incluye software y hardware. (Lorés, 2002).

Porque humana- computer interaction estudia un humano y una máquina en conjunto, se basa en el apoyo a los conocimientos tanto en el equipo y el lado humano. Por el lado de la máquina, las técnicas de gráficos por ordenador, sistemas operativos, lenguajes de programación y entornos de desarrollo son relevantes. En el lado humano, la teoría de la comunicación, la gráfica y las

disciplinas de diseño industrial, la lingüística, las ciencias sociales, la psicología cognitiva y factores humanos tales como la satisfacción del usuario de la computadora son relevantes. Los métodos de ingeniería y de diseño también son relevantes. (Lorés, 2002).

4.3.1. Historia De La IPO

A finales de los años 1970 y principios de los 80, empiezan a aparecer áreas de estudio como la de diseño de menús, que utilizan tópicos como el de nombres de menús o profundidad versus anchura. Hacia finales de los 80 y en los 90, Se hace hincapié en las comunicaciones más avanzadas, en las aplicaciones multimedia, en los ordenadores multitarea y en la realidad virtual. También es necesario que se consideren otros tópicos como son: trabajo en grupo, integración e interacción de medios y el impacto de estas tecnologías en el lugar de trabajo, en la casa y en la sociedad en general. (Toni Granollers, 2002)

La IPO surgió como un campo entrecruzado de disciplinas:

- Gráficos por ordenador (Gráficos Interactivos).
- Sistemas operativos (Sistemas de Gestión de la Interfaz de Usuario y Kits de Herramientas —toolkits— de la Interfaz de Usuario).
- Ergonomía.

Los cambios más significativos en el mundo de la IPO:

- Manipulación directa de objetos gráficos
- El ratón (mouse)
- Las ventanas (windows)
- Programas de dibujo y pintura
- Edición y procesamiento de texto

- Hoja de cálculo
- Hipertexto
- Reconocimiento de gestos

Durante el periodo comprendido de 1963 a 1982 se crea La Manipulación Directa de Objetos Gráficos. (Toni Granollers, 2002)

El ratón

Es el segundo dispositivo de entrada más utilizado. El mouse o ratón es arrastrado a lo largo de una superficie para maniobrar un apuntador en la pantalla del monitor. Fue inventado por Douglas Engelbart y su nombre se deriva por su forma la cual se asemeja a la de un ratón.
Otro tipo de dispositivo similar es el lápiz óptico, la tableta digitalizadora o incluso las pantallas táctiles. (Sergi Lana, 2002)

Las ventanas

Periodo de 1968 a 1995, donde podemos destacar:
1979. Lisp Machines Inc. Uno de los sistemas comerciales que primero introdujo el uso de ventanas fue en Lisp Machines Inc. (LMI) y en Symbolics Lisp Machines.
1981. Xerox PARC. El Cedar Window Manager fue el gestor de ventanas con más relevancia de la época [SWI86].
1983. Carnegie Mellon University's Information Technology Center. Al anterior gestor de ventanas, le siguió el Andrew.
1984. Apple Macintosh. El Apple Macintosh, junto con sus predecesores (Xerox Star en 1981 y Apple Lisa en1982), ayudó a popularizar el uso de las ventanas en las aplicaciones informáticas. Este sistema utilizaba ventanas traslapadas.

1984. MIT. El sistema de ventanas estándar X Window (que utilizan muy a menudo los sistemas UNIX), fue desarrollado en 1984 por investigadores del MIT [SCH86].
1985. MicroSoft. Sale la primera versión de MS-Windows (1.0) sobre MS-DOS.
1992. MicroSoft. MS-Windows for WorkGroups (3.1).
1995. MicroSoft. MS-Windows NT 3.51 y MS-Windows 95.

Programas de dibujo y pintura

Periodo de 1963 a 1985 para mencionar algunos:

1965. NLS. Uno de los periféricos imprescindibles en el mundo de las aplicaciones de dibujo es el ratón. El uso de este dispositivo de entrada fue demostrado en 1965 por NLS.
1968. Ken Pulfer & Grant Bechthold, National Research Council of Canada. En 1968 se construyó un ratón que se utilizó en un sistema de animación de largometrajes para dibujar los fotogramas de una película.
1974-1975. Dick Shoup, Xerox PARC. El primer programa de ordenador para pintar fue el Superpaint.
1985. Digital Research, Inc. GEM Paint Version 2.01. Programa de dibujo de Mapa de Bits. (Lorés, Granollores, Delgado, 2005)

Edición y procesamiento de textos.

Periodo de 1962 a primeros años de los 90, donde podemos mencionar:

1962. Engelbart, Stanford Research Lab. Se implementó un procesador de textos con word wrap automático, buscar y reemplazar, macros definidas por el usuario, scroll y comandos para mover, copiar y borrar caracteres, palabras o bloques de texto.
1967. MIT. TECO fue uno de los primeros editores de texto de pantalla.

1990. Ventura Software Inc. Desarrollan este postprocesador de textos que usaba la técnica de WYSIWYG muy avanzada. Aunque ya empezaron a aparecer versiones en 1986 fue en 1990 cuando se puede utilizar en entorno PC junto con GEM.
Actualmente ha derivado a la familia de productos Corel.
primeros años 1990. Varios procesadores de texto fueron populares en entornos PC entre los que cabe destacar el “Word Star”, el “MS-Word” –el precursor del mayormente utilizado hoy en día–. (Lorés, Granollores, Delgado, 2005)

Hoja de cálculo

1977-1978. Frankston & Bricklin, MIT & Harward Business School. Estos estudiantes realizaron la primera hoja de cálculo de la historia.
1982. LOTUS DEVELOPMENT CORPORATION. Se funda la compañía e introduce la primera versión de Lotus 1-2-3.

Hipertexto

1945. Vannevar Bush. Se introdujo la idea de Hipertexto (MEMEX idea).
1965. Ted Nelson. Acuñó la palabra hipertexto [NEL65].
4.6.2. Objetivos de la IPO
Los objetivos de la IPO son desarrollar o mejorar la seguridad, utilidad, efectividad, eficiencia y usabilidad de sistemas que incluyan computadoras [DIA89]. Cuando decimos sistemas no nos referimos tan solo al hardware y al software sino también a todo el entorno.

4.3.2. La interfaz de usuario

La interfaz es una superficie de contacto entre dos entidades. En la interacción persona-ordenador estas entidades son la persona y el ordenador.

En la vida cotidiana tenemos muchos ejemplos de interfaz. En el caso de la puerta, la manilla de la puerta es la interfaz entre la puerta y la persona. El volante, el acelerador y otros instrumentos y herramientas son la interfaz entre un coche y el conductor. Es muy importante darse cuenta en un primer nivel de que la interfaz refleja las cualidades físicas de las dos partes de la interacción. (Lorés, Granollores, Delgado, 2005)

4.3.3. La interdisciplinariedad de la IPO

Como vemos, hemos de tener en cuenta muchas disciplinas para comprender toda la problemática que supone el desarrollo de interfaces. Esto supone que, a menudo, tengamos que pensar en un equipo interdisciplinario para el desarrollo de sistemas interactivos, cuando el problema a tratar es de la suficiente envergadura. En otros casos, no obstante, cuando el equipo de trabajo sea reducido, las mismas personas deberán jugar los diferentes papeles y tratar de hacerlo bien por sí mismos.

Principales disciplinas a tener en cuenta:

Psicología

La psicología contribuye a la IPO mediante conocimientos y teorías acerca de como los sujetos se comportan, procesan la información y actúan en grupos y organizaciones, pero también proporciona metodologías y herramientas para evaluar y determinar el grado de satisfacción de éstos a nuestros diseños.

Diseño

Actividad encaminada a conseguir la producción en serie de objetos útiles y bellos. Esta es una disciplina muy importante dentro de la IPO, tal como podemos ver en su definición para conseguir programas usables.

Etnografía - Sociología

Es la ciencia que estudia las costumbres y las tradiciones de los pueblos. Estas compañías están convencidas que las herramientas de investigación etnográfica —observación detallada, entrevistas sutiles, documentación sistemática—,

Ergonomía o factores humanos

Su objetivo es maximizar la seguridad, la eficiencia y la fiabilidad para simplificar las tareas e incrementar la sensación de confort y satisfacción. La ergonomía o factores humanos se centran el estudio de las características físicas de la interacción, por ejemplo el entorno físico donde se produce esta y la forma y las peculiaridades físicas de la pantalla.

Describir completamente la ergonomía podría ocupar casi un curso completo, tanto como para hacer una disciplina aparte. (Lorés, Granollores, Delgado, 2005)

Algunos de los aspectos que se podrían considerar son los siguientes:

- El usuario ha de poder acceder fácilmente a todos los controles y ver toda la información que se le presenta sin mover excesivamente el cuerpo.
- La información más importante ha de estar situada a la altura de los ojos.
- El entorno físico de la interacción.
- Aspectos de salud. la posición física, la temperatura, la luz, el ruido, el tiempo de permanencia delante del ordenador, etc.
- El uso de los colores. su distinción no se debe ver afectada por los cambios de contraste.

Programación

La programación es la herramienta que nos permite “decirle al ordenador lo que debe hacer”.

Un programa:

- ✓ debe funcionar
- ✓ no debe tener dificultades
- ✓ debe estar bien documentado
- ✓ debe ser eficiente

Dentro de la programación podemos englobar tanto los propios lenguajes de programación (Java, C, C++, Pascal, Fortran, BASIC, etc.) como todas las herramientas de diseño y programación que permiten el desarrollo de las interfaces.

Básicamente podemos clasificar los tipos de programación en cinco grandes grupos:

ORIENTADA a OBJETOS.
Un Objeto es una instancia de una Clase.
Es la programación mayormente utilizada hoy en día.
Ejemplos: C++, Java

IMPERATIVA (Entrada, procesamiento y salidas de Datos):
Es llamada así porque esta basada en comandos que actualizan variables que están en almacenamiento.
Su principal característica es que están orientados a la arquitectura del computador.

Ejemplos: PASCAL-ADA

- FUNCIONAL:

La principal característica de este tipo de programación reside en que “los datos son funciones y los resultados pueden ser un valor o una función”.
La Programación Funcional se caracteriza por el uso de Expresiones y Funciones mientras que la Imperativa por el uso de variables, comandos y procedimientos.

- DECLARATIVA:

Los lenguajes declarativos están basados en el modo de pensar de los humanos en lugar del ordenador.

- CONCURRENTE:

La Programación Concurrente debe frecuentemente estar acompañada por el avance del Hardware y sus arquitecturas correspondientes.
Ejemplos: ADA es un lenguaje familiar y tiene una aproximación a la Concurrencia.

Inteligencia artificial

La inteligencia artificial trata de diseñar programas de ordenador inteligentes que simulen diferentes aspectos del comportamiento humano inteligente. (Lorés, Granollores, Delgado, 2005)

Ingeniería de software

Es importante tener en cuenta la ingeniería de software en el desarrollo de un sistema interactivo. Esta disciplina estudia técnicas de diseño y desarrollo del software. Para realizar diferentes aplicaciones se deben utilizar procedimientos propios de ingeniería. Sólo con estos procedimientos y técnicas vamos a obtener un software de calidad.

4.3.4 Usabilidad

Para que un sistema interactivo cumpla sus objetivos tiene que ser usable y, además, debido a la generalización del uso de los ordenadores, accesible a la mayor parte de la población humana.

Podemos definir la usabilidad como la medida en la que un producto se puede usar por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado.

Principios generales de la usabilidad

Principios generales que se pueden aplicar a un sistema interactivo:

- Facilidad de aprendizaje.
- Consistencia.
- Flexibilidad.
- Robustez.
- Recuperabilidad.
- Tiempo de respuesta.
- Adecuación de las tareas.
- Disminución de la carga cognitiva.

4.3.5 El diseño centrado en el usuario.

El diseño de sistemas interactivos implica realizar un diseño pensando en el usuario, centrando nuestro sistema de desarrollo en él e implicarlo tanto como sea posible, hasta pensar en incluir usuarios en el equipo de diseño. (Lorés, Granollores, Delgado, 2005)

CAPÍTULO 5
IDENTIFICACIÓN DE LA METODOLOGÍA DE ENSEÑANZA

5.1. MÉTODOS DE APRENDIZAJE

Algunos de los métodos de enseñanza inspeccionados son:

5.1.1. Método de Aprendizaje por Proyectos

Este método es basado en la teoría constructivista del aprendizaje, anteriormente mencionada en el marco teórico. Se fija en los intereses del estudiante para que proyecten, realicen y analicen investigaciones que hagan parte de su entorno, además de identificar un problema para hallar su solución. (Blank, 1997).

El aprendizaje por proyectos fue creado en 1918 con el ánimo de mejorar el aprendizaje, brindándole autonomía al estudiante. Las TIC, contribuyen a esta metodología que, a pesar de tener casi un siglo de antigüedad, sigue aún vigente. (Vallejo, 2012)

Las TIC aportan al aprendizaje por proyectos los siguientes elementos . (Vallejo, 2012):

- Procurar la comunicación con los compañeros, la comunidad educativa, personas de otros países y en situaciones muy distintas a las nuestras, facilitando romper barreras geográficas, de idiomas, etc.
- Permiten la creación de redes y entornos personales de aprendizaje.
- Facilitan el acceso a la información y a una amplia gama de recursos.
- Ofrecen gran variedad de experiencias de aprendizaje.
- Ponen a nuestra disposición herramientas para la preparación, organización, seguimiento y presentación de los proyectos.

- Aportan herramientas concretas para la elaboración de proyectos colaborativos

5.1.2. Método de Aprendizaje Basado en Problemas

Este es un método de aprendizaje basado en el uso de problemas como inicio para la adquisición de nuevos conocimientos. Es entonces, una metodología concentrada en la investigación y reflexión sobre la resolución de problemas, planteados en principio por un tutor. (Barrows, 1986).

El aprendizaje basado en problemas muestra que teniendo en cuenta lo que hacen los estudiantes, pueden mejorar su calidad de educación en diversos ámbitos. Las competencias que más se destacan al usar el ABP son (De Miguel, 2005):

• Resolución de Problemas
• Toma de decisiones
• Trabajo cooperativo
• Capacidad de análisis al evitar el ensayo y error.
• Habilidades de comunicación (argumentación y presentación de la información)
• Desarrollo de actitudes y valores: precisión, revisión, tolerancia.

5.1.3 Método del Aprendizaje Activo

Este método no es más que permitirle al niño que aprenda haciendo. El tutor en este tipo de aprendizaje, se convierte en una guía, dejando que el mayor esfuerzo lo hagan los niños.

Con el aprendizaje activo se puede garantizar que tanto el niño como el tutor van participar, puesto que se tiene que conocer primero el tema, reflexionar y realizar una construcción de su propio conocimiento. El tutor en este caso, debe cuestionar al estudiante en la marcha y corregirlo, para buscar una

retroalimentación efectiva. En numerosas ocasiones los niños comprenden la teoría pero no están capacitados para aplicar estos conocimientos.

Entre las habilidades de orden superior que se promueven son:

- Análisis,
- Síntesis,
- Evaluación,
- Interpretación,
- Inferencia.

5.2. TEMAS APLICADOS EN LOS NIVELES DEL VIDEOJUEGO

Como introducción al conocimiento del proceso básico de la programación, es necesario hacer un repaso por la lógica proposicional, puesto que contiene el rompecabezas de dicho proceso. Para la elaboración de los niveles fue necesario utilizar lenguaje formal con situaciones reales, como ejemplo, si se dice: «Caver sale de paseo con Nícolas», esta proposición queda simbolizada en el lenguaje formal mediante la variable p o q, o r, o s.

Además de estas variables, la lógica proposicional utiliza otros símbolos, llamados constantes, cuyo significado siempre es el mismo, ya que modifican o unen a las variables. Estos símbolos constantes se llaman operadores lógicos y pretenden simbolizar lo juicios. (Enderton, 1972)

El videojuego comprende 3 niveles importantes con la participación de estas constantes. En el primer nivel se tiene como tema, la lógica condicional a partir de situaciones cotidianas, para que el niño elija. Como ejemplo: Si los personajes están con ropa abrigada, entonces van a la nieve.

En el segundo nivel los niños deben hacer uso del conector lógico conjuntivo en lenguaje formal. Como ejemplo: la rueda de piedra es de Caver y la llanta de Nícolas. Este nivel resulta correcto si los objetos de Caver y Nícolas, pertenecen correctamente a cada uno.

En el tercer nivel se utiliza el complemento de varios operadores lógicos, para lograr sobrepasar obstáculos. Por ejemplo, una proposición de este nivel sería: Si primero camino y luego salto, entonces sobrepasaré la roca.

En este nivel la lógica proposicional se puede hacer con la inferencia lógica de proposiciones a partir de proposiciones, sin tener en cuenta la estructura interna de las proposiciones más simples.

5.3. SELECCIÓN DE LA METODOLOGÍA DE ENSEÑANZA

Aprendizaje Activo

Esta metodología permitiría el análisis de los niveles y la síntesis de la historia a través de la guía y el conocimiento propio. Además, le otorga entrar en participación al tutor, que será quien cuente la historia y explique la forma en que se juega. Como valor agregado se podrán presentar temas de otras disciplinas, como la prehistoria básica, brindándole al docente la opción y responsabilidad de convertirlo en una muestra complementaria.

Por otro lado, el método activo genera autodisciplina y esfuerzo voluntario, al generar interés jugando; algo que se busca al nombrar la palabra ‘incentivar’ en los objetivos específicos. El rol del tutor en este caso será el de una guía, que utiliza una herramienta para buscar el progreso del niño por sí mismo.

En teoría, Jean Piaget (1981), autor guía para este documento, plantea que es el docente, quien debe orientar el proceso de enseñanza y debe conocer, por

su experiencia, las habilidades a requerirles a los estudiantes según el nivel en que se desempeñe, por ende debe incitar nuevas formas de pensar mediante la solución de ciertos problemas. En síntesis, los objetivos de la educación en general son: crear hombres ingeniosos, inventores, y formar mentes que analicen, comprueben y refuten lo que se expone. En consecuencia es necesario formar niños activos, que aprendan pronto a investigar por su cuenta, teniendo siempre presente que las adquisiciones y descubrimientos realizadas por si mismo son mucho más enriquecedoras y productivas. (Piaget, Inhenler, 1981)

También se han tenido en cuenta el juego y su clasificación a partir de las teorías de Piaget (1961). La clasificación es: el juego de ejercicio, que no tiene un carácter lúdico; los juegos simbólicos y los juegos con reglas. El presente videojuego se basa primero, en el juego simbólico, donde el niño asimila la realidad y la incorpora para revivirla, dominarla o compensarla. Pero es realmente el juego con reglas lo que estructura el prototipo, puesto que se limitan las posibilidades para realizar las cosas y un solo camino aparente. Según Piaget (1961) desde los 6 a los 11 años, el juego adquiere una dimensión más social y las reglas entre jugadores se hacen más complejas, requiriendo una representación simultánea y más abstracta por parte de ellos.

Por todo lo anterior, el aprendizaje activo es más acorde con la propuesta de este proyecto, pues involucra todos los elementos necesarios, de acuerdo a las necesidades, para una eficiente experiencia de juego con los temas a evaluar.

CAPÍTULO 6
DESARROLLO DEL PROYECTO

6.1. SELECCIÓN DE LA METODOLOGÍA PARA EL DESARROLLO

El Modelo de Proceso de la Ingeniería de la Usabilidad y la Accesibilidad consiste en una serie de actividades bien organizadas que a grandes rasgos podemos clasificar como:

- Actividades estructuradas de los análisis de los requisitos de la usabilidad
- un conjunto de actividades explícito de objetivos de usabilidad
- actividades de soporte a una aproximación estructurada del diseño de la interfaz de usuario
- actividades de evaluación de los objetivos de usabilidad mediante iteraciones –hacia dichos objetivos– en el diseño

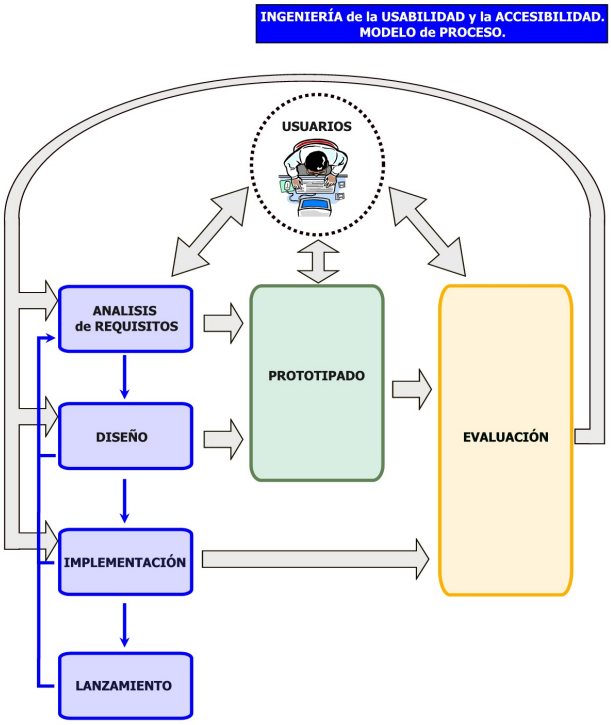

Figura 4. Ingeniería de la usabilidad y la accesibilidad. Modelo de proceso

Análisis de requisitos

Se estipulan los requisitos que determinan qué debe hacer el sistema y cómo debe hacerlo. Los tipos de requisitos suelen ser:

Funcionales, que describen una funcionalidad o un servicio del sistema y no funcionales, que suelen ser restricciones al sistema (p. e. tiempo de respuesta) o para su proceso de desarrollo (utilizar un determinado lenguaje).

Análisis Etnográfico.

Es la forma en que un usuario hace sus actividades en un contexto y un ambiente específico y como usa herramientas para lograr sus objetivos, además permite identificar las relaciones que hay entre las personas y las herramientas, cómo las usan, qué conocimiento tienen para usarla, como es la percepción de su mundo cuando van a realizar una tarea o actividad.

Perfil de Usuario.

Esto nos servirá para poder tomar decisiones a la hora de diseñar la interfaz de usuario y también a identificar las categorías de usuarios que deberán tratarse en el Análisis Contextual de Tareas.

Análisis Contextual de Tareas.

No se trata aquí de realizar el análisis de las tareas sino de determinar todas las tareas que el sistema es capaz de realizar actualmente, o sea, antes de empezar a implementar el nuevo sistema (entendiendo como sistema no al sistema interactivo sino a la manera con sus métodos y herramientas que actualmente se realizan las cosas).

Actores.

Es muy importante para poder modelar tareas poder identificar los diferentes tipos de actores relevantes que intervienen en el sistema.

Roles.

Los roles indican clases de actores los cuales tienen asignados ciertos subconjuntos de tareas, ya sea por elección propia o como resultado de la organización.

Organización. La organización describe la estructura jerárquica y de delegación de responsabilidades entre roles, así cómo el papel de los actores en los diferentes roles.de tareas. Los objetos pueden ser cosas físicas o conceptuales, como mensajes.

Objetos. Cada cosa que sea relevante en el trabajo en una cierta situación es un objeto en el sentido del análisis os, passwords, firmas, etc.

Plataforma (posibilidades/restricciones). Este punto va directamente relacionado con la plataforma tecnológica escogida para albergar el producto.

Perfil del Entorno. El entorno donde se realiza un determinado trabajo suele influir muy directamente en la manera en que este se realiza.

Objetivos de la Usabilidad.

- ✓ facilidad en el aprendizaje
- ✓ consistencia
- ✓ flexibilidad
- ✓ robustez
- ✓ recuperabilidad
- ✓ tiempo de respuesta
- ✓ adecuación a las tareas
- ✓ disminución de la carga cognitiva

Objetivos de la Aplicación. A la hora de recoger los objetivos de la aplicación deberemos tener en cuenta tanto los requisitos funcionales como los no funcionales (tiempos de respuesta, utilización de un determinado lenguaje de programación,

Diseño

La finalidad esencial del diseño de sistemas interactivos es el proporcionar el soporte necesario a las personas en su trabajo diario, lo cual indica la importancia de este proceso de desarrollo de cualquier aplicación y su importancia. Por tanto tiene que tenerse en cuenta desde el principio.

Deberá empezarse con una idea clara de cómo queremos la interfaz del usuario y como serán las interacciones con este para después, desarrollar las especificaciones funcionales que sirvan de guía al diseño posterior.

Actividades del Análisis de Requisitos

Análisis de Tareas. Basándose en todos los requisitos obtenidos en la fase anterior y los objetivos obtenidos de la misma se rediseñarán las tareas del usuario para racionalizar la organización del trabajo y explotar las capacidades que la automatización proporciona.

Modelo Conceptual.

El Modelo Conceptual puede entenderse como una descripción del sistema propuesto en términos de un conjunto integrado de ideas y conceptos sobre lo que este debe hacer, como debe comportarse y como debe parecer, que será comprensible por los usuarios de la forma en que se ha propuesto [PRS02].

- Definir Estilo. El propósito de esta tarea o fase es, cómo su nombre indica, el definir un estilo que garantice la coherencia visual y funcional de toda aplicación.

- Definir Estilo: Estándares Generales. Existen una serie de estándares generales –la mayoría de ellos vienen por imposición industrial– que son principios y guías que se deben seguir.

- Definir Estilo: Metáforas. El termino metáfora normalmente va asociado al lenguaje y hace referencia a cuando se utiliza una palabra que expresa literalmente una cosa para manifestar otra que tenga cierto parecido con aquella.

- Definir Estilo: Colores.
- La buena selección de la combinación de colores ayuda a disponer de una interfaz de usuario agradable; también las funcionalidades principales y/o críticas serán mayormente accesibles dependiendo de los colores utilizados.

- Definir Estilo: Estándares Particulares.
- Este apartado recoge aspectos particulares del cliente final de la aplicación. Suele tener mayor importancia cuando este se trata de corporaciones en lugar de usuarios concretos, pues estas disponen de elementos que les suele caracterizar –logotipo de la empresa, colores corporativos, simbología adaptada de otras aplicaciones existentes, argot propio, etc.–.

- Este apartado incluye las metáforas y colores que provienen no de un trabajo de inspección de campo sino por el hecho del destinatario de la aplicación. Una misma aplicación podría servir para dos contextos

diferentes cambiando ciertos elementos que utilizamos en un contexto u otro podrían incluso significar funcionalidades totalmente opuestas.

Diseño Detallado. Esta fase es el resultado de la evolución lógica de las fases anteriores una vez prototipadas y evaluadas como mínimo una vez –aunque el número de iteraciones necesarias para definir esta fase dependerá enormemente de la mayor o menor magnitud del proyecto–.
Se procede a realizar un diseño de la interfaz que recoge todos los detalles aglutinados en tareas anteriores y con el mayor detalle posible hasta disponer de una versión que dispone de todos los mismos detalles que la versión software definitiva.

Prototipo

El concepto de prototipado engloba todas las herramientas que permiten realizar a los diseñadores de sistemas estas simulaciones.

Actividades del Análisis de Requisitos

- Herramientas de Diagramación.
- Narrativa. Una historia completa de la interacción hecha con la existente o con un diseño nuevo
- Flowchart. Una representación gráfica de las series de acciones y decisiones extraídas de la narrativa
- Texto procedural. Una descripción paso a paso de las acciones del usuario y las respuestas del sistema.
- Vídeos. El prototipo en vídeo puede ser muy útil en el diseño de interfaces multimodales en el que por ejemplo se realiza una interacción por voz o en el diseño de escenarios futuros de los que todavía no se dispone de la tecnología.

- Prototipos Software. Los prototipos de Software son primeras versiones de ciertas funcionalidades del sistema las cuales ya son realizadas con el lenguaje de programación escogido para desarrollar la aplicación.

Evaluación

El prototipo es una herramienta muy útil para hacer participar al usuario en el desarrollo y poder evaluar el producto ya en las primeras fases del diseño (modelo del ciclo de vida basado en prototipos).

Actividades del Análisis de Requisitos

- Inspección. El término inspección aplicado a la usabilidad aglutina un conjunto de métodos para evaluar la usabilidad en los que hay unos expertos conocidos como evaluadores que explican el grado de usabilidad de un sistema basándose en la inspección o examen de la interfaz del mismo.

- Indagación. El proceso de indagación trata de llegar al conocimiento de una cosa discurriendo o por conjeturas y señales.

- Test. En los métodos de usabilidad por test usuarios representativos trabajan en tareas utilizando el sistema

Implementación

El apartado o fase de la implementación agrupa todo el trabajo de codificación de la aplicación que hasta este punto se ha ido construyendo.

Llegados a este punto, a groso modo, es cuando debe empezarse a programar, lo cual implica haber escogido el o los lenguajes de programación que mejor se

adapten a nuestro proyecto, las bases de datos correspondientes que se precisen, la tecnología que garantice el éxito, etc..

Lanzamiento

La fase de lanzamiento de todo proyecto, sea interactivo o de otra índole, suele ser una de las mas críticas de todo el proceso.

.

Resumiendo, podemos indicar que el éxito total del producto dependerá de dos factores muy importantes: Por un lado que el usuario se sienta cómodo con el sistema. Entendiendo cómo sentirse cómodo el que no le dé errores, que no le resulte complicado usarlo, que recuerde fácilmente donde están las diferentes opciones y sus funcionalidades, etc., y por otro lado que los responsables obtengan los resultados esperados.

Por lo indicado anteriormente podemos ver que en esta fase el factor más importante es lo que se suele conocer como “User Feedback” (“reacción o realimentación del usuario”).

Incrementar el autoaprendizaje en cuanto a la usabilidad (toda nueva experiencia supone un incremento en cuanto a conocimientos ya sean nuevos o mejoras de los ya adquiridos).

6.2. OTROS MODELOS DE DESARROLLO

6.2.1. RUP (Metodología Rational Unified Process)

«Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia.

moEl equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar.» [37]

Ing. MILAGROS, de la Escuela de Ingenieria, define RUP, mediante las siguientes características

La forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo. (Quién hace qué, cuándo y cómo).
Un cambio en las etapas de vida del sistema incrementaría notablemente el costo.
Es un marco del proyecto que describe una parte del proceso que son interactivos.

Requiere un grupo grande de programadores para trabajar con esta metodología.

RUP es un marco del proyecto que describe una clase de los procesos que son iterativos e incrementales. Define un manojo entero de las actividades y de los artefactos que usted necesita elegir de para construir el propios, proceso individual.

RUP es el proceso de desarrollo más general de los existentes actualmente.

Los procesos de RUP estiman tareas y horario del plan midiendo la velocidad de iteraciones concerniente a sus estimaciones originales. Las iteraciones tempranas de proyectos conducidos RUP se enfocan fuertemente sobre arquitectura del software; la puesta en práctica rápida de características se retrasa hasta que se ha identificado y se ha probado una arquitectura firme.

[37] Consultado en¨ www.usmp.edu.pe/publicaciones. Boletin.

RUP proporciona muchas ventajas sobre XP le da énfasis en los requisitos y el diseño.

La ventaja principal de RUP es que se basa todo en las mejores prácticas que se han intentado y se han probado en el campo. (en comparación con XP que se basa en las prácticas inestables que utilizaron juntas se evita que se derribe). Define un manojo entero de las actividades y de los artefactos que usted necesita elegir para construir sus el propios, proceso individual.

RUP es el proceso de desarrollo más general de los existentes actualmente.

Los procesos de RUP estiman tareas y horario del plan midiendo la velocidad de iteraciones concerniente a sus estimaciones originales. Las iteraciones tempranas de proyectos conducidos RUP se enfocan fuertemente sobre arquitectura del software; la puesta en práctica rápida de características se retrasa hasta que se ha identificado y se ha probado una arquitectura firme.

RUP proporciona muchas ventajas, siendo la principal que se basa todo en las mejores prácticas que se han intentado y probado en el campo.

Se divide en cuatro fases:

- ✓ Inicio (Define el alcance del proyecto)
- ✓ Elaboración (definición, análisis, diseño)
- ✓ Construcción (implementación)
- ✓ Transición (fin del proyecto y puesta en producción)

RUP define nueve disciplinas a realizar en cada fase del proyecto:

- o Modelado del negocio
- o Análisis de requisitos

- Análisis y diseño
- Implementación
- Test
- Distribución
- Gestión de configuración y cambios
- Gestión del proyecto
- Gestión del entorno

El proceso define una serie de roles:

Los roles se distribuyen entre los miembros del proyecto y que definen las tareas de cada uno y el resultado (artefactos) que se espera de ellos.

Todos los miembros del equipo comparten:
1 Base de conocimiento
1 Proceso
1 Vista de cómo desarrollar software
1 Lenguaje de modelamiento (UML)

RUP
Realiza un levantamiento exhaustivo de requerimientos.
Busca detectar defectos en las fases iniciales.
Intenta reducir al número de cambios tanto como sea posible.
Realiza el Análisis y diseño, tan completo como sea posible.
Diseño genérico, intenta anticiparse a futuras necesidades.
Las necesidades de clientes no son fáciles de discernir.
Existe un contrato prefijado con los clientes.

6.2.2. Modelo Secuencial

Se tiene un modelo que lleva un desarrollo incremental, esto nos dice que se desarrolla el software en etapas y que después del termino de una etapa no es posible regresar a ella, este modelo tiene cuatro etapas que son:

Planificación: Se determinan los objetivos, metas, requerimientos y restricciones en el proyecto

Análisis de riesgos: identificación de situaciones inconvenientes para evitarlas y solucionarlas.

Ingeniería: desarrollo del producto con respecto al diseño y otras consideraciones planteadas.

Evaluación del cliente: valorización de los resultados del proyecto(producto obtenido).

Se tiene implícita la realización de estas actividades:

-Análisis de requerimientos: En esta etapa se procede a recopilar todos los requisitos que debe cumplir el software a desarrollar, el cliente aqui tiene un papel fundamental, ya que analiza junto con los desarrolladores del software los requisitos que debe cumplir el software a desarrollar.

-Diseño: esta etapa esta enfocada hacia el desarrollo de un esbozo de la arquitectura, la interfaz y el manejo de datos del software.

- Generación de código: es cuando se implementa el diseño del software en algún lenguaje de programación definido en el mismo diseño.

- Pruebas: se hace una revisión de los procesos que realiza el software, para determinar que este haciendo lo planteado para cumplir con los requerimientos.

- Mantenimiento: se verifica el correcto funcionamiento del software en su entorno de uso, y si existen errores o defectos, proceden a corregirlos[38]

---

38 http://members.fortunecity.es/odi39/modelo.htm

CAPÍTULO 7.
DESARROLLO DEL VIDEOJUEGO MEDIANTE EL MODELO ESCOGIDO.

El modelo de desarrollo escogido es el Modelo de Desarrollo de Ingeniería de la Usabilidad y la accesibilidad debido a que sus etapas se centran más en el usuario. Algo que eso se busca ya que uno de los objetivos es que se enseñe de manera indicada a los usuarios.

7.1 ANÁLISIS DE REQUISITOS

Se estipulan los requisitos que determinan qué debe hacer el sistema y cómo debe hacerlo. Los tipos de requisitos suelen ser:

Funcionales, que describen una funcionalidad o un servicio del sistema y no funcionales, que suelen ser restricciones al sistema (p. e. tiempo de respuesta) o para su proceso de desarrollo (utilizar un determinado lenguaje).

Actividades del análisis.

- **Análisis Etnográfico:**

Observación, comportamientos culturales e interacciones: Acercamiento a los usuarios, seleccionando una muestra de estudiantes de un colegio bilingüe, que comprende el rango de estratos: 3, 4 y 5 de la ciudad de Cali, la observación se hizo duran la clase de informática.

En primera instancia se ha concertado en la forma como interactúan los niños en espacios académico, se pueden ver en los laboratorios de clases a niños utilizando sus computadores normalmente; sin embargo, la concentración es de períodos cortos puesto que el profesor tiene que indicarles a todos que deben

hacer en el computador, y no es el software quien induce el comportamiento de los niños. Los temas que se abarcan en la clase de tecnología son la lógica y formulación de problemas mediante fichas lúdicas, los docentes de informática utilizan el juego **Cat-Shmat** para la resolución de problemas lógicos. En este juego realizado en *Adobe Flash* el objetivo principal es entregarle la moneda al gato (Ver figura 5) ; entendiendo como se construye camino con los elementos proporcionados, para lograr superar los obstáculos. El problema principal de dicho producto interactivo es que no está acorde con las necesidades del usuario: no brinda un tutorial que explique la función de cada elemento, ni son intuitivos; además los niveles son de una dificultad para un desempeño cognitivo mayor.

Figura 5. Nivel 6 de Cat-Shmat

El siguiente análisis etnográfico es realizado a una muestra total de 30 niños de un colegio de la ciudad de Cali, en el rango de edad de 7 a 11 años. Las preguntas fueron respondidas originando por parte del interrogador, acciones e interacciones para causar la realidad del niño.

Género:

Niños: 41%

Niñas: 59%

Acceso a computador:

Si: 89%

No: 11%

Acceso a computador con internet:

Si: 91%

No:8%

Gusto por videojuegos web:

Si: 76%

No: 24%

Gusto por juegos similares en propósito:

Si: 68%

No: 32%

Duración promedio en el nivel de atención en más de 3 niveles en un videojuego:

2 minutos.

Perfil del usuario: Es propicio mencionar que para realizar un producto, se debe identificar qué perfil es el usuario para saber lo que comparten y lo que los diferencia.

A continuación se hará una descripción detallada con cada módulo:

- **Descripción general:** Niños de 7 a 11 años de edad, de cualquier estrato social. Obligatoriamente estudiantes de colegio. Niños, con acceso a computador e internet. Niños con gusto por los videojuegos.

- **Nivel de estudios:** Grado segundo de primaria a quinto de primera básica. Conocimientos en el idioma español, leer.
- **Características relevantes para el test:** Que tengan experiencia intermedia-baja en manejo del mouse, manejo de ventanas, identificación videojuegos sencillos para pc.
- **Características concretas que comparten y sean relevantes para el test:** Experiencia autónoma en manejo de videojuegos, como mínimo que todos estén familiarizados con ventanas de aplicaciones.

Análisis contextual de tareas: La prioridad ha sido la de notar qué patrones de trabajo utilizan. Según los usuarios anteriormente mencionados, las tareas que realizan en la actualidad, son llevadas a cabo con dispositivos electrónicos, buscando diversión, sencillez y rapidez en el proceso de llevar a cabo una tarea.

- Por lo general no leen manuales, ni el tutorial de ayuda.
- Se fijan poco en las cinemáticas de los videojuegos.
- Su metodología es más de ensayo y error, por su edad.
- No concluyen un proceso, si requiere demasiadas tareas para llegar al objetivo.
- Para leer fluido, deben ser palabras de su cotidianidad.
- Identifican botones en una nueva aplicación, por el estándar que se ha marcado a la hora de hacer los íconos. Por ej., el ícono de play de la mayoría de videojuegos.
- Les gusta recibir más que dar.
- Se animan a utilizar una aplicación si la interfaz de usuario es llamativa, novedosa, sin embargo cuando es tan saturada, inmediatamente pierden un objetivo claro.

Objetos: Los más relevantes de la aplicación serán botones, personajes, stages, fondos, zonas de arrastre, consola de opciones,

Plataformas: La plataforma principal es computadores con navegador y acceso a internet.

Lenguaje de programación.

- **HTML:**

Es el lenguaje usado desde sus inicios por la World Wide Web, como sus siglas lo indican, significa HyperText Markup Language y consiste básicamente en un lenguaje para traducir la estructura textual de páginas y contenido, como imágenes y videos.

HTML ha estado en proceso de evolución desde que fue incluido en Internet en la década de 1990, hasta ahora, HTML5 es la quinta versión de este lenguaje y contiene cambios en la sintaxis en comparación a la cuarta versión, convirtiéndose en el futuro de la web y las aplicaciones para dispositivos móviles, con nuevas etiquetas que evitan ambigüedades en el marcaje, siendo más fácil y simple, con el propósito de plasmar una web orientada al DOM . Es importante mencionar que la estructura básica de un sitio web ha cambiado, anteriormente la etiqueta div asignaba los elementos de esta, pero ahora el mismo elemento posee, en algunas ocasiones, su propia etiqueta.

Para continuar con la semántica, presentamos elementos importantes de este lenguaje:

§ audio y video: Este es sin duda unos de los complementos más interesantes en la nueva versión puesto que ya no es necesario utilizar módulos de Flash.

§ embed: Usado para contenidos de plugins, tales como el de Flash.

§ canvas: Es un elemento avanzado que permite dibujar dentro de él y renderizar gráficos dinámicos sobre la marcha, para nuestro proyecto es un elemento de suma importancia para la acceder a los píxeles de la imagen.

JavaScript:

JavaScript, abreviado como JS, es un lenguaje interpretado ligero, orientado a objetos con funciones de primera clase, famosas y catalogadas como el lenguaje de programación para páginas web, pero también utilizado en muchos entornos sin navegador como node.js o Apache CouchDB .

El estándar de JavaScript es ECMAScript, a partir de 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1. Los navegadores más antiguos toleran al menos ECMAScript 3.

- **Enchant.js:**

Es un motor simple hecho en Javascript para crear juegos y aplicaciones in HTML5 + JavaScript que fue creado en 2011, pero que ya cuenta con más de r de 1.000 juegos y aplicaciones publicadas. La licencia es de código abierto (bajo licencia MIT), y por ello se tiene total libertad de uso. Brinda una extensa documentación, además es posible aumentar su funcionalidad con otros plugins desarrollados. Como gran ventaja se puede recalcar que está en continuo desarrollo y es mantenida por los miembros del Centro de Investigación en Akihabara UEI. Las características principales son:

- **Posibilidades:** La garantía de tener siempre en cualquier lugar la aplicación, el uso tiene la contingencia de ser masivo. Versatilidad. Cualquier sistema operativo, cualquier dispositivo móvil. Rapidez en el tiempo de respuesta. Liviano.

- **Restricciones:** Depender de una conexión a internet. Las gráficas no deben ser de tamaños muy grandes porque se exigiría mayor velocidad de conexión; por ende las gráficas no pueden ser de alto rendimiento, como en una consola de videjuegos, por ejemplo.

Perfil de entorno: El entorno donde se realiza la actividad puede darse en espacios donde la concentración se pueda llevar a cabo, en aula de clase, biblioteca o en casa. Aunque los niños pueden ser demasiado dispersos en compañía, la aplicación es desarrollada para que la carga cognitiva pueda manejarse mientras se está con ruido y acompañado.

Objetivos de la usabilidad:

- **Facilidad en el aprendizaje:** El tiempo que debe tardar el usuario en conocer bien la aplicación, debe ser mínimo. A futuro, se propondrán cinemáticas de ayuda, de máximo 30 segundos de duración cada una. Para construir la facilidad se evocará el sistema como **Sintetizable**: el usuario debe presenciar cambios de la parte inicial de la tarea, al resultado final de esta; y **Familiar**: Debe correlacionar los conocimientos previos adquiridos y la contextualización del mundo real. Por ejemplo, hacer que las prendas de vestir tengan que ver con la ocasión.
- **Flexibilidad:** Elegir al usuario como controlador del sistema dándole un rol y dejándolo intentar cuantas veces quiera, posibilidad de corroborar errores en la parte de arrastrar y soltar las opciones, puede retractarse y cambiar el orden. Brindar un puntaje y una validación de lo que se está haciendo.
- **Consistencia:** Tener una misma línea en la forma en que se realizan los procesos, en el diseño de la aplicación, generar guías y estándares. Como ejemplo, la misma tipografía, la misma forma de ilustración de los personajes, de las situaciones. Paleta de colores establecida para las diferentes partes de la estructura de la interfaz.

- **Robustez:** Permitir a los usuarios, poner sus opciones como les parezca, y hacer sus tareas sin interrupciones.
- **Recuperabilidad:** El usuario podrá recibir inducciones para corregir una acción mal hecha, como sería el caso de dejar alguna zona de arrastre sin su personaje.
- **Tiempo de respuesta:** Cortos, tolerantes a la relación con el tiempo que tienen. Deben poderse ver resultados en no más de 6 clics. Cuando se entra, no hay necesidad de registrarse y crear un usuario, y así hace que los niños puedan utilizarla por sí solos. Para , para publicar un estado de ánimo con una canción, se necesita sólo seleccionar el color, luego seleccionar la sección compartir; escribir una descripción opcionalmente, seleccionar la canción y finalmente publicar.
- **Adecuación a las tareas:** Cambiar de estado de ánimo en un toque. Permitirle comentar, dar un reconocimiento, ver sus contactos, publicar varias veces. Darle todo lo que al que alrededor de la aplicación se puede hacer, en el momento que quiera y como desee.
- **Disminución de la carga cognitiva:** Si algo debe tener la aplicación es ser sutil y comprensible, tener estrictamente lo necesario. Evitar descripciones a los botones, sino asociarlo a formas, para mejor recordación. Manejar la asociación de formas y objetos similares con los significados.

Objetivos usabilidad del videojuego: Desarrollar los procesos de las principales tareas, como iniciar el videojuego, arrastrar las opciones, comprobar las opciones, mostrar el cuadro de puntaje. Hacer de esta un videojuego rápido, innovador y sobre todo, fácil de usar. Llevar a cabo la praxis de la carrera Ingeniería Multimedia y el curso de Usabilidad y HCI. Cumplir con la motivación de cambiar la forma en que se aborda el aprendizaje de la lógica y el proceso básico de la programación en la primera infancia.

7.2 DISEÑO

Análisis de tareas: En la siguiente descripción de las tareas, se encuentran las que tienen mayor relevancia.

- **Iniciar el videojuego:**

  Objetivo: Empezar a jugar
  Tarea: Presionar el ícono de iniciar.
  Acciones: Clic en el botón de *play* para iniciar el juego.

- **Seleccionar Nivel:**

  Objetivo: Escoger un nivel
  Tarea: Entrar en el nivel.
  Acciones: Clic en el botón del nivel uno para iniciarlo

- **Jugar nivel 1**

  Objetivo: Poner correctamente los personajes a su debida ocasión.
  Tarea: Analizar las 4 ocasiones, arrastrar los personajes con el atuendo indicado para cada ocasión.
  Acciones: Clic mantenido para arrastrar a los personajes y soltar en la zona de la ocasión correcta.

- **Jugar nivel 2**

  Objetivo: Ubicar correctamente los objetos asociados al armario de cada personaje.

Tarea: Poner cada elemento en el lugar indicado, los lados deben tener los objetos asociados correctamente

Acciones: Clic mantenido en el objeto, arrastrar al armario indicado, soltar en la zona de la ocasión correcta.

- **Jugar nivel 3**

Objetivo: Caver debe llegar al otro lado del escenario y rescatar a Nícolas de un dinosaurio.

Tarea: Dar las instrucciones para sobrepasar los obstáculos.

Acciones: Clic mantenido en la instrucción, arrastrar al la zona de arrastre de la instrucción, soltar en la zona de arrastre de la instrucción.

Modelo Conceptual: El modelo conceptual es una alternativa de diseño, siendo una descripción del sistema, muy básico. El tipo de modelo conceptual al que se llega es por medio de la manipulación de los prototipos de papel.

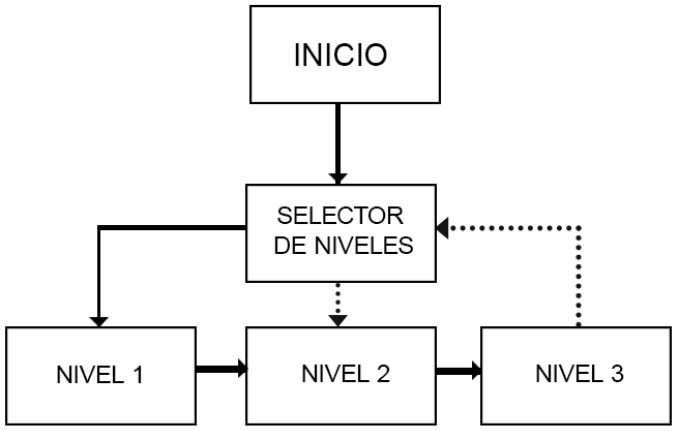

Figura 6. Modelo conceptual

Definición del estilo: Para que el videojuego tenga estándares visuales se plantea la siguiente configuración predeterminada y se siguen lineamientos para tener consistencia en el diseño:

Game Document Design (GDD)

Género: El género del videojuego a desarrollar es *Educativo,* puesto que está dirigido a niños de 7 años a 11 años de edad como herramienta de aprendizaje.

Jugadores: La modalidad de juego es Individual, el usuario interactúa con la máquina.

Historia: Nícolas es un niño que se encuentra en el pasado por alguna razón desconocida, aunque su astucia lo saca de apuros a menudo, esta vez se ve obligado a conocer una región lejana, al parecer prehistórica puesto no que no conoce a nadie en la zona. En uno de esos días soleados, Nícolas caminaba mientras un murmullo extraño proveniente de una cueva, le llamaba la atención. Decidió curiosear en la cueva y tuvo la fortuna de encontrarse con Caver, el más conocido, solamente, como el hombre de las cavernas o sea cavernícola. Esta es la historia de Caver y Nícolas, una pareja de amigos que necesitan de tu ayuda para superar los retos que nacen de sus aventuras, como por ejemplo escoger el atuendo adecuado para determinado clima. ¿Será que Caver y Nícolas son capaces de superar estos retos totalmente solos?. ¡Ayúdalos en esta magnífica aventura!

Nombre: Ayuda a Caver & Nícolas.

Definición de Estilo: para definir la línea gráfica del videojuego en general, se ha tomado como referencia el manual de identidad visual, es decir un documento que tiene los siguientes puntos relevantes:

Inspiración:

Figura 7. Cuevas de la India

La inspiración permite tener una idea de cómo serían las ilustraciones y los colores a partir de imágenes reales. Se ha tenido en cuenta la historia, para la búsqueda de las imágenes. En este caso analizamos las cuevas de Bhimbetka abren sus bocas entre las rocas y peñascos de una colina que sobresale en la llanura de Madhya Pradesh, en la zona sur de la gran meseta central de la India, conservadas hace 2,5 millones de años. [Fullola, Josep Mª; Nadal, Jordi (2005). *Introducción a la prehistoria. La evolución de la cultura humana* (primera edición). Ed. UOC. p. 210.]

Paleta de Colores.

Figura 8. Paleta de colores para el logo y objetos, basadas en las imágenes

Proceso de diseño Logotipo.

Proporción Áurea en Logo: La proporción áurea, es un mosaico de cuadrados cuyos lados son la longitud encantaba de los números de Fibonacci sucesivamente.

La proporción áurea, se basa en el número phi (!) = 1.61803398874... descubierto por el matemático italiano Fibonacci. Estas mismas proporciones son encontradas en la naturaleza para lograr balance, armonía y belleza en sus propias creaciones de arte, arquitectura, color, diseño, composición, espacio e incluso música. Desde el Partenón hasta la Monalisa, desde las pirámides de Egipto hasta las tarjetas de

crédito, phi está en todos lados, siempre. The Golden Section and the Golden Rectangl, Michael Lahanas[39].

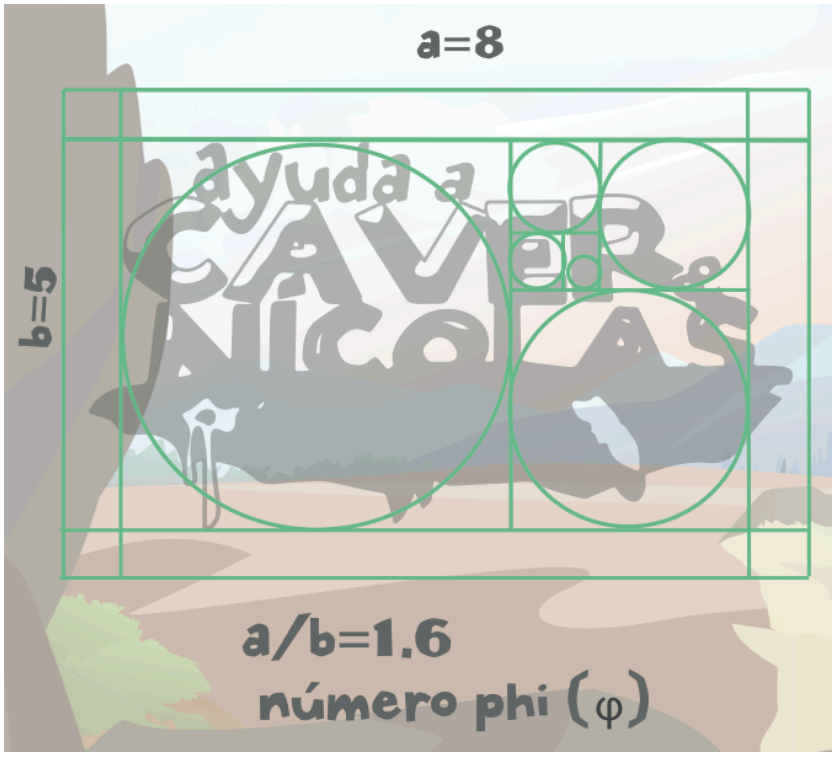

Figura 9. Proporción Áurea en boceto del logo

[39] http://blogmidisena.blogspot.com/2012_09_01_archive.html

Diseño de Logotipo.

Figura 10. Logotipo final

Uso del Logo.

Figura 11. Uso prohibido del logotipo. Ejemplo 1

Figura 12. Uso permitido del logotipo. Ejemplo 2

Prohibido utilizar el logo en tamaño inferior a 10 mm, permitido usar el Usar la tipografía sin el símbolo (Fragmento de Roca)

Figura 13. Uso prohibido del logotipo. Ejemplo 3

Figura 14. Uso prohibido del logotipo. Ejemplo 4

Prohibido utilizar la tipografía separada y con espacios diferentes, y utilizar el logo sin la frase “ayuda a”.

Tipografía.

Tipografía principal:

Figura 15. Caracteres de la fuente principal

Estonge Hinge, fuente gratuita empleada para títulos y subtítulos.

Tipografía complementaria.

Figura 16. Caracteres de la fuente principal empleada para parrafos

Arial Black, fuente empleada para párrafos.

Perfl de los personajes

Caver: Cavernícola o hombre de las cavernas, no sabe hablar, como es un cavernícola, le cuesta razonar algunas cosas.

Nícolas: Es un niño que viaja al pasado por una razón desconocida, tiene 14 años, tiene habilidad para resolver problemas, pero no conoce el lugar donde está.

Construcción de Personajes:

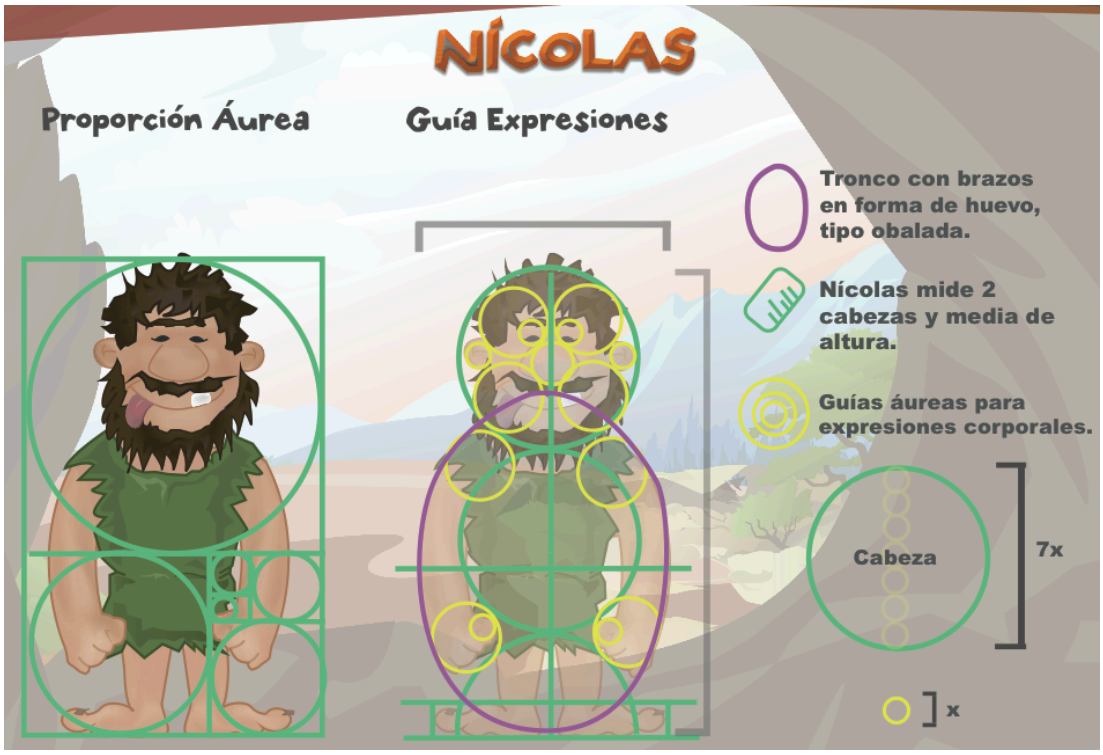

Figura 17. Estructura del personaje principal.

Personaje diseñado con guía de expresiones de proporción áurea.

Items

Figura 18. Items relacionados con el personaje principal y su entorno

Look and Feel (PDF): La cámara se maneja en 2D. Los gráficos, personajes y escenarios se han implementado en 2D también.

Objetivos: El objetivo final es salvar a el personaje Nícolas quien es atrapado por un dinosaurio. Los objetivos por niveles son superar los retos, cómo llegar hasta cierto punto, saltar las piedras, saltar y obtener la soga.

Reglas: No habrán vidas por nivel, si se pierde, debe iniciar de nuevo todo el nivel, no habrá tiempo límite para finalizar el nivel, las combinaciones entre los botones debe ser correcta cuando se presiona Play en algunos casos. El jugador gana cuando el nivel es completado 100%. A medida que pasan los retos la dificultad aumenta..

Requerimientos técnicos: Los requerimientos son un computador con una RAM de mínimo 512MB, que contenga acceso a internet y un navegador instalado.

7.3 PROTOTIPADO

El concepto de prototipado engloba todas las herramientas que permiten realizar a los diseñadores de sistemas estas simulaciones.

Actividades del Análisis de Requisitos.

- Herramientas de Diagramación.
- Narrativa. Una historia completa de la interacción hecha con la existente o con un diseño nuevo
- Flowchart. Una representación gráfica de las series de acciones y decisiones extraídas de la narrativa
- Texto procedural. Una descripción paso a paso de las acciones del usuario y las respuestas del sistema.
- Vídeos. El prototipo en vídeo puede ser muy útil en el diseño de interfaces multimodales en el que por ejemplo se realiza una interacción por voz o en el diseño de escenarios futuros de los que todavía no se dispone de la tecnología.
- Prototipos Software. Los prototipos de Software son primeras versiones de ciertas funcionalidades del sistema las cuales ya son realizadas con el lenguaje de programación escogido para desarrollar la aplicación.

Para esta etapa se escogió el método narrativo acompañado de prototipos de papel. En primera instancia se reunieron a 15 niños para contarles la historia y preguntarles sobre los elementos del juego. La organización fue determinada por los entrevistadores, sin embargo el niño debía aprobar los botones y escoger los

que mejor entendían. Se plantearon los botones: iniciar, ir a menú, escoger nivel, seleccionar nivel, la posición del logo.

Una de las etapas iniciales para el desarrollo del videojuego fue el prototipado, donde se les explicó a los niños la historia y se les preguntó sobre los cavernícolas. En contexto, la mayoría de los niños entrevistados tienen alguna noción sobre estos individuos y cuando se nombra este término, piensan en palabras relacionadas como: hombre, prehistoria, cuevas, montañas, rocas, dinosaurios, “ropa de selva”, era de hielo, campos, animales, fuego.

Al mostrarles el boceto del personaje principal, la mayoría reconoció que la ilustración representa a un cavernícola. Ver figura 19.

Figura 19. Bocetos del personaje principal. Cavernícola

Entrando en materia del diseño de la interfaz, los botones fueron planteados cada uno con 3 opciones propuestas por el autor. En la Figura 14 se pueden ver

los botones de: ‘empezar a jugar’, ‘retroceder o ir al inicio’, ‘cerrar’, ‘aceptar’ y ‘ejecutar o comprobar lo hecho’. Entre las opciones de ‘empezar’, los niños decidieron el apropiado según ellos. Entre las tres opciones la mayoría escogió el botón con el símbolo universal de ’*play’.* Recalcaron que los demás botones eran innecesarios y “aburridos”. Entre las opciones del botón ‘retroceder o ir al inicio’, un poco más de la mitad escogió el símbolo de una cueva, ya que se asemeja a una casa (cuyo significado universal es el de ir a ‘inicio’), no obstante, afirmaron que el botón con la flecha hacia atrás también podría ser apropiado. El botón mejor relacionado a la acción de ‘cerrar’ por los niños fue, el que tenía una ‘x’, argumentando este, por su parecido al botón de cerrar ventanas en el sistema operativo Windows. La mayoría de niños recalcaron que el botón ‘ejecutar’ puede ser usado con cualquiera de las opciones propuestas por el diseñador y que puede ser definido por quien hace el videojuego. Ver figura 20.

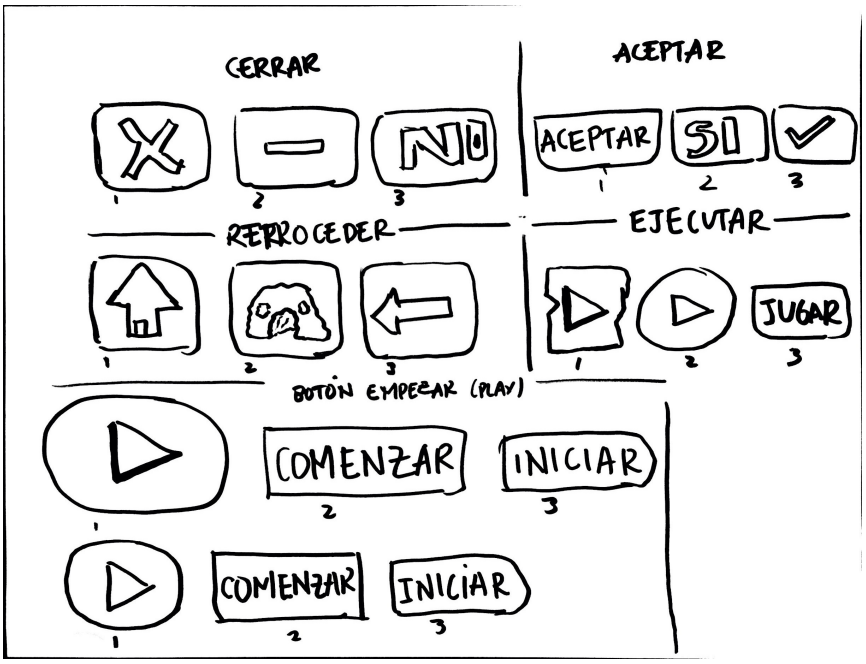

Figura 20. Bocetos de los botones de la interfaz

Para el diseño y la organización de la pantalla inicial del videojuego, se tomaron en cuenta las ideas que tuvieron cuando se les narró la historia. Los niños

indicaron que iniciar dentro de una cueva es de su agrado. Mostraron gusto hacia la propuesta de tener al personaje principal en la primera pantalla. En cuanto a la organización del botón ‘play’ coincidieron en que debería estar en la parte inferior y el logo en la parte central. Ver figura 21.

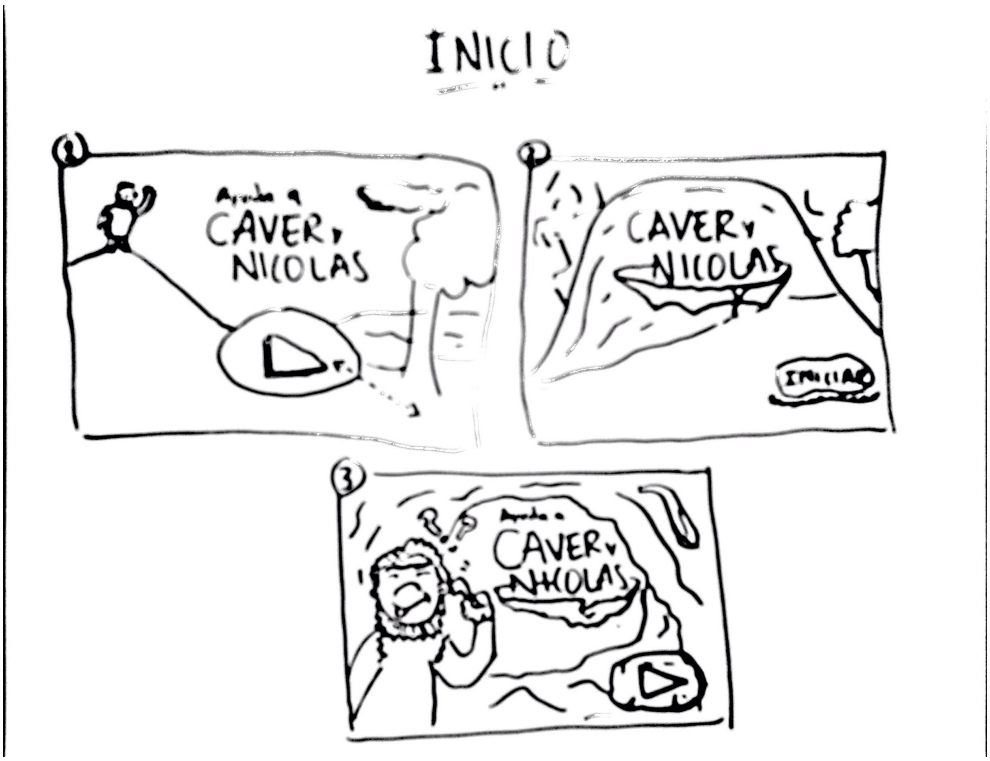

Figura 21. Bocetos opciones pantalla de inicio

Para el Nivel I se les preguntó cuáles son las actividades que un niño moderno junto con un cavernícola pueden hacer; a lo que respondieron: ir a nadar, dar un paseo en el campo, dormir, jugar en la nieve, jugar con rocas, jugar en la caverna, ir al bosque, saltar e ir a los nevados. En cuanto a la cantidad de actividades a las que van a ir Caver y Nícolas dispusieron de 4 en total. Entre las seleccionadas favoritas están: nadar, dormir, dar un paseo en el campo e ir a los nevados. Ver Figura 22.

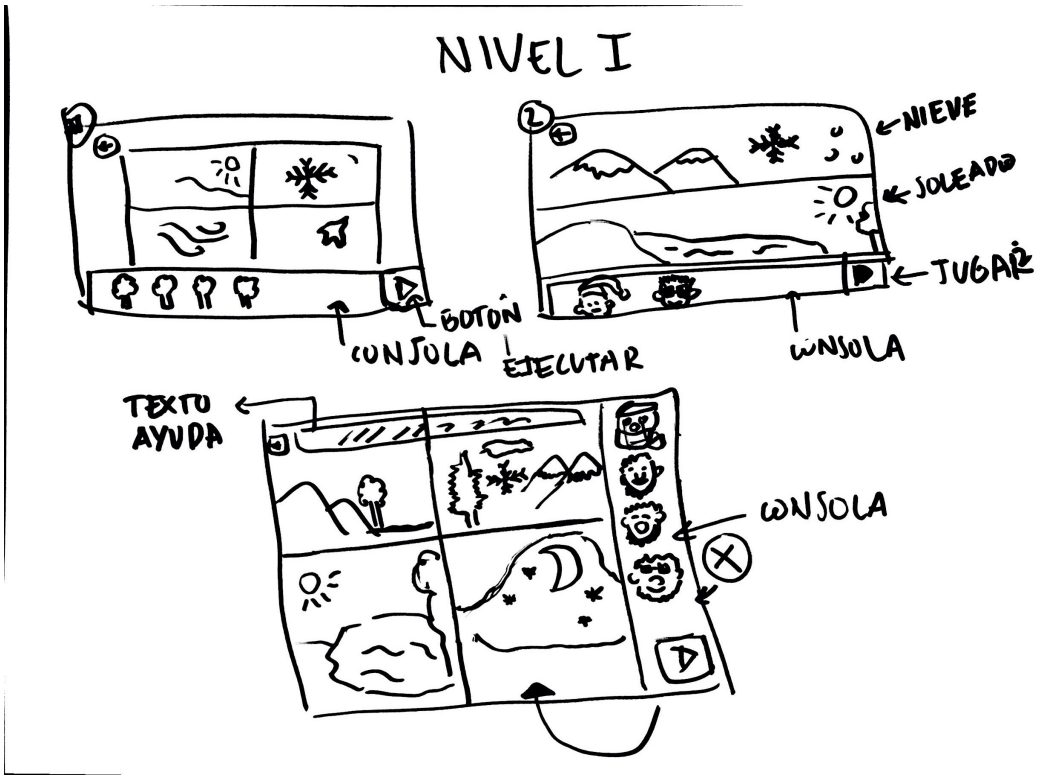

Figura 22. Bocetos del nivel 1

En el Nivel II se hizo una lista con los niños, de los objetos que pueden pertenecer a un niño moderno y a un cavernícola, pero que puede tener relación en las dos épocas. En la sección del cavernícola escogieron: la rueda de piedra, un trozo de roca, un palo con fuego, una prenda hecha con hojas, una liana. Para los objetos del niño expresaron que serían apropiados los siguientes: una llanta, un martillo, una vela, una camiseta y un lazo. Para la organización de las cajas o 'dropzones' donde se sueltan los objetos de cada uno, aseguraron que pueden estar a lados opuestos, y el contenedor de todos los objetos en el área central. El botón de 'probar' o 'ejecutar' debe estar en la parte superior, donde –según ellos- tendría más facilidad de uso. Ver Figura 23.

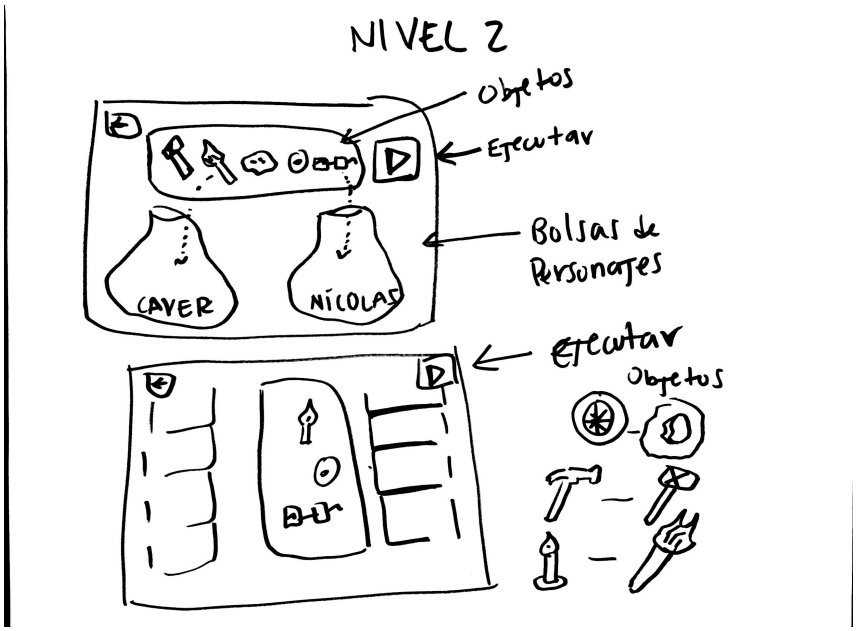

Figura 23. Bocetos del nivel 2 y sus elementos

Con ánimos de concluir la última fase del prototipado, se relató un prefacio de la historia del Nivel III para que, mediante la deducción, los niños conocieran la historia completa y supieran qué debían hacer. Cuando supieron que el personaje iba a ser dirigido por ellos para ganar el nivel, hicieron el ejercicio de imaginarse cuáles serían los pasos para llegar al objetivo final. Los pasos que tenían en mente eran: moverse a la derecha (caminar), saltar la roca, brincar a coger el lazo y pelear. Ver figura 24.

Figura 24. Bocetos del nivel 1 y sus elementos

De igual manera fueron evaluados los botones del nivel III, Los botones para cada paso tienen opciones con símbolos que se relacionan visualmente con la acción a realizar. Aún así, los niños escogieron el estilo de botón que contenía las flechas, esto debido a que –según ellos- asociarlos con las acciones era más practicable que las figuras. Ver figura 25.

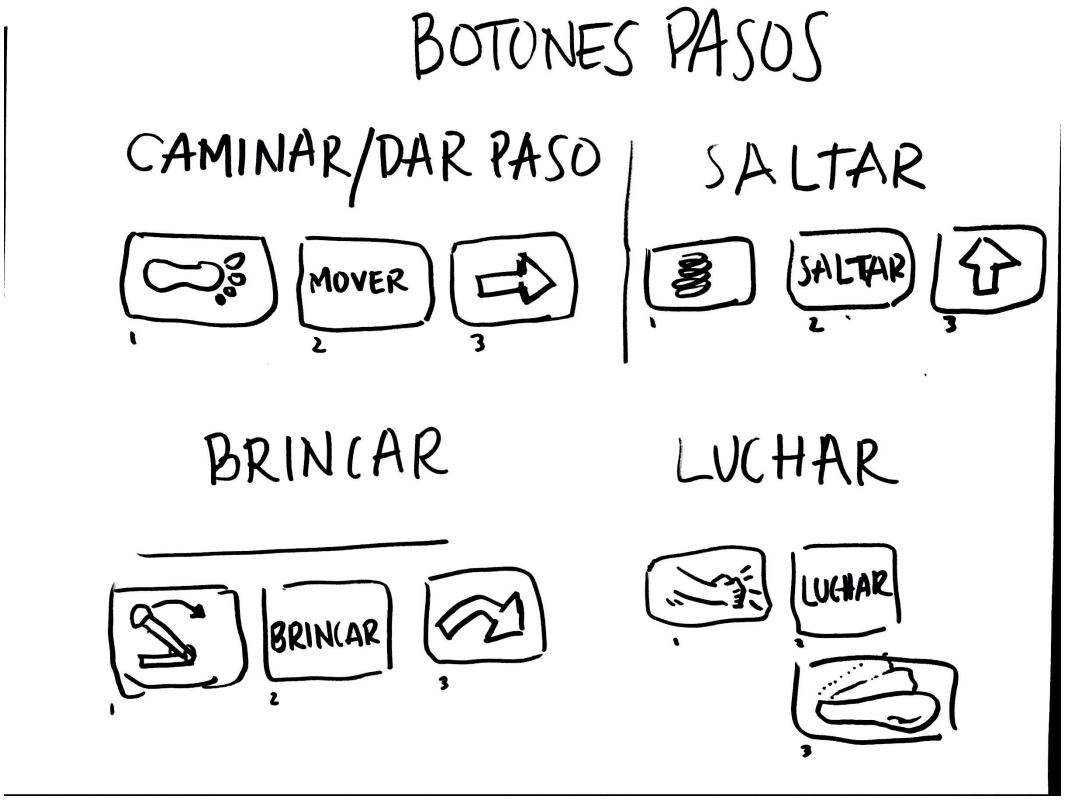

Figura 25. Bocetos de botones del nivel 3

7.4 EVALUACIÓN

7.4.1. Instrumentos de Evaluación

La presente evaluación es llevada a cabo, para identificar la experiencia de los usuarios con el prototipo final del videojuego. Comenzando esta etapa fue necesario concretar la clasificación de evaluación con los siguientes criterios: lugar de realización, tipo de técnica, automatización y participantes.

Refiriéndose al tipo de técnica de comprobación, se escogió el método de evaluación por test, donde los usuarios objetivos manipulan el sistema y entregan resultados cualitativos de cómo la interfaz de usuario ayuda a ejecutar las tareas.

Por el lado de la automatización se decide usar el *método manual* donde, a pesar de tener un prototipo funcional, relucen aspectos “fuera de lo normal” de cada interacción.

Quien interviene en la evaluación es denominado: participante, en este caso se manejó un grupo de 15 participantes, de edades representativas para el proyecto.

Adicional a esto se tienen como implicados directos, a los evaluadores, permitiéndoles captar información valiosa. Por medio de encuestas se conocen y se analizan las opiniones sobre los docentes.

El lugar donde se realizó la evaluación fue en el propio entorno, donde los usuarios realizan sus labores habitualmente, ya que de esta manera se puede ver cómo realizan las tareas en el instante. En aras de obtener una visión amplia de la evaluación ha sido propicio evaluar los usuarios en 2 contextos: escolar y familiar.

7.4.2 Resultados Observación en Contexto General

Contexto Escolar:

Se hizo una minuciosa observación en una clase de tecnología e informática, dictada en un colegio privado de la ciudad de Cali en el grado segundo de primaria, el cual está conformado por 18 niños que oscilan entre los 7 y 8 años de edad. Dicha observación fue realizada con el fin de analizar y conocer la motivación que demuestran los estudiantes por jugar e interactuar con algunas características importantes del videojuego “Ayuda a Caver y Nícolas” como el diseño, el método de enseñanza y su eficacia.

Teniendo en cuenta lo anterior, se evidenció que el docente utiliza una estrategia adecuada para introducir el juego, inicialmente les presenta la historia correspondiente, a través de diapositivas y les realiza preguntas con el propósito

de evaluar su concentración y deseo por jugar, fue visible el ánimo de los estudiantes por empezar con la actividad. Sin embargo, cuando se dirigían a los computadores se analizó un aspecto importante: los niños debían compartir sus computares y esto podría afectar con sus períodos de atención; aunque se demuestra lo contrario, debido a que el trabajo así, promovió aspectos importantes entre pares, como por ejemplo la comunicación de ideas y pensamientos frente a lo que creerían que sucedería con los personajes, el ánimo por querer avanzar, destacándose el trabajo en equipo y la paciencia, al respetar el turno por jugar.

Así mismo, se certificó que los niños estaban adquiriendo conocimientos de manera lúdica y divertida, cabe aclarar que esto se afirma al escuchar las respuestas de los alumnos, puesto que a medida que iban jugando la profesora evaluaba sus procesos de aprendizaje a través de preguntas, brindándoles un acompañamiento positivo y demostrando su comprensión por el objetivo del juego; además los niños resaltaron su gusto por los personajes y les llamó la atención los "colores y dibujos" que presentaba el juego, con lo anterior se evidenció que el diseño fue apropiado y se ajustaba al agrado de los niños evaluados. El diseño de la interfaz facilitó la ejecución de todas las tareas, es decir, que el diseño de la interfaz no presentó problemas.

Contexto Familiar

Con el propósito de evaluar la reacción de los niños con el videojuego "Ayuda a Caver y Nícolas", se realizó una observación en un **contexto familiar**, teniendo en cuenta la motivación de los niños frente a las características principales (diseño, procesos de aprendizaje, eficacia).

Cabe aclarar que la observación se llevó a cabo en un hogar habitado por una madre y su hija de 7 años de edad, al presentarle al adulto el videojuego se evidenció su interés por conocer más acerca de este y comentó que es una estrategia apropiada para promover los procesos de aprendizaje de los niños. Al

ser visible su ánimo por adquirirlo, inicialmente se consideró que se obtendría un acompañamiento participativo por su parte.

No obstante, en principio reflejó lo que se afirma anteriormente, dado que la madre de familia fue la encargada de encender el computador e iniciarlo, cuando observó que el espacio se encontraba adecuado para que la niña pudiera jugar se retiró y continuó con sus deberes, esto hizo que su hija presentara dudas continuamente y que no se lograra uno de los objetivos más importantes del juego que es incentivar sus procesos de aprendizaje, al fijarse simplemente en la diversión y dejando a un lado la enseñanza que tiene la creación.

En conclusión, se puede decir que mientras exista un ambiente adecuado para los niños y un acompañamiento participativo y activo del adulto se pueden lograr importantes objetivos como un aprendizaje significativo, incentivar el trabajo en equipo, adentrar en el aprendizaje de la lógica y el proceso básico de la programación y cultivar otros aspectos que buscan una prospera educación.

7.4.3 Análisis de la Observación Individual de Usuario.

Observación Usuario.

Nombre: Juan Pablo B.

Edad: 8 años.

Lugar: Laboratorio de Informática, Colegio Claret.

El niño procede a ubicarse en su sitio para jugar, en el momento que abre la aplicación tiene una sensación de agrado, se ríe y manifiesta que el personaje tiene la misma barba de su abuelo. Sin mirar mucho el logo del videojuego, logra identificar de manera rápida y eficaz el botón de inicio, asegurando que era una roca. Acto seguido, al entrar al selector de niveles, ingresó debidamente en el nivel 1. Cabe resaltar que la docente, le explicó la historia del videojuego con anterioridad. De esta manera, fue fácil identificar los personajes con el rol

indicado. Al iniciar el nivel 1, se preguntó a sí mismo qué debía hacer, después de leer la indicación detenidamente prosiguió con la resolución de este. Bastó con no más de 25 segundos aproximadamente, para empezar a colocar los personajes a las situaciones indicadas. Manifestó que por medio de la asociación de las formas logró resolver todo el nivel, incluso, sin necesidad de mirar los entornos de cada situación. Al finalizar la acomodación de los elementos pudo reconocer el botón '*play*' como rectificador, al presionarlo obtuvo un resultado satisfactorio con el puntaje máximo y exclamó que le había gustado. Para seguir, presionó correctamente el botón 'aceptar' y regresó al selector de niveles.

Continuó con el nivel 2, entrando fácilmente. Una vez adentro, se detuvo a observar los objetos y los personajes. Su manera de jugar fue la de ir cogiendo los objetos sin un orden e ir arrastrando cada uno a su personaje relacionado. La llanta, fue la primera que agarró, pero la arrastró al personaje Cavernícola (acción equivocada); sin embargo, se retractó y la ubicó en el personaje apropiado. Con los demás objetos no tuvo problema. En este nivel existe una peculiaridad: y es que, el botón de ejecutar o "play" se encontraba en un lugar distinto al anterior nivel. No obstante, no tuvo problemas en identificarlo, por ser equivalente en color y forma. Al comprobar sus acciones, resultó que su puntaje no era el más alto, porque había fallado una vez; pero, sin prestarle mucha atención, avanzó en el juego.

En el último nivel, se detuvo a observar los botones de la interfaz, al leer la ayuda empezó a arrastrar los pasos. El niño indicó que necesitaba una guía en lo que debía hacer. La profesora le explicó que debía llegar hasta el Dinosaurio, esquivando los obstáculos para salvar a Nícolas, al escuchar esto el niño mostró su asombro y empezó a comunicarle los pasos para llegar allá. El niño empezó a describir cada uno en el orden de la secuencia sin antes arrastrarlos. Su primer intento fue fallido, ya que en vez de saltar la roca, se dirigió hacia ella. En el segundo intento logró sobrepasar la roca, pero no logró identificar cómo pasar el río. Fue hasta su tercer intento y con ayuda de la docente como hizo la secuencia

de pasos adecuada. Al terminar, concluyó diciendo que le había parecido un poco difícil hallar la secuencia, pero que era una buena idea haber salvado a Nícolas.

Los resultados finales, después de la observación, indican que el niño logró interactuar con todos los elementos del videojuego. Sin embargo, necesita de la ayuda de su tutora o docente para que se haga una idea clara de la historia, claramente esto era lo que buscaba el videojuego, que el docente fuera una guía.

A partir de lo anterior se decretó que la motivación es alta, el diseño de la interfaz es claro, y el tipo de aprendizaje consigue que se lleve a cabo la realización de los niveles y el conocimiento de los temas, con la participación del adulto o docente.

7.4.4 Resultados Obtenidos en las Encuestas a Docentes

Al momento de analizar los datos obtenidos se eligieron las preguntas con más relevancia para cumplir con el objetivo de analizar cada característica del prototipo: diseño, interactividad, motivación, método de enseñanza y su eficacia.

El siguiente es el análisis de las encuestas a 7 docentes de las áreas de Tecnología, de Lógica y de Introducción a la Programación. Cada uno de ellos se encuentran clasificados en: 4 profesores de tecnología, 2 profesores de lógica y 1 de introducción a la programación.

- **Historia del Videojuego**

Pregunta 1: ¿El videojuego incita al jugador a conocer los temas básicos de la lógica y el proceso básico de la programación a través de situaciones cotidianas?

Análisis: El 73% de los encuestados están muy de acuerdo en que la historia le permite a los niños conocer sobre la lógica y el proceso básico de la programación a través de ejemplos cotidianos. El 27% estuvo de acuerdo con lo anterior. Nadie estuvo en desacuerdo.

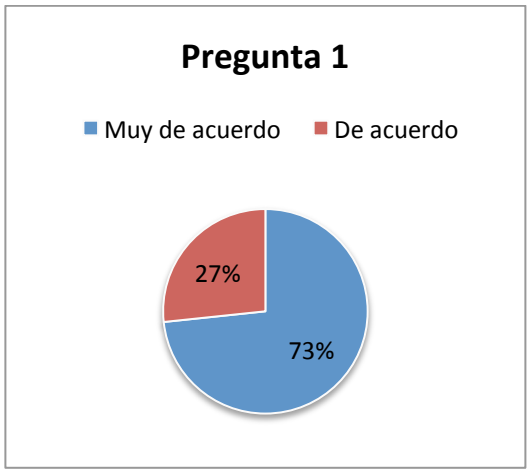

Gráfico 1. Porcentaje de respuestas. Pregunta 1

Pregunta 2: ¿Los términos e instrucciones del juego son entendibles por el jugador?

Análisis: El 43% de los encuestados están muy de acuerdo en que los términos e instrucciones son entendibles por el jugador, el 43% estuvo de acuerdo y el 14% estuvo con una postura neutral. Nadie estuvo en desacuerdo.

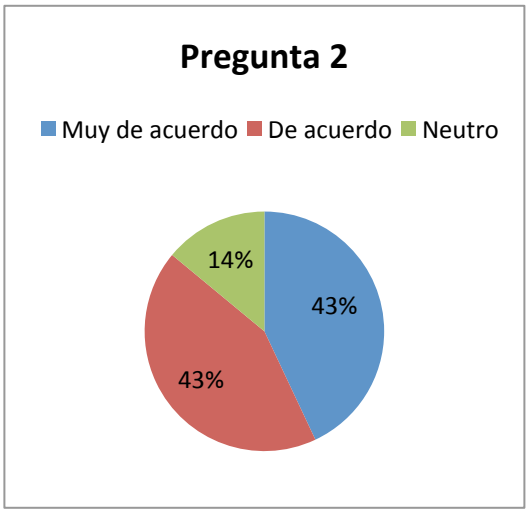

Gráfico 2. Porcentaje de respuestas. Pregunta 2

- **Diseño del Videojuego**

Pregunta 3: ¿Los personajes del videojuego se asocian con la historia?

Análisis: El 80% de los encuestados están muy de acuerdo en que los personajes están asociados a la historia, el 20% estuvo de acuerdo. En esta pregunta no se contaron con respuestas contradictorias.

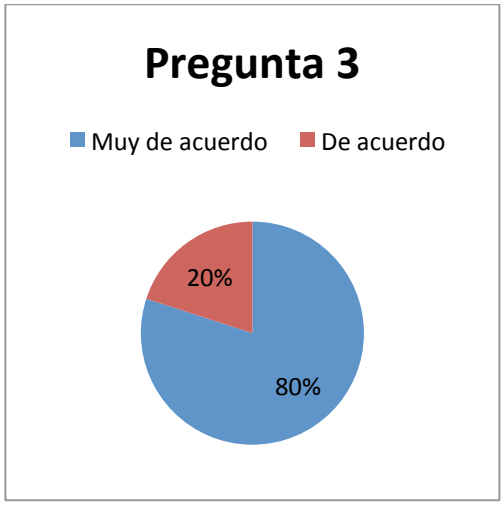

Gráfico 3. Porcentaje de respuestas. Pregunta 3

Pregunta 4: ¿ El diseño en general, de los ítems y los botones es adecuado?

Análisis: El 57% de los encuestados están muy de acuerdo en que los ítems y botones están adecuados, el 29% estuvo de acuerdo; pero el 14% estuvo en desacuerdo.

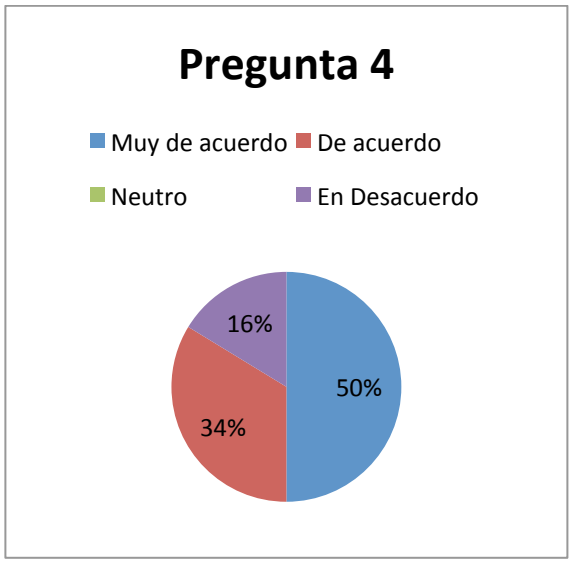

Gráfico 4. Porcentaje de respuestas. Pregunta 4

- **Interactividad**

Pregunta 5: ¿El jugador interactúa fácilmente con los objetos del videojuego?

Análisis: El 57% de los encuestados están muy de acuerdo en que se puede interactuar fácilmente con los elementos del videojuego, el 14% se encuentra neutro y el 29% está en contra de esta pregunta. De esta manera más de la mitad de los docentes creen que el jugador interactúa fácilmente.

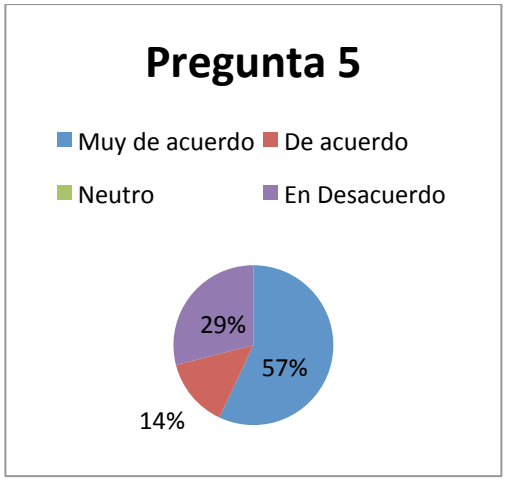

Gráfico 5. Porcentaje de respuestas. Pregunta 5

- **Método de enseñanza**

Pregunta 6: ¿A través del videojuego se utilizan, manifiestan o demuestran conceptos de la lógica matemática y el proceso básico de la programación?

Análisis: El 57% de los encuestados, está muy de acuerdo con que a través del videojuego se utilizan estos conceptos. Aunque un 29% está en desacuerdo, existe un 14% que está de acuerdo que respaldan la afirmación. Para un total de 71% en respuestas positivas, se puede concluir, que se hizo ajustada la planeación de los niveles y se adaptaron al videojuego. En general, se puede inferir que, para los docentes es una herramienta de aprendizaje en los temas planteados.

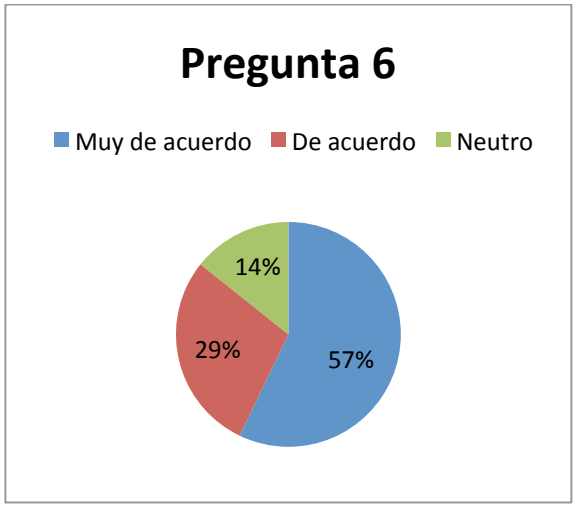

Gráfico 6. Porcentaje de respuestas. Pregunta 6

Pregunta 7: ¿El uso de este videojuego puede ser útil para reforzar e incentivar los conocimientos de la lógica matemática y el proceso básico de la programación?

Análisis: El 14% de los encuestados están muy de acuerdo en que dicho videojuego, utiliza o demuestra conceptos de la lógica matemática y el proceso básico de la programación, el 71% está de acuerdo, el 29% se encuentra en desacuerdo. Sin embargo, las opciones “muy de acuerdo” y “de acuerdo” tienen una totalidad de 85%, un resultado exitoso y positivo para la investigación. Se

puede decir que para los profesores, “Ayuda a Caver & Nícolas” es una herramienta para incentivar el aprendizaje de la lógica y el proceso básico de la programación.

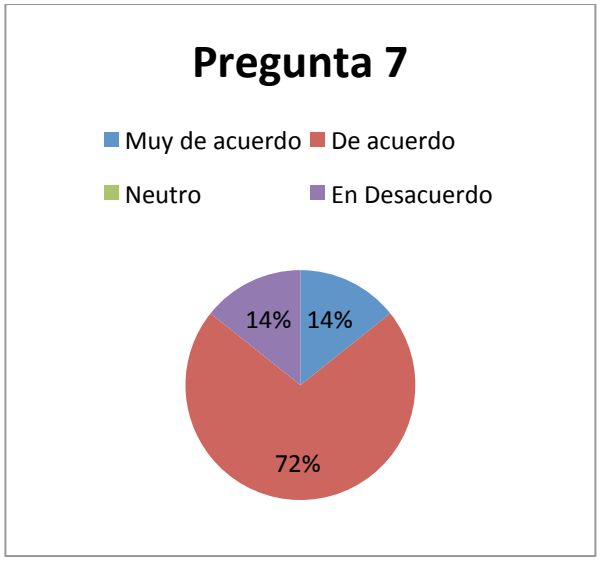

Gráfico 7. Porcentaje de respuestas. Pregunta 7

7.1.4.5 Resultados Entrevistas Abiertas a Usuarios.

Para los presentes resultados se tuvieron en cuenta las preguntas más relevantes para la investigación. El resto de respuestas se pueden ver en el ANEXO E

- **El gusto y motivación por el juego:**

En la primera pregunta los niños respondieron abiertamente, y 7 de ellos afirmaron que les había gustado el videojuego. Entre los demás: 1 de ellos aseguró que le había parecido aburrido. Los otros dos respondieron que ese tipo de videojuegos no eran de su agrado, ya que le gustaban más de otros géneros, como el de aventura**.** Con el excelente resultado, se mostró que a la mayoría de niños les gustó el videojuego.

En la segunda pregunta manifestaron si le gustaba o no los personajes de la historia, 7 de ellos respondieron que les gustaba y recalcaron aspectos como: la vestimenta, la gracia, el significado y los colores.

En el tema de la motivación con respecto al videojuego, la mayoría de niños manifestó que les interesó empezar a jugar, desde que les contaron la historia.

- **Metodología de Enseñanza:**

Analizando las preguntas se pudo ver que los niños entendieron el tema de generar algoritmos con el uso de la lógica y conocer el proceso básico de la programación. En la pregunta 7, indicaron que pudieron crear una solución al problema para salvar al personaje, mediante la creación del algoritmo con los pasos. Para 9 de los niños entrevistados, aprendieron algo en el videojuego. Explicaron que aprendieron a darle órdenes al cavernícola, crearon un algoritmo para la resolución del problema; y seleccionaron, diferenciaron y agruparon diferentes objetos.

Se puede concluir, observando lo anterior, que el método de aprendizaje escogido fue efectivo, por generar en ellos una enseñanza de hacer por ellos mismos, aún teniendo el apoyo del profeso

7.5 IMPLEMENTACIÓN

El apartado o fase de la implementación agrupa todo el trabajo de codificación de la aplicación que hasta este punto se ha ido construyendo.

Llegados a este punto, a groso modo, es cuando debe empezarse a programar, lo cual implica haber escogido el o los lenguajes de programación que mejor se

adapten a nuestro proyecto, las bases de datos correspondientes que se precisen, la tecnología que garantice el éxito, etc.

El Framework “enchant.js” es orientado a objetos, así que cada elemento mostrado en pantalla es realmente un objeto. Las ventajas de usar este Framework que es multiplataforma, corre libremente en iOs, Android, Osx, Windows, Linux, etc. Cualquier dispositivo que posea un buscador de internet es libre de correrlo. Esta basado en *event listeners* lo cual permite una comunicación de forma asíncrona entre el usuario y la aplicación.

A continuación se hará una pequeña introducción a *enchant.js* con las descripciones de sus componentes:

```
1   <!doctype html>
2   <html><head><script src="enchant.js"></script></head>
3   <body style="margin:0; padding: 0;">
4   <script>
5       enchant(); // initialize
6       var game = new Core(320, 320); // game stage
7       game.preload('chara1.png'); // preload image
8       game.fps = 20;
9
10      game.onload = function(){
11          var bear = new Sprite(32, 32);
12          bear.image = game.assets['chara1.png'];
13          game.rootScene.addChild(bear);
14          bear.frame = [6, 6, 7, 7];   // select sprite frame
15
16          bear.tl.moveBy(288, 0, 90)   // move right
17              .scaleTo(-1, 1, 10)      // turn left
18              .moveBy(-288, 0, 90)     // move left
19              .scaleTo(1, 1, 10)       // turn right
20              .loop();                 // loop it
21      };
22
23      game.start(); // start your game!
24  </script>
25  </body>
26  </html>
```

Figura 26. Ejemplo de la estructura de código de un juego sencillo con enchant.js

Lo primero que se hace al implementar enchant.js es crear un script el objeto enchant el cual viene siendo el juego como tal, un objeto que contiene el juego y lo representa en un canvas autogenerado, se inicializa con enchant(), lo siguiente que se crea es el core del juego, el nucleo del escenario “var game = new

Core(320, 320);” estos dos parámetros vienen siendo el ancho y alto final del juego, a este objeto se le asignan todas las características del juego, tales como fps(frames per second) y los assets del juego. En esta primera etapa se establecen las características del entorno del desarrollo, todo lo que se debe tener en cuenta antes de comenzar a estructurar la aplicación.

Luego viene el Core.onload(){ } o en este caso game.onload(){ }, esta función se ejecuta una sola vez y en ella se establece la estructura del juego como tal, se cargan los sprites y se posicionan, se añaden los event listeners, para definir la naturaleza de la interaccion con el usuarios y la lógica del comportamiento de los objetos en el juego y dentro de estos alterar dinámicamente la estructura del juego (movimientos de sprites, carga de niveles, creación de nuevos objetos, etc).

En este caso se crea un objeto sprite de 32px de ancho por 32px de alto “var bear = new Sprite(32, 32); “ ,se le asigna la ruta del sprite que contiene la apariencia del objeto “bear.image = game.assets['chara1.png'];”, y se añade a un objeto Scene, o como en este caso, se añade a la scene por defecto que es la rootScene “game.rootScene.addChild(bear);”, se seleccionan los sprites a utilizar “bear.frame = [6,6,7,7];” (mas adelante se entrará mas en detalle en el como se comportan estos objetos y el porque de estos parámetros de entrada), en este momento ya el sprite como tal esta creado y falta decirle como debe comportarse.
En este caso especial, se va a animar los sprites con el método “moveBy()” el cual es un comando de movimiento muy sencillo, y muy útil para ciertos casos donde se necesite un movimiento de sprites básico, y que puede ser concatenado con métodos de transformación de sprites tales como “Scale()”, “Rotate()”, etc. En este método “moveBy(288, 0, 90)” el primer parámetro corresponde a la posición en el eje x al cual se va a ir, el segundo parámetro corresponde a la posición en el eje y a la cual se va a ir y el tercer parámetro corresponde a la cantidad de frames que va a requerir este desplazamiento.

Para el desarrollo del videojuego lo primero que se hizo fue crear clases para los diferentes objetos que tuvieran un método constructor el cual ahorrara declaraciones a la hora de añadirlo a una escena.

```
13▼    var ImageObject = enchant.Class.create(enchant.Sprite,{
14▼        initialize: function( src, width, height, x, y, scene ){
15             this.width = width;
16             this.height = height;
17             enchant.Sprite.call(this, this.width, this.height);
18             this.x = x;
19             this.y = y;
20             this.scene = scene;
21             this.frame = 0;
22             this.image = game.assets[src];
23             scene.addChild(this);
24         }
25     });
```

Figura 27. Código de clase de objetos de imágenes

En la figura 25 se puede observar cómo se crea una clase básica con parámetros de entrada como la dirección del *sprite*, el ancho, el alto, su posición en el eje *x*, su posición en el eje *y*, y la escena a la cual se va a añadir, dentro de este método constructor se puede apreciar como cada parámetro es instanciado dentro del objeto.

En enchant.js hay que pensar en las escenas como un *deck*, donde se añaden todos los sprites que va a contener, y en el momento que se requiera, se carga la escena.

```
747        var main = new Scene;
748        main.name = "main";
749        var stage_selection = new Scene;
750        stage_selection.name = "stage_selection";
751        var stage1 = new Scene;
752        stage1.name = "stage1";
753        stage1.unlocked = true;
754        stage1.finished = false;
755        var stage2 = new Scene;
756        stage2.name = "stage2";
757        stage2.unlocked = false;
758        stage2.finished = false;
759        var stage3 = new Scene;
760        stage3.name = "stage3";
761        stage3.unlocked = false;
762        stage2.finished = false;
```

Figura 28. Estructura de creación de las escenas de la pantalla principal, la selección de niveles y los tres niveles

La estructura está conformada por un main, una escena de selección de niveles y tres niveles diferentes, donde el main es el primer acceso que se tiene al videojuego y es la pantalla inicial, la selección del nivel es donde se puede accesar a todos los niveles, cada nivel instanciado con su nombre y un estado de desbloqueado y otro de finalizado, esto último para controlar el acceso al nivel junto con un posible nuevo acceso ya sea para mejorar el score, o simplemente para experimentarlo nuevamente. Ver **Figura 29.**

```
772        var background_main = new ImageObject(main_background,720,480,0,0,main);
773        main.addEventListener(Event.ENTER_FRAME,function(){
774            //console.log("aqui");
775            if(background_main.frame == 0){
776                background_main.frame = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19];
777            }else if(background_main.frame == 19){
778                var jugar = new ButtonObject(boton_jugar,200,150,500,310,main);
779                jugar.addEventListener(Event.TOUCH_END,function(){
780                    game.pushScene(stage_selection);
781                });
782                background_main.frame = [20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44];
783            }
784        });
785
786        game.pushScene(main);
787
```

Figura 29. Creación de objeto de clase ImageObject y adición de EventListener

En esta imagen se puede observar como se crea un *ImageObject* que cumple la función de imagen de fondo y se le añade un *evenListener*, en este caso es un *Event.ENTER_FRAME*, el cual llama a su callback function cada vez que cambia

de frame, dentro de esta se cargan los *frames* del *sprite* de *background*, todo esto para dar paso a la animación inicial del desarrollo. Paralelo a esto se puede apreciar como al llegar a cierto *frame* se añade al *stage* un objeto de clase *ButtonObject*, se inicializa y además se le añade un *eventListener*, esta vez es un *Event.TOUCH_END*, el cual se activa cada que se termina un *click* dentro del objeto, y dentro de su *callback function* encontramos como desde el core se usa el método *pushScene*() para cargar una escena diferente, asi que finalmente cuando el sprite deja de ser presionado, automáticamente carga la escena de selección de niveles.

NIVEL 1

Figura 30. Interfaz del nivel 1

En el nivel uno se encuentran objetos de clase *DragableObject, DropzoneObject, ButtonObject* e *ImageObject*. En este nivel basado en relación de

figuras y formas, consta de 4 objetos arrastrables y una zona para dejarlos, cada uno corresponde a una zona específicamente. La complejidad de este nivel es hacer que cada objeto ocupe una posición y la *dropzone* se muestre ocupada por un objeto, y que al retirar este objeto, se retire a su vez el estado ocupado de la *dropzone*, evaluar que el objeto que esta ocupando la *dropzone* es efectivamente el que le pertenece y evaluar que no quede ningún objeto sin posicionar. Para ello se ha desarrollado el siguiente diagrama de flujo.

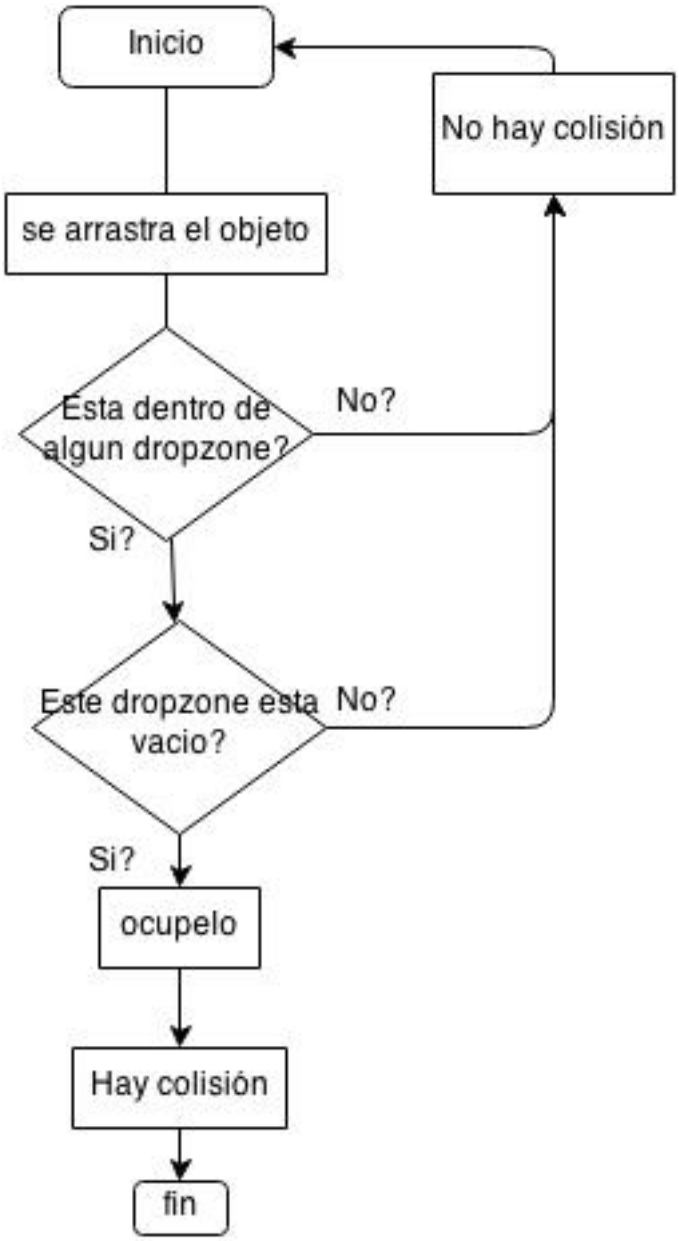

Figura 31. Diagrama de flujo de las acciones del nivel 1

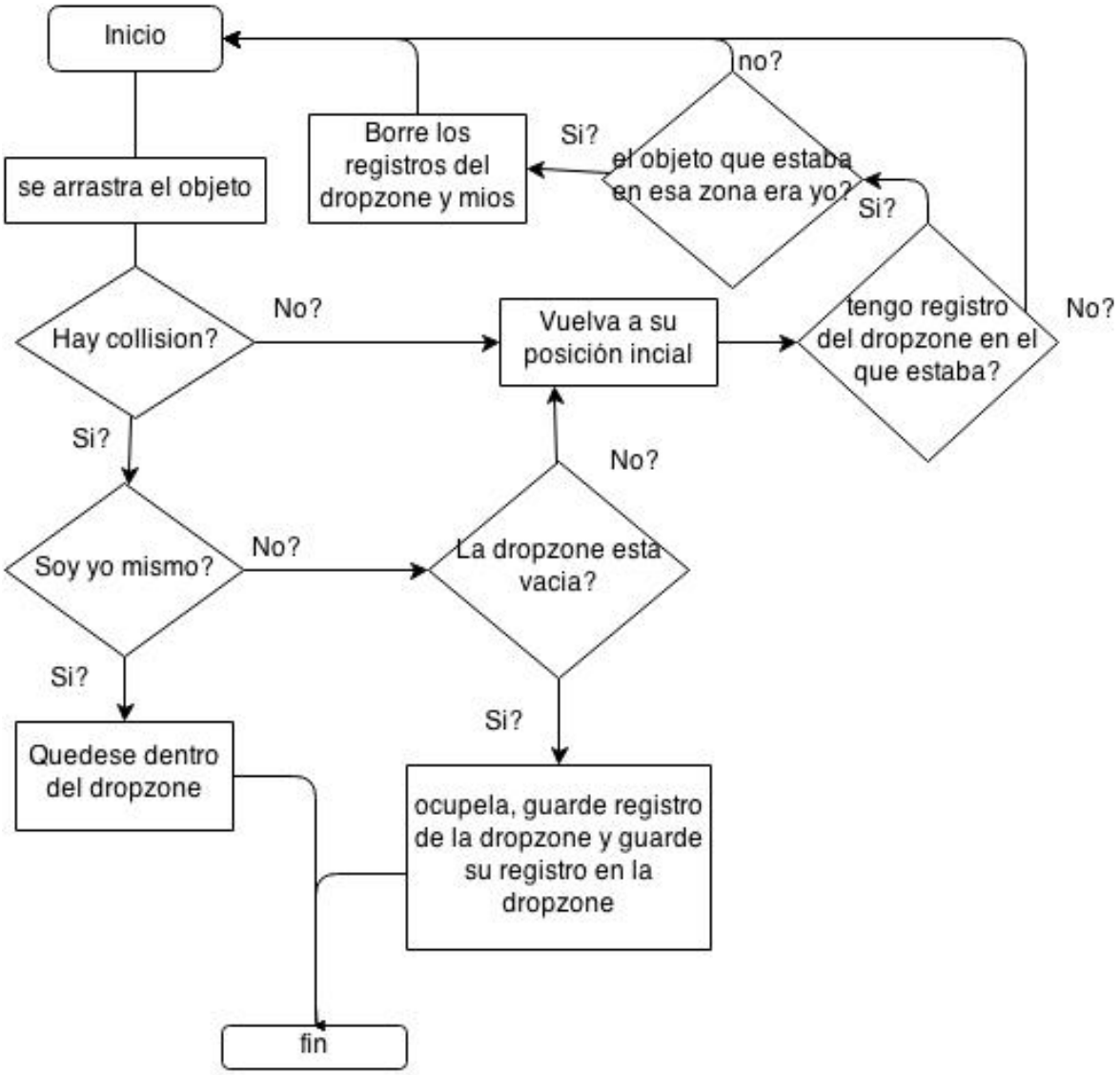

Figura 32. Diagrama de flujo de las acciones de los objetos de clase DraggableObject

Siguiendo esta lógica se ha logrado que los objetos ocuparan los *dropzones* cuando estaban vacíos, al ser levantados y puestos en el mismo sitio se evitaba que la *dropzone* figurará como llena, que al tener el objeto en una *dropzone* y pasarlo a otra, ambas quedarán ocupadas con el mismo objeto bloqueándolas, que al poner un objeto dentro de una *dropzone* ocupada esta se llenará con la información del segundo objeto pero figurará con el primero. Entonces así, al presionar el botón de *play*, se recorren los *dropzones* y se comparan los registros,

para saber con certeza cuales tienen su objeto correspondiente y cuales no, retroalimentando al jugador con las posiciones de las respuestas buenas y malas, y permitiendo la corrección de las mismas.

NIVEL 2

Figura 33. Interfaz del nivel 2

En el nivel dos se encuentran objetos de clase *DragableObject, DropzoneObject*, *ButtonObject* e *ImageObject*, muy similar al nivel uno, pero con la diferencia que a un *dropzone* pertenecen varios objetos. Este nivel basado en teoría de conjuntos básica, donde un grupo de objetos pertenecen a un conjunto determinado, consta de ocho objetos, cuatro de los cuales pertenecen a un grupo y los demás a otro, y dos *dropzones* alusivas a los conjuntos. El desarrollo de este nivel consiste en que los objetos que sin similares en contexto se agrupen dentro

de su respectivo grupo, en este cazo un *dropzone*. La complejidad de este nivel se presenta a la hora de determinar de una forma eficiente que objetos pertenecen a determinado grupo y que objetos no. Para esto los objetos se instancian con una data correspondiente a su *dropzone*. De esta forma al estar todos los objetos organizados y enviar la respuesta, se determina cuantos objetos no están en su conjunto respectivo y se retroalimenta al usuario indicando cuales son estos objetos. De esta forma el usuario puede corregir sus respuestas y llegar a determinar cual es la naturaleza del factor que determina cuales son los grupos y el por que de estos. Para determinar que la correcta posición de los objetos se desarrollo un diagrama de flujo del proceso de evaluación de la distribución al presionar el botón de *play.*

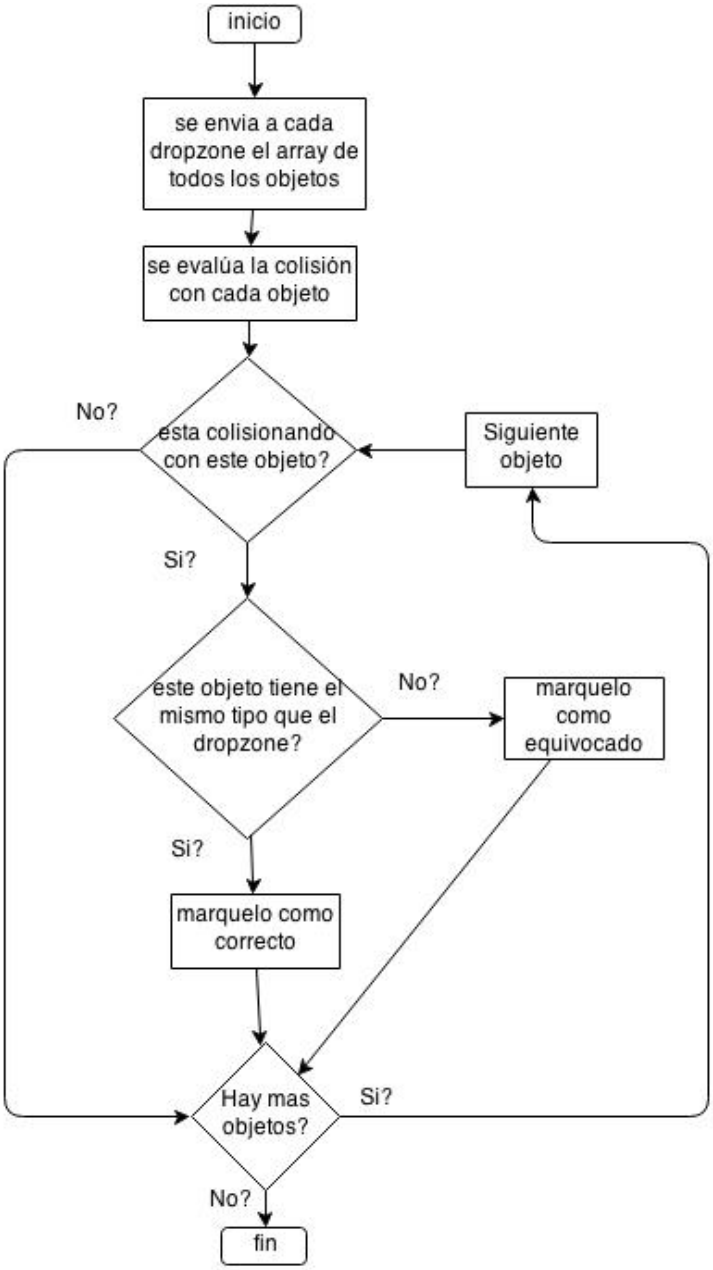

Figura 34. Diagrama de flujo del nivel 2

Al terminar la evaluación de las respuestas, si todas son correctas, se indica el puntaje del nivel y se da paso a desbloquear es siguiente nivel y a cargar la escena del siguiente nivel, en caso contrario se indican las respuestas erróneas y el usuario tiene una nueva oportunidad para acomodar sus respuestas.

NIVEL 3

Figura 35. Interfaz del nivel 3

En el nivel tres se encuentran objetos de clase *DragableObject*, *DropzoneObject, ButtonObject* e *ImageObject* y un objeto llamado player, este nivel basado en algoritmos y consta de cuatro *DragaavleObject* con la característica de que al ser arrastrado se debía crear un nuevo objeto de las mismas características en esa posición, y si al soltarlo no hacia contacto con algún *dropzone* debía ser removido, 6 *dropzones* en donde se estructura la secuencia lógica a seguir, un objeto tipo Player que se comporta de acuerdo a las ordenes que se le den, y finalmente tres *ImageObject*, dos de ellos usados como obstáculos y el ultimo usado para interactuar con el entorno, tiene un grado de complejidad mayor puesto que su característica principal se basa en la reproducción de algoritmos para lograr un cometido, en el desarrollo de este nivel

se presentaron una serie de grandes problemáticas, como la reproducción de cada comando en intervalos de tiempo, la interacción entre las ordenes y el *player,* para solucionar esto ultimo se determino que la mejor solución era tener métodos en el *player* que manejaran estados booleanos que fueran escuchados desde el *Event.ENTER_FRAME* y de acuerdo al estado, se comportara de la forma adecuada, para complementar esta implementación, los objetos correspondientes a los comandos se instanciaron con un nombre igual al de su método correspondiente.

```
1047        for(i=0;i<dropzone_array_stage3.length;i++){
1048
1049            if(i != 0){
1050                // var foo = (condition) ? true : false;
1051                nextDelay = (i != dropzone_array_stage3.length - 1)
1052                ? player[dropzone_array_stage3[i+1].data+"_delay"]
1053                : player[dropzone_array_stage3[i].data+"_delay"];
1054                delay += (nextDelay > 0) ? nextDelay : 0;
1055            }
1056            //console.log("Ejecutando metodo: "+player[dropzone_array_stage3[i].data]+" con un delay acomulado de: "+delay);
1057            check = executeMethod(player,dropzone_array_stage3[i].data,'',delay);
1058        }
```

Figura 36. Código de la reproducción de las acciones del nivel 3

En la imagen se puede apreciar como se recorren los *dropzones* y se ejecuta la siguiente función:

```
1069        function executeMethod(player,method,params,delayMS){
1070
1071            //setTiemout(metodo,delay,parametros);
1072            var timer = setTimeout(function(){
1073
1074                        //console.log("true");
1075                        if(player.collision == false){
1076                            player[method](params);
1077                        }else{
1078                            clearTimeout(timer);
1079                        }
1080
1081                        return true;
1082
1083                },delayMS);
1084        }
```

Figura 37. Función de ejecución de las acciones con delay

La función *setTimeout*, recibe dos parámetros, el primero es la función a ejecutarse, pudiendo declarar esta dentro, o estableciéndola aparte e invocándola

simplemente, y el segundo parámetro consta de un *delay* en milisegundos, de esta forma la ejecución de las funciones se pone en *qeue list* y quedan en espera de ser ejecutadas. Paralelo a lo anterior, se instancio un estado de colisión del *player* con los diferentes obstáculos, siendo esto evaluado cada *frame* para que al momento de interceptar un obstáculo, se detenga la reproducción de funciones y se elimine la *qeue list*, logrando detener el movimiento.

```
687        forward: function(){
688            this.jumpp = false;
689            this.upp = false;
690            this.hit = false;
691            this.frame = 1;
692            this.current_x = this.x;
693            this.fwd = true;
694            //console.log("forward");
695        },
696        up: function(){
697            this.fwd = false;
698            this.jumpp = false;
699            this.hit = false;
700            this.frame = 2;
701            this.jump_dist = -1;
702            this.up_count = 1;
703            this.upp = true;
704            //console.log("up");
705        },
706        jump: function(){
707            this.upp = false;
708            this.fwd = false;
709            this.hit = false;
710            this.frame = 3;
711            this.jump_dist = -1;
712            this.current_x = this.x;
713            this.current_y = this.y;
714            this.jumpp = true;
715
716        },
717        bat: function(){
718            this.upp = false;
719            this.jumpp = false;
720            this.fwd = false;
721            this.up_count = 1;
722            this.hit = true;
723            //console.log("bat");
724        },
```

Figura 38. Métodos de la clase Player de ejecución de las acciones

En esta imagen se puede apreciar la lista de métodos del objeto de clase player, y como se comportan los estados que dan pie a las acciones, a continuación se verá un diagrama de flujo del comportamiento de estos estados en el Event.ENTER_FRAME.

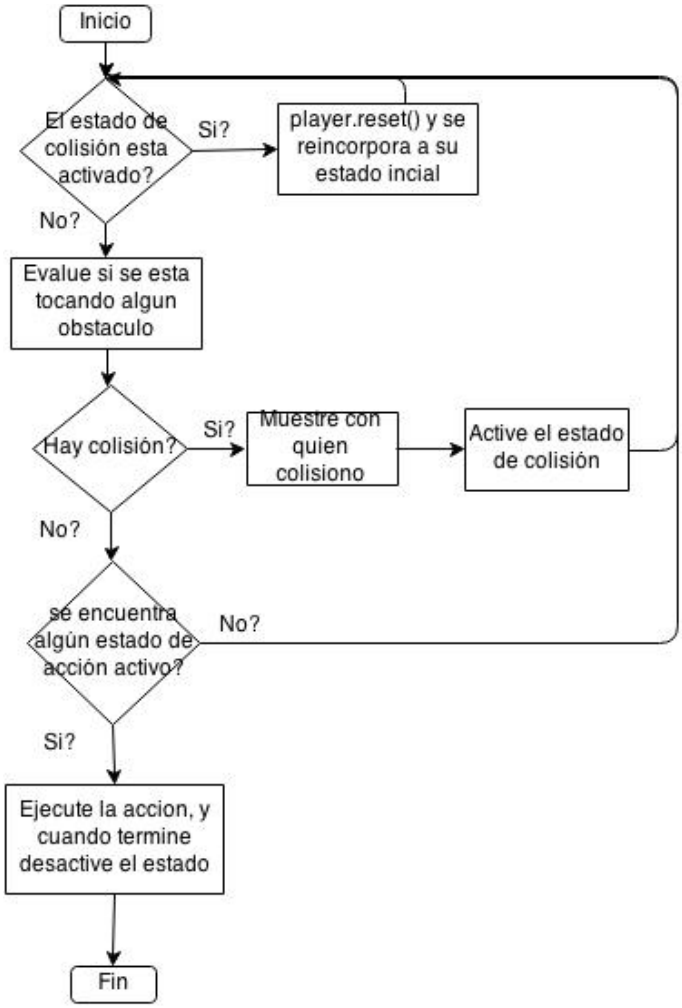

Figura 39. Diagrama de flujo de detección de colisiones con los obstáculos

ACCIONES

Todas las acciones de el objeto player del nivel tres, están conformados por booleanos que activan o desactivan condicionales dentro de una función Event

Listener dentro de un Event “enterframe”. Cada frame se evalúan los condicionales para dar o no, paso a las acciones de los métodos activados por los comandos del nivel explicados anteriormente.

ACCIÓN FORWARD

```
602    if (this.fwd == true) {
603        if (this.x<(this.current_x+88)){
604            this.x+=3;
605        }else{
606            this.frame = 0;
607        }
608    }
```

Figura 40. Ejecución de la acción Forward

Ésta acción es activada por la instancia del objeto “player” de tipo boolean “fwd”. Durante esta acción el objeto “player” se moverá única y exclusivamente en el eje x y su recorrido durante esta acción son 88 pixeles. Asi que para evaluar el recorrido, se guarda un registro de la posición en el eje x antes de iniciar la acción y en cada frame se evalua que la posición en x actual sea menor que la posición en x inicial + 88, cada frame que ingresa en este condicional, el objeto adopta el sprite de “caminar” el cual esta en la posición objec.frame=1 y se incrementa la posición en x por 3 pixeles. Al conseguir la posición deseada, el objeto no cumple la condición del condicional y cambia su apariencia a “normal” contenida en object.frame = 0. De esta forma se concluye con la acción, si no hubo colisión alguna, el objeto “player” permanece en reposo nuevamente hasta que se ejecute la siguiente acción, de lo contrario el objeto vuelve a su posición inicial, a la espera de un nuevo intento.

ACCIÓN JUMP

```
610  if(this.jumpp == true){
611      if(this.x<(this.current_x+109)){
612          var y_temp = 0;
613          var parabola = 0;
614
615          parabola = ((-1)*(this.jump_dist)*(this.jump_dist)) + (4*(this.jump_dist)) + 5;
616
617          y_temp = ((54/9)*(parabola))+(1);
618
619          this.x += 3.67;
620          this.y = this.y + ((this.init_y-y_temp)-this.y);
621          this.jump_dist += 0.2;
622      }else{
623          this.y = this.init_y;
624          this.x = this.current_x+110;
625          this.frame = 0;
626      }
627  }
```

Figura 41. Ejecución de la acción Jump

Ésta acción es activada por la instancia del objeto “player” de tipo boolean “jump”. Durante esta acción el objeto “player” toma una trayectoria parabólica hacia adelante, emulando un salto alto hacia el frente. Este movimiento en particular es ligeramente mas complejo por la forma de dibujar la trayectoria del objeto. Se determino que la curvatura que cumple con los requerimientos es la ecuación parabólica: $y = -x^2 + 4x + 5$.

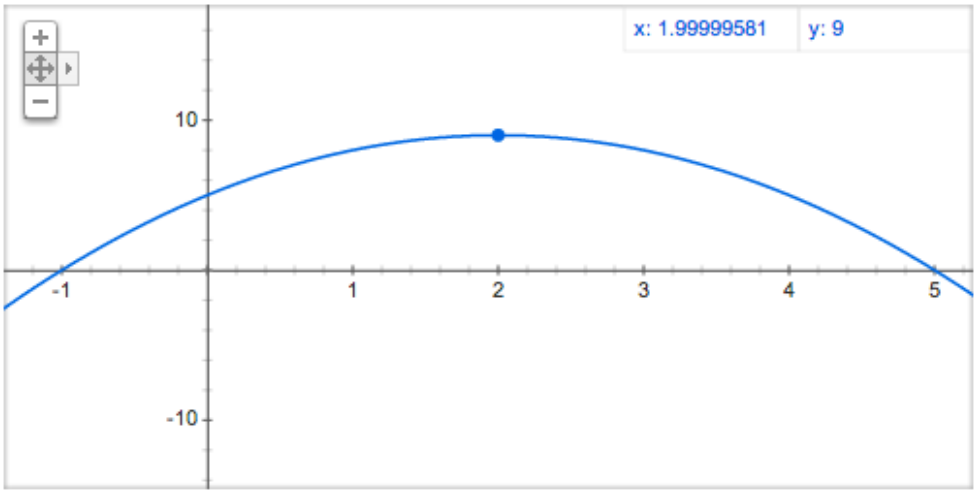

Figura 42. Gráfico de la función parabólica $\boldsymbol{y = -x^2 + 4x + 5}$

Ecuación en la cual en el intervalo cerrado x[-1,5], toma valores positivos y que van desde 0 hasta 9. Esta función cumple con la trayectoria requerida para el movimiento, de esta forma, se instancio un entero de nombre “jump_dist” el cual es inicializado en -1, aumentando en cada iteración en 0.2 ya que se determino que la acción debía llevar a cabo en 30 iteraciones, y finalmente finalizando el movimiento en 5. Adicional a esto, se requirió que la trayectoria del objeto fuese mayor en el eje y, de esta forma, justo después de evaluar la parábola, se mapea al rango requerido que es y[1,45].

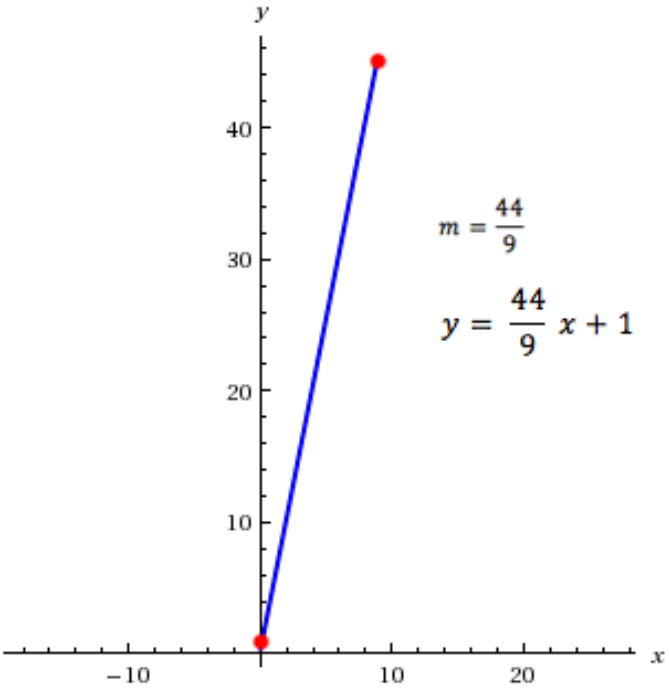

Figura 43. Gráfico de la función de escalamiento de la posición en el eje Y del movimiento parabolico

La distancia requerida a recorrer en x para esta acción se determinó en 110 pixeles, y la cantidad de frames requeridos para la acción se determinó en 30 frames. Asi que durante cada frame se incremento la posición en el eje x en 3.67, y la posición en el eje y se determino por las ecuaciones anteriores. Al finalizar la

acción, y como medida preventiva se ubico el objeto en la posición Y incial, y la posición en x incial + 110. Al finalizar la acción, si no hubo colisión con alguno de los obstáculos, el objeto permanece en estado de reposo a la espera de una nueva acción, o, vuelve a su posición inicial para un nuevo intento.

ACCIÓN UP

```
629  if(this.upp == true){
630      if(this.up_count <15){
631          this.y -= 3;
632      }
633      if(this.up_count == 15){
634          if(this.intersect(this.obstacles[2])){
635              this.liana = true;
636          }else{
637              this.y -=3;
638          }
639      }
640      if(this.up_count > 15 && this.up_count <30){
641          if(this.liana==true){
642              if(this.up_count == 16){
643                  this.frame = 0;
644              }
645              if(this.up_count >= 30 && this.up_count <= 35){
646                  this.y += 7.05;
647              }
648              var y_temp = 0;
649              var parabola = 0;
650              parabola = ((-1)*(this.jump_dist)*(this.jump_dist)) + (4*(this.jump_dist)) + 5;
651              y_temp = ((14/9)*(parabola))+(1);
652              this.x += 14.5;
653              this.jump_dist+=0.4;
654              this.y = this.y + (((this.init_y-45)+y_temp)-this.y);
655          }else{
656              this.y +=3;
657          }
658      }
659      if(this.up_count == 36){
660          this.y = this.init_y;
661          this.frame = 0;
662      }
663      this.up_count++;
664  }
```

Figura 44. Ejecución de la acción Up

Ésta acción es activada por la instancia del objeto “player” de tipo boolean “upp”. Durante esta acción el objeto “*player*” se moverá única y exclusivamente en el eje *y*, con un recorrido de 45 pixeles durante esta acción. Según la estructuración del nivel, con esta acción es la única forma en la cual el jugador consigue alcanzar un objeto “*liana*” para cruzar el lago y llegar a la otra orilla, en consecuencia, cada

que el objeto realiza esta acción y se encuentra en su punto mas alto, se debe evaluar si hay contacto o no con el objeto "*liana*", de no ser así, el objeto continua con su descenso normalmente, pero en el caso de que se produzca el contacto, automáticamente el objeto realiza una trayectoria parabólica hacia el frente, emulando colgar de la liana, y un descenso rápido al finalizar esta trayectoria, esto se consiguió, usando la misma ecuación parabólica de la acción "*jump*" con la diferencia de que el mapeo en el eje y es menor (y[1,15]) y que se implementa de forma invertida, en vez de aumentar la distancia, se resta, consiguiendo así, un desplazamiento adecuado hacia la otra orilla.

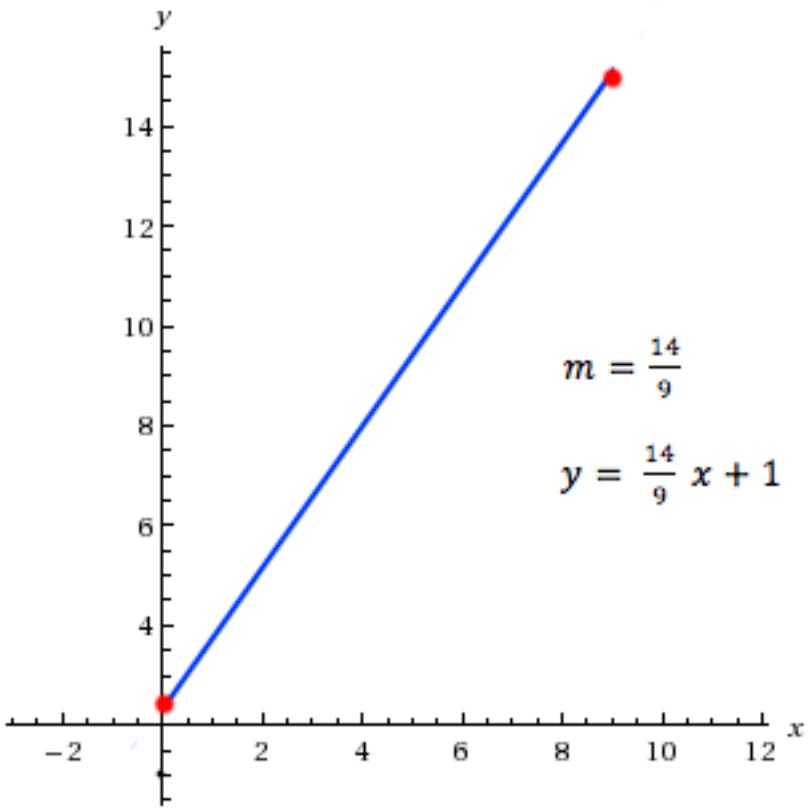

Figura 45. Gráfico de la función de escalamiento de la posición en el eje Y del movimiento parabólico de la liana

Finalmente al terminar la acción, de no haber colisión con algún obstáculo y de no haber contacto con el objeto "liana", el objeto entra en reposo en la misma

posición donde inició la acción, de haber contacto con la liana, el objeto incrementa su posición en x en 420,5 pixeles, después de cumplir con la trayectoria antes mencionada, de haber colisión el objeto vuelve a su posición inicial.

ACCION BAT

```
666  if(this.hit == true){
667      if(this.up_count <= 3){
668          if(this.up_count == 3){
669              this.frame = 1;
670          }
671          this.x += 3;
672      }
673      if(this.up_count == 3){
674          this.frame = 4;
675      }
676      if(this.up_count == 20){
677          this.frame = 0;
678          this.hit = false;
679      }
680      this.up_count++;
681  }
```

Figura 46. Ejecución de la acción Bat

Ésta acción es activada por la instancia del objeto "*player*" de tipo *boolean* "*hit*". Durante esta acción el objeto "*player*" incrementa su posición en x, 9 pixeles, y al cumplimiento de esto, carga el frame de golpear object.frame = 4.

Esta acción es usada en la parte final del nivel para luchar contra el dinosaurio que intenta atacar a Nícolas. De haber contacto con este "enemigo" se concluye el nivel exitosamente, de lo contrario y de no haber colisión con algún obstáculo, el objeto "*player*" entra e estado de reposo a la espera de la ejecución del siguiente comando.

CONCLUSIONES

La presente investigación se realizó con el objetivo de desarrollar un videojuego web que facilite el aprendizaje del proceso básico de la programación y la lógica matemática en niños de 7 a 11 años de edad, luego de llevar a cabo el proceso investigativo a través de siete capítulos, cada uno con diferentes propósitos pero que convergen a una misma intención, se concluyó que:

- Los videojuegos son un medio atractivo para la población infantil, permitiendo que la educación a través de estos sea lúdica y significativa para los niños que oscilan entre los 7 y 11 años de edad, logrando captar su atención y siendo participes de sus procesos de enseñanza-aprendizaje.

- Es importante considerar a los diferentes jugadores que harían uso del videojuego , dado que conocer el comportamiento de cada usuario, su personalidad, culturalidad, género y aspiraciones, permitiría prevenir algún tipo de adicción, o por el contrario se buscaría invitar al niño a que se anime por aprender de una manera innovadora y llamativa.

- Los jugadores y/o estudiantes no solo lograrían mejorar los aspectos académicos y cognitivos, sino que fomentarían otras habilidades importantes como lo son el trabajo en equipo, el desarrollo del pensamiento reflexivo y la interacción entre varias personas.

El videojuego fue desarrollado en web por la facilidad de uso en múltiples plataformas, esto debido a que se puede acceder a este mediante un ordenador, una Tablet, un Smartphone, o cualquier dispositivo con acceso a internet y soporte *javascript*, de esta manera se eliminaron barreras geográficas y de plataforma. Por otra parte, al estar desarrollado por completo en *javascript*, hace que su soporte

sea casi universal, ya que no requiere el uso de *plugins* u otros aditamentos que podrían llegar a ser una complicación a la hora de usar de la aplicación.

Este es un proyecto con un interés más que educativo: social, puesto que se busca que mas allá de desarrollar la capacidad lógica de los niños, derrumbe los grandes paradigmas que circulan en torno a las disciplinas tecnológica. Mediante el presente se pudo derribar un poco el pensamiento casi general, que la lógica de la programación es algo complejo, demostrando que esta se encuentra en nuestras actividades diarias, en cada decisión y acción del día a día.

Los niños deben tener un acompañamiento escolar o familiar para obtener un aprendizaje significativo y un desarrollo de su lógica (objetivos principales de la investigación) puesto que al no ser así, el videojuego puede convertirse en una distracción, haciendo que el infante se centre demasiado en el juego y no en el aprendizaje.

REFERENCIAS

ALCAIDE, E., 1996. “La interjección”, en Fuentes, C. y Alcaide, E. La expresión de la modalidad en el habla de Sevilla. Sevilla: Servicio de Publicaciones del Ayuntamiento de Sevilla.

ALCAIDE, E., 1996. “La interjección”, en Fuentes, C. y Alcaide, E. La expresión de lamodalidad en el habla de Sevilla. Sevilla: Servicio de Publicaciones del Ayuntamiento de Sevilla.

ALÉN, C., 2004. Localización do programa Infarwiew 3.85. Trabajo fin de carrera. Universidad de Vigo.

ALLAN, Bromley. Charles Babbage’sAnalyticalEngine. Annals of theHistory of Computing, Volume 4, Number 3, July 1982. [En línea]. [Consultado en Abril de 2013]. [Consultado el 15 de Mayo de 2013]. Disponible en: http://athena.union.edu/~hemmendd/courses/cs80/an-engine.pdf

ALLAN, Bromley. Charles Babbage’sAnalyticalEngine. Annals of theHistory of Computing, Volume 4, Number 3, July 1982. [En línea]. [Consultado en Abril de 2013]. [Consultado el 15 de Mayo de 2013]. Disponible en: http://athena.union.edu/~hemmendd/courses/cs80/an-engine.pdf

ALLAN, Bromley. Charles Babbage’sAnalyticalEngine. Annals of theHistory of Computing, Volume 4, Number 3, July 1982. [En línea]. [Consultado en Abril de 2013]. [Consultado el 15 de Mayo de 2013]. Disponible en: http://athena.union.edu/~hemmendd/courses/cs80/an-engine.pdf

APARICI, Roberto. (2006). Narrativa Digital. Madrid. Programa Modular en Tecnologías Digitales y Sociedad del Conocimiento de la U.N.E.D.

ARAUJO, E.(1998).Rendimiento Académico en la Asignatura Fundamentos de Biología y su Relación con el Programa de Estudio, los Promedios de Notas del Ciclo Diversificado y los Hábitos de Estudio. Tesis de Maestría, Universidad Pedagógica Experimental Libertador. Caracas http://es.wikipedia.org/wiki/Estudio

ARGOTE, José Ignacio. (2005). Bitácora. Crear con ordenador: control. Una experiencia en el VII Master de Informática Educativa de la U.N.E.D, en VV.AA. V Congreso Internacional Virtual de Educación. Cive. Recuperado de http:// www.cibereduca.com

ARDERIU, X., 2002. Llengua i localització. Revista Tradumàtica – Traducció i Tecnologies de la Informació i la Comunicació 01: La Localització. Disponible en URL: http://www.fti.uab.cat/tradumatica/revista [Fecha de consulta 19 de noviembre de 2009]

ARENCIBIA-Jorge, R., Leydesdorff, L., Chinchilla-Rodríguez, Z., Rousseau, R., & Paris, S. W. (2009). Retrieval of very large numbers of items in the Web of Science: an exercise to develop accurate search strategies. *El Profesional de la Información*, *18*(5), 529-533.

ARENCIBIA-Jorge, R., Leydesdorff, L., Chinchilla-Rodríguez, Z., Rousseau, R., & Paris, S. W. (2009). Retrieval of very large numbers of items in the Web of Science: an exercise to develop accurate search strategies. *El Profesional de la Información*, *18*(5), 529-533.

ARGOTE, José Ignacio. (2005). Bitácora. Crear con ordenador: control. Una experiencia en el VII Master de Informática Educativa de la U.N.E.D, en VV.AA. V Congreso nternacional Virtual de Educación. Cive. Recuperado de http:// www.cibereduca.com

ARRIZABALAGA, M.I., 2007. “Revisión de la teoría de los polisistemas. De los Estudios Literarios a la Teoría de la Catástrofe; Una mirada al gusto de Bo KampmannWalther” en Revista Observaciones Filosóficas. Nº5. Disponible en URL: http://www.observacionesfilosoficas.net/revisiondelateoriadelos.html [Fecha deconsulta: 6 de septiembre de 2009

AUSTERMÜHL, F., 2001. Electronictoolsfortranslators. Manchester: St Jerome Publishing.

AUSTERMÜHL, F., 2003. “Training translatorstolocalize” en Lozalization and Translator Training an online conference 20-29 November 2003. Disponible en URL: http://isg.urv.es/seminars/2003_localization_online/austermuehl.html [Fecha de consulta: 19 de noviembre de 2009]

ÁVILA, A., 1997. El doblaje. Madrid: Cátedra.

ÁVILA, A., 1997b. La censura del doblaje cinematográfico en España. Barcelona: Cims.

BALLARD, M., 1992. De Ciceron à Benjamin. Lille: PressesUniversitaires de Lille.

BARRERAS, P., 2003. “Localización de videojuegos” en Meristation. Disponible en URL: http://www.meristation.com/v3/des_articulo.php?id=4425&pic=GEN [Fecha de consulta: 27 de enero 2010]

BARROWS, H.S. (1986). A Taxonomy of problem-based learning methods, en Medical Education, 20/6, 481–486.

BASSNETT-McGUIRE, S., 1980. TranslationStudies, Londres: Routledge.

BLANK, Steve. (1997). Authentic instruction. In W.E. Blank & S. Harwell (Eds.), Promising practices for connecting high school to the real world (pp. 15–21). Tampa, FL: University of South Florida. (ERIC Document Reproduction Service No. ED407586)

BRINGUIER, Juan.C. Conversaciones con Piaget. Gedisa, Barcelona. 1977. [Consultado en el 10 de Marzo de 2013].

BRUCKMAN, Sears, Published in Te Human-ComputerInteractionHandbook: Fundamentals, Evolving Technologies, and EmergingApplications, HCI forKids. Lawrence Erlbaum and Associates, 2002. [En linea]. [Consultado el13 de Mayo de 2013] Disponible en: http://www.cc.gatech.edu/~asb/papers/hci-for-kids.pdf

CODE.ORG [Video]. Starring; Bill Gates, Mark Zuckerberg, will.i.am, Chris Bosh, Jack Dorsey, Tony Hsie, Drew Houston, Gabe Newell, RuchiSanghvi, Elena Silenok, Vanessa Hurst, and HadiPartovi: Whatmostschoolsdon'tteach.Realización: LesleyChilcott, Hadh and AliPartovi. USA: Org.org, Febrero de 2013. [En línea]. [Consultado el 15 de Marzo de 2013]. Disponible en: https://www.youtube.com/watch?feature=player_embedded&v=nKIu9yen5nc

CODE.ORG [Video]. Starring; Bill Gates, Mark Zuckerberg, will.i.am, Chris Bosh, Jack Dorsey, Tony Hsie, Drew Houston, Gabe Newell, RuchiSanghvi, Elena Silenok, Vanessa Hurst, and HadiPartovi: Whatmostschoolsdon'tteach.Realización: LesleyChilcott, Hadh and AliPartovi. USA: Org.org, Febrero de 2013. [En línea]. [Consultado el 15 de Marzo de

2013]. Disponible en:
https://www.youtube.com/watch?feature=player_embedded&v=nKIu9yen5nc

CODE.ORG [Video]. Starring; Bill Gates, Mark Zuckerberg, will.i.am, Chris Bosh, Jack Dorsey, Tony Hsie, Drew Houston, Gabe Newell, RuchiSanghvi, Elena Silenok, Vanessa Hurst, and HadiPartovi: Whatmostschoolsdon'tteach.Realización: LesleyChilcott, Hadh and AliPartovi. USA: Org.org, Febrero de 2013. [En línea]. [Consultado el 15 de Marzo de 2013]. Disponible en:
https://www.youtube.com/watch?feature=player_embedded&v=nKIu9yen5nc

CORTES BOHIGAS, M. (1988): «Rendimiento académico», en Diccionario de Ciencias de la Educación. Santillana, Madrid.

CUENCA LÓPEZ, José María (2006) Los juegos informáticos de simulación en la enseñanza y el aprendizaje de las Ciencias Sociales. *Iber. Didáctica de las Ciencias Sociales, Geografía e Historia. No. 30,* pp. 69 – 81.

DECROLY. Ovide. El Juego Educativo. Edit. Morata. 2° edición. Madrid. España. 2002.

EVANS, Richard. Jean Piaget, The man and his ideas. E.P. Dutton, New York.1973.[Consultado en el 10 de Marzo de 2013].

ENDERTON, Heberth (1972). A Mathematical Introduction to Logic. Academic Press.

GROS, Begoña (2008). Videojuegos y aprendizaje. Barcelona: Graó.

GROS, Begoña., & AGUAYOS, José. (2004) Pantallas, juegos y educación: La alfabetización digital en la escuela. Bilbao: Desclée de Brouwner

HARVEY Brian. Language ServeKids and ComputerScientists?Bringing “No Ceiling” toScratch: Can One.Constructionism 2010, Paris. ComputerScienceDivision, University of California, Berkeley.[Consultado el 15 Marzo de 2013].

JENKINS, Tony. OntheDifficulty of LearningtoProgram. 3rd Annual LTSN-ICS Conference, Loughborough University. School of Computing. University of Leeds. Leeds, UK. 2002. Consultado el 20 de Abril de 2013 Disponible en: http://books.google.com/books?id=XPC9ucFbddsC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false.

JENKINS, Tony. OntheDifficulty of LearningtoProgram. 3rd Annual LTSN-ICS Conference, Loughborough University. School of Computing. University of Leeds. Leeds, UK. 2002. Consultado el 20 de Abril de 2013 Disponible en: http://books.google.com/books?id=XPC9ucFbddsC&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false.

JOLIE, O’Dell, Whyyour 8-year-old should be coding. Venturebeat Magazine Online. [En línea].[Consultado el 13 de Abril de 2013]. Disponible en: http://venturebeat.com/2013/04/12/why-your-8-year-old-should-be-coding/

JOLIE, O’Dell, Whyyour 8-year-old should be coding. Venturebeat Magazine Online. [En línea].[Consultado el 13 de Abril de 2013]. Disponible en: http://venturebeat.com/2013/04/12/why-your-8-year-old-should-be-coding/

JOLIE, O’Dell, Whyyour 8-year-old should be coding. Venturebeat Magazine Online. [En línea].[Consultado el 13 de Abril de 2013]. Disponible en:

http://venturebeat.com/2013/04/12/why-your-8-year-old-should-be-coding/

KARRAY, Fakhreddine; ALEMZADEH, Milad; SALEH, Jamil Abou; ARAB, Mo Nours. Human-ComputerInteraction: OverviewonState of the Art. PatternAnalysis and Machine IntelligenceLab., International Journalon Smart Sesing and IntelligentSystems, Vol1, No,1, Marzo de 2008. [Consultado el13 de Mayo de 2013]

LLINÁS, María del Carmen. El Pensamiento Lógico en la Educación Infantil. Revista Digital, No,22, Febrero de 2006 . [En Línea]. [Consultado 13 de Abril de 2014]. Disponible en:http://www.juntadeandalucia.es/averroes/~cepco3/competencias/mates/infantil/pensamiento_logico_en_la_educacion_infantil.pdf

LEIVA, María del Carmen. El Pensamiento Lógico en la Educación Infantil. Revista Digital, No,22, Febrero de 2006 . [En Línea]. [Consultado 13 de Abril de 2014]. Disponible en:http://www.juntadeandalucia.es/averroes/~cepco3/competencias/mates/infantil/pensamiento_logico_en_la_educacion_infantil.pdf

MAHONEY, Michael S. Te History of Computing in theHistory of Technology.Program in History of Science, Princeton University, Princeton, NJ. Annals of theHistory of Computing 113-125, 10(1988). [Consultado el 11 de Mayo de 2013.] Disponible en: http://www.princeton.edu/~hos/mike/articles/hcht.pdf

“MILENIO” [Web]. Jugar Videojuegos tiene grandes beneficios: Especialistas. Monterey, 2012. [Consultado el 15 Marzo de 2013]. http://www.milenio.com/cdb/doc/noticias2011/215cf5be355aeb0b371d1f895d2247fc

DE MIGUEL, M. (2005). Metodologías de enseñanza para el desarrollo de competencias. Orientaciones para el profesorado universitario ante el Espacio Europeo de Educación Superior. Madrid: Alianza.

"SANGESH"[Web]. Importance of programming languages. Articlesbase. 2010. Consultado el 20 de Abril de 2013. Disponible en: http://www.articlesbase.com/web-design-articles/importance-of-programming-languages-2845476.html#

SQUIRE, Kurt. Video Games in Education. Comparative Media StudiesDepartment, 14N-205 Massachusetts Institute of Technology, Cambridge, MA. 02139 USA. [Consultado el 15 Marzo de 2013].

TED, Talks [Video] (2013). Starring; Mitch Resnik: Enseñemos a los niños a codificar.Realización: Talks, TedxGroup. USA: ted.com, Febrero de 2013. [En línea]. [Consultado en el 15 Marzo de 2013]. Disponible en: http://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code.html

PEARSON, G., YOUNG, A. (2002). Technicallyspeaking: Why all Americans need to know more about technology. National Academy of Engineering., &NationalResearch. Council (U.S.). Washington, D.C: NationalAcademyPress. [Consultado en el 10 de Marzo de 2013].

PRIETO, L. (2006). Aprendizaje activo en el aula universitaria: el caso del aprendizaje basado en problemas, en Miscelánea Comillas. Revista de Ciencias Humanas y Sociales Vol.64. Núm.124. Págs. 173-196 [Consultado en el 10 de Marzo de 2013].

PIAGET, Jean. (1952) Autobiography. [Autobiografía]. Clark University Press.

En E. Boring (Ed) History of psychology in autobiography. Vol. 4. Worcester, [Consultado en el 11 de Marzo de 2013].

PIAGET, Jean., INHELDER, Barbell, (1972). El nacimiento de la inteligencia en el niño. Madrid. [Consultado en el 11 de Marzo de 2013].

PIAGET, Jean e Inhelder, Barbell, (1981), Psicología del niño, Ediciones Mórula, S.A., España. [Consultado en el 11 de Marzo de 2013].

PIAGET, Jean., (1961) La formación del símbolo en el niño. México: F.C.E. pág.205 [Consultado en el 13 de Marzo de 2013].

RODRÍGUEZ, Leoncio. (1996). Aplicación de las NN.TT. en la rehabilitación del lenguaje y en la atención a la diversidad. La Gaveta. Disponible en: http://www.educa.rcanaria.es/cep_sc_tenerife/recursos/revista/gaveta1_jun96/aplic.htm. [Consultado en el 12 de Marzo de 2013]..

SANCHEZ Y NUÑEZ. Hábitos de estudio y rendimiento en EGB y B UP. Un estudio comparativo. Pontificia Universidad Católica de ChileRevista complutense de educación, ISSN 1130-2496, Vol. 2, Nº 1, 1991 , págs. 43-68

SHAFFER, D. W., Squire, K. R., Halverson, R., & Gee, J. P. (2005). Video Games and theFuture of Learning. Phi Delta Kappan, 87 (2), pp. 104-111.

SHAFFER, D. W. (2006). Epistemic frames for epistemic games. Computers and Education. 46 (3), pp. 223-234

STAGER, Gary. (2003), The Case for Computing. Snapshots: Educational Insights del Centro Thornburg . Lake Barrington, IL: Thornburg Center, [Consultado el

15 Marzo de 2013]. Disponible en: http://www.eduteka.org/modulos/9/272/224/1

UNIVERSIDAD DE GRANADA, (2001). Hábitos de Estudio. Ed. Gabinete Pedagógico. España.. Revista electrónica de investigación psicoeducativa N 8, vol4 (1) 2006, Pág. 35 – www.ugr.es/~ve/gpp.html gpp2@ugr.es.

VALLEJO, César. (2012) [Web]. MONOGRÁFICO: Aprendizaje por proyectos y TIC. Recursos Tic, Gobierno de España. Consultado el 21 de Abril de 2013. Disponible en: http://recursostic.educacion.es/observatorio/web/es/component/content/article/1057-aprendizaje-por-proyectos-y-tic?showall=1

ANEXOS

Anexo A. Index.html

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta http-equiv="x-ua-compatible" content="IE=Edge">
        <meta name="viewport" content="width=device-width, user-
scalable=no">
        <meta name="apple-mobile-web-app-capable" content="yes">
        <meta name="apple-mobile-web-app-status-bar-style"
content="black-translucent">
        <title>Ayuda a Caver & Nicolas</title>
        <script type="text/javascript" src="libs/enchant.js"></script>
        <script type="text/javascript"
src="libs/plugins/tl.enchant.js"></script>
        <script type="text/javascript" src="js/tags.js"></script>
        <script type="text/javascript" src="js/game.js"></script>
        <style type="text/css">
            body {
                margin: 0;
            }
            @font-face {
                font-family: 'StoneHinge';
                src: url('assets/font/StoneHinge.ttf');
            }
        </style>
    </head>
    <body>
    </body>
</html>
```

Anexo B. game.js

```
enchant();
window.onload = function(){
    var game = new Core(720,480);
    game.fps = 15;
    game.preload(missing_panel,cancel_button,pause_board,level_selection_
button,close_button,restart_level,main_background,boton_jugar,board,back_
button,check_image,stage_selection_background,accept_button,level1_stage_
selection_button,home_button,
        stage1_background,stage1_winter_dropzone,stage1_summer_dropzone,s
tage1_spring_dropzone,stage1_autum_dropzone,stage1_tree_panel,stage1_run_
button,
```

```
        stage1_winter_tree,stage1_summer_tree,stage1_spring_tree,stage1_a
utum_tree,stage1_congratulations,stage1_wrong,stage2_background,stage2_ca
ver_dropzone,stage2_nicolas_dropzone,
        stage2_wrong,stage2_congratulations,stage2_run_button,stage2_nico
las_llanta,stage2_nicolas_camiseta,stage2_nicolas_martillo,stage2_nicolas
_vela,stage2_caver_fuego,stage2_caver_martillo,stage2_caver_rueda,stage2_
caver_vestido,           stage3_background,stage3_dropzone,stage3_forward
,stage3_up,stage3_jump,stage3_bat,stage3_player,stage3_rock,stage3_water,
stage3_liana,dino);
    game.onload = function(){
//-----------------------------------------------------------------------
---
//      CLASES
//-----------------------------------------------------------------------
---
    var ImageObject = enchant.Class.create(enchant.Sprite,{
        initialize: function( src, width, height, x, y, scene ){
            this.width = width;
            this.height = height;
            enchant.Sprite.call(this, this.width, this.height);
            this.x = x;
            this.y = y;
            this.scene = scene;
            this.frame = 0;
            this.image = game.assets[src];
            scene.addChild(this);
        }
    });

    var ButtonObject = enchant.Class.create(enchant.Sprite,{
        initialize: function( src, width, height, x, y, scene ){
            this.width = width;
            this.height = height;
            enchant.Sprite.call(this, this.width, this.height);
            this.count = 0;
            this.x = x;
            this.y = y;
            this.scene = scene;
            this.frame = 0;
            this.image = game.assets[src];
            this.addEventListener(Event.TOUCH_START, function(){
                this.frame = 1;
            });
            this.addEventListener(Event.TOUCH_END, function(){
                this.frame = 0;
                //TODO cargar algo
            });
            scene.addChild(this);
        }
    });
```

```
    var Menu = enchant.Class.create(enchant.Sprite,{
        initialize: function( src, width, height, x, y, scene ){
            this.width = width;
            this.height = height;
            enchant.Sprite.call(this, this.width, this.height);
            this.x = x;
            this.y = y;
            this.scene = scene;
            this.frame = 0;
            this.image = game.assets[src];
            this.addEventListener(Event.TOUCH_START, function(){
                this.frame = 1;
            });
            this.addEventListener(Event.TOUCH_END, function(){
                this.frame = 0;
                var panel = new
ImageObject(pause_board,500,400,110,40,this.scene);
                var main = new
ButtonObject(level_selection_button,259,65,230.5,215,this.scene);
                main.addEventListener(Event.TOUCH_END,function(){
                    this.scene.removeChild(panel);
                    this.scene.removeChild(close);
                    this.scene.removeChild(this);
                    game.pushScene(stage_selection);
                });
                var close = new
ButtonObject(close_button,40,40,115,45,this.scene);
                close.addEventListener(Event.TOUCH_END,function(){
                    this.scene.removeChild(panel);
                    this.scene.removeChild(main);
                    this.scene.removeChild(this);
                });
                //TODO cargar algo
            });
            scene.addChild(this);
        }
    });

    var LevelButtonObject = enchant.Class.create(enchant.Sprite,{
        initialize: function( src, width, height, x, y, scene, stage ){
            var self = this;
            this.width = width;
            this.height = height;
            enchant.Sprite.call(this, this.width, this.height);
            this.x = x;
            this.y = y;
            this.scene = scene;
            this.stage = stage;
            this.image = game.assets[src];
```

```
            this.scene.addEventListener(Event.ENTER,function(){
                if(stage.unlocked==true){
                    this.frame = 0;
                }else{
                    this.frame = 2;
                }
                if(stage.finished==true){
                    this.frame = 3;
                }


            });
            this.addEventListener(Event.TOUCH_START, function(){
                this.frame = 1;
            });
            this.addEventListener(Event.TOUCH_END, function(){
                if(this.stage.finished==true){
                    var alert = new
ImageObject(restart_level,500,400,110,40,this.scene);
                    var score_label = new enchant.Label();
                    score_label.text = self.stage.score;
                    score_label.width = 400;
                    score_label.height = 64;
                    score_label.font = "24px 'StoneHinge'";
                    score_label.color = "#ffffff";
                    score_label.x = 285;
                    score_label.y = 108;
                    scene.addChild(score_label);
                    var accept = new
ButtonObject(accept_button,259,65,110,360,self.scene);
                    var cancel = new
ButtonObject(cancel_button,259,65,360,360,this.scene);
                    cancel.addEventListener(Event.TOUCH_END,function(){
                        this.scene.removeChild(alert);
                        this.scene.removeChild(score_label);
                        this.scene.removeChild(accept);
                        this.scene.removeChild(this);

                        self.frame = 0;
                    });
                    accept.addEventListener(Event.TOUCH_END,function(){
                        if(self.stage.name == "stage1"){
                            winter_dropzone.reset();
                            autum_dropzone.reset();
                            summer_dropzone.reset();
                            spring_dropzone.reset();
                            autum_tree.reset();
                            summer_tree.reset();
                            spring_tree.reset();
                            winter_tree.reset();
```

```
                    stage1_run.count = 0;
                    this.scene.removeChild(alert);
                    this.scene.removeChild(score_label);
                    this.scene.removeChild(cancel);
                    game.pushScene(self.stage);
                    this.scene.removeChild(this);
                }
                if(self.stage.name == "stage2"){
                    nicolas_dropzone.reset();
                    caver_dropzone.reset();
                    caver_martillo.reset();
                    caver_vestido.reset();
                    caver_rueda.reset();
                    caver_fuego.reset();
                    nicolas_camiseta.reset();
                    nicolas_vela.reset();
                    nicolas_martillo.reset();
                    nicolas_llanta.reset();
                    stage2_run.count=0;
                    this.scene.removeChild(alert);
                    this.scene.removeChild(score_label);
                    this.scene.removeChild(cancel);
                    game.pushScene(self.stage);
                    this.scene.removeChild(this);
                }
                if(self.stage.name == "stage3"){
                    player.reset();
                    dropzone_1.reset();
                    dropzone_2.reset();
                    dropzone_3.reset();
                    dropzone_4.reset();
                    dropzone_5.reset();
                    dropzone_6.reset();
                    liana.frame = 0;
                    stage3_run.count=0;
                    this.scene.removeChild(alert);
                    this.scene.removeChild(score_label);
                    this.scene.removeChild(cancel);
                    game.pushScene(self.stage);
                    this.scene.removeChild(this);
                }

            });

        }else{
            if(this.stage.unlocked){
                game.pushScene(this.stage);
            }
        }
    });
```

```
            scene.addChild(this);
        }
    });

    var DropzoneObject = enchant.Class.create(enchant.Sprite,{
        initialize: function( src, width, height, x, y, data, scene,name
){
            this.width = width;
            this.height = height;
            enchant.Sprite.call(this, this.width, this.height);
            this.x = x;
            this.y = y;
            this.init_x = x;
            this.init_y = y;
            this.name = name;
            this.full = false;
            this.intersected_object = null;
            this.data = data;
            this.scene = scene;
            this.frame = 0;
            this.image = game.assets[src];
            this.addEventListener(Event.ENTER_FRAME,function(){
                if(this.full == true){
                    this.frame = 1;
                }else{
                    this.frame = 0;
                }
            });
            scene.addChild(this);
        },
        reset: function(){
            this.x = this.init_x;
            this.y = this.init_y;
            this.full = false;
            this.intersected_object = null;
        }
    });

    var DropzoneObject3 = enchant.Class.create(enchant.Sprite,{
        initialize: function( src, width, height, x, y, scene){
            this.width = width;
            this.height = height;
            enchant.Sprite.call(this, this.width, this.height);
            this.x = x;
            this.y = y;
            this.init_x = x;
            this.init_y = y;
            this.full = false;
            this.intersected_object = null;
            this.data = null;
```

```
            this.scene = scene;
            this.frame = 0;
            this.image = game.assets[src];
            this.addEventListener(Event.ENTER_FRAME,function(){
                //console.log(this.data);
                if(this.full == true && this.intersected_object!=null){
                    this.data = this.intersected_object.data;
                }else{
                    this.data = null;
                }
            });
            scene.addChild(this);
        },
        reset: function(){
            this.x = this.init_x;
            this.y = this.init_y;
            this.full = false;
            this.intersected_object = null;
            this.data = null;
        }
    });

    var DropzoneObject2 = enchant.Class.create(enchant.Sprite,{
        initialize: function(src, width, height, x, y, data, scene){
            this.width = width;
            this.height = height;
            enchant.Sprite.call(this, this.width, this.height);
            this.x = x;
            this.y = y;
            this.init_x = x;
            this.init_y = y;
            this.data = data;
            this.scene = scene;
            this.frame = 0;
            this.image = game.assets[src];
            scene.addChild(this);
        },
        check: function(objects){
            var my_objects = new Array();
            for(i=0;i<objects.length;i++){
                if(this.intersect(objects[i])){
                    my_objects.push(objects[i].data);
                    console.log('aqui');
                }
            }
//ajustar la comparacion con el numero final de objetos para cada uno
            if(my_objects.length==4){
                for(i=0;i<my_objects.length;i++){
                    if(my_objects[i]!=this.data){
                        return false;
```

```
                        break;
                    }
                }
                return true;
            }else{
                return false;
            }
        },
        reset: function(){
            this.x = this.init_x;
            this.y = this.init_y;

        }
    });

    var DragableObjectStage2 = enchant.Class.create(enchant.Sprite,{
        initialize: function(src, width, height, x, y, data,
scene,dropzones){
            this.width = width;
            this.height = height;
            enchant.Sprite.call(this, this.width, this.height);
            this.x = x;
            this.y = y;
            this.init_x = x;
            this.init_y = y;
            this.collision = false;
            this.intersected_dropzone = null;
            this.scene = scene;
            this.data = data;
            this.frame = 0;
            this.image = game.assets[src];
            this.addEventListener(Event.TOUCH_MOVE,function(e){
                this.frame = 1;
                this.x = e.x - (this.width/2);
                this.y = e.y - (this.height/2);
            });
            this.addEventListener(Event.TOUCH_END,function(){
                this.frame = 0;
                for(i=0;i<dropzones.length;i++){
                    if(this.intersect(dropzones[i])){
                        this.collision = true;
                        this.intersected_dropzone = dropzones[i];
                        break;
                    }else{
                        this.collision = false;
                        this.intersected_dropzone = null;
                    }
                }

                if(this.collision==false){
```

```
                    this.x = this.init_x;
                    this.y = this.init_y;
                }

            })
            scene.addChild(this);
        },
        reset: function(){
            this.x = this.init_x;
            this.y = this.init_y;
            this.collision = false;
            this.intersected_dropzone = null;
            this.frame = 0;

        }
    });

    var DragableObjectStage3 = enchant.Class.create(enchant.Sprite,{
        initialize: function( src, width, height, x, y, data,
scene,dropzones){
            this.width = width;
            this.height = height;
            enchant.Sprite.call(this, this.width, this.height);
            this.x = x;
            this.y = y;
            this.init_x = x;
            this.init_y = y;
            this.scene = scene;
            this.data = data;
            this.collision = false;
            this.frame = 0;
            this.dropzones = dropzones;
            //this.check = false;
            //this.imagen_check = new
ImageObject(check_image,50,50,x,y,scene);
            //this.imagen_check.frame = 0;
            this.intersected_dropzone = null;
            this.image = game.assets[src];
            this.addEventListener(Event.TOUCH_MOVE,function(e){
                this.frame = 1;
                this.x = e.x - (this.width/2);
                this.y = e.y - (this.height/2);
            });
            this.addEventListener(Event.TOUCH_START,function(){
                window[this.data] = new
DragableObjectStage3(src,this.width,this.height,this.init_x,this.init_y,t
his.data,this.scene,this.dropzones);
            });
            this.addEventListener(Event.TOUCH_END,function(){
                this.frame = 0;
```

```
                for(i=0 ; i < dropzones.length ; i++ ){
                    if(this.within(dropzones[i], 30)){
                        if(this.intersected_dropzone == null){
                            this.collision = true;
                            this.intersected_dropzone = dropzones[i];
                            break;
                        }else{
                            this.intersected_dropzone.intersected_object
= null;
                            this.intersected_dropzone.full = false;
                            this.collision = true;
                            this.intersected_dropzone = dropzones[i];
                            break;
                        }
                    }else{
                        this.collision = false;
                    }
                }

                if(this.collision==false){

                    if(this.intersected_dropzone!=null){
                        if(this.intersected_dropzone.intersected_object
== this){
                            this.intersected_dropzone.full = false;
                            this.intersected_dropzone.intersected_object
= null;
                            this.intersected_dropzone = null;
                        }
                    }
                    this.scene.removeChild(this);
                }

                if(this.collision == true){
                    if(this.intersected_dropzone.full == true &&
this.intersected_dropzone.intersected_object!=this){
                        this.scene.removeChild(this);
                    }else if(this.intersected_dropzone.full == true &&
this.intersected_dropzone.intersected_object == this){
                        this.x = this.intersected_dropzone.x+13;
                        this.y = this.intersected_dropzone.y+6.5;
                    }else if(this.intersected_dropzone.full == false){
                        this.x = this.intersected_dropzone.x+13;
                        this.y = this.intersected_dropzone.y+6.5;
                        this.intersected_dropzone.full = true;
                        this.intersected_dropzone.intersected_object =
this;
                    }
                }
```

```
            });
            scene.addChild(this);
        },
        reset: function(){
            this.x = this.init_x;
            this.y = this.init_y;
            this.collision = false;
            this.intersected_dropzone = null;
        }
    });

    var DragableObjectStage1 = enchant.Class.create(enchant.Sprite,{
        initialize: function( src, width, height, x, y, data,
scene,dropzones ){
            this.width = width;
            this.height = height;
            enchant.Sprite.call(this, this.width, this.height);
            this.x = x;
            this.y = y;
            this.init_x = x;
            this.init_y = y;
            this.scene = scene;
            this.data = data;
            this.collision = false;
            this.frame = 0;
            this.check = false;
            this.imagen_check = new
ImageObject(check_image,50,50,x,y,scene);
            this.imagen_check.frame = 0;
            this.intersected_dropzone = null;
            this.image = game.assets[src];
            this.addEventListener(Event.TOUCH_MOVE,function(e){
                this.frame = 1;
                this.x = e.x - (this.width/2);
                this.y = e.y - (this.height/2);
            });
            this.addEventListener(Event.TOUCH_START,function(){
                this.imagen_check.frame = 0;
                //console.log(this.collision);
            });
            this.addEventListener(Event.TOUCH_END,function(){
                this.frame = 0;

                for(i=0 ; i < dropzones.length ; i++ ){
                    if(this.within(dropzones[i], 30)){
                        if(this.intersected_dropzone == null){
                            this.collision = true;
                            this.intersected_dropzone = dropzones[i];
                            break;
                        }else{
```

```
                            this.intersected_dropzone.intersected_object
= null;
                            this.intersected_dropzone.full = false;
                            this.collision = true;
                            this.intersected_dropzone = dropzones[i];
                            break;
                        }
                    }else{
                        this.collision = false;
                    }
                }

                if(this.collision==false){
                    this.x = this.init_x;
                    this.y = this.init_y;
                    if(this.intersected_dropzone!=null){
                        if(this.intersected_dropzone.intersected_object
== this){
                            this.intersected_dropzone.full = false;
                            this.intersected_dropzone.intersected_object
= null;
                            this.intersected_dropzone = null;
                        }
                    }
                }

                if(this.collision == true){
                    if(this.intersected_dropzone.full == true &&
this.intersected_dropzone.intersected_object!=this){
                        this.x = this.init_x;
                        this.y = this.init_y;
                    }else if(this.intersected_dropzone.full == true &&
this.intersected_dropzone.intersected_object == this){
                        this.x = this.intersected_dropzone.x+109.25;
                        this.y = this.intersected_dropzone.y+15;
                    }else if(this.intersected_dropzone.full == false){
                        this.x = this.intersected_dropzone.x+109.25;
                        this.y = this.intersected_dropzone.y+15;
                        this.intersected_dropzone.full = true;
                        this.intersected_dropzone.intersected_object =
this;
                    }
                }
            });
            scene.addChild(this);
        },
        run: function(){
            var self = this;

            this.imagen_check.x = self.intersected_dropzone.x+200;
```

```
            this.imagen_check.y = self.intersected_dropzone.y+100;
            if(this.data == this.intersected_dropzone.data){
                this.imagen_check.frame = 1;
                this.check = true;
            }else if(this.data != this.intersected_dropzone.data){
                this.imagen_check.frame = 2;
                this.check = false;
            }
        },
        reset: function(){
            this.x = this.init_x;
            this.y = this.init_y;
            this.collision = false;
            this.check = false;
            this.intersected_dropzone = null;
            this.imagen_check.frame = 0;

        }
    });

    var Player = enchant.Class.create(enchant.Sprite,{
        initialize: function(src, width, height, x, y, scene,obstacles){
            var self = this;
            this.width = width;
            this.height = height;
            enchant.Sprite.call(this, this.width, this.height);
            this.x = x;
            this.y = y;
            this.init_x = x;
            this.init_y = y;
            this.obstacles = obstacles;
            this.fwd = false;
            this.upp = false;
            this.up_count = 1;
            this.jumpp = false;
            this.jump_dist = -1;
            this.hit = false;
            this.current_x = 0;
            this.current_y = 0;
            this.liana = false;
            this.src = src;
            this.scene = scene;
            this.frame = 0;
            this.collision = false;
            this.checked = false;
            this.image = game.assets[src];
            this.forward_delay = 3000;
            this.up_delay = 3000;
            this.jump_delay = 3000;
            this.bat_delay = 3000;
```

```
            this.addEventListener(Event.ENTER_FRAME,function(){

                if(this.collision == false){

                    for(i=0;i<this.obstacles.length;i++){
                        if(this.intersect(this.obstacles[i])){

                            if(this.obstacles[i].name=='rock'){
                                if(this.jumpp != true){
                                    this.collision = true;
                                    var alert = new
ImageObject(board,500,400,110,40,this.scene);
                                    var alert_label = new
enchant.Label();
                                    alert_label.text = "Caver se golpeo
con la roca";
                                    alert_label.width = 400;
                                    alert_label.height = 64;
                                    alert_label.font = "24px
'StoneHinge'";
                                    alert_label.x = 260;
                                    alert_label.y = 70;
                                    this.scene.addChild(alert_label);
                                    var accept = new
ButtonObject(accept_button,259,65,230.5,360,this.scene);
                                    accept.addEventListener(Event.TOUCH_E
ND,function(){
                                        this.scene.removeChild(alert);
                                        this.scene.removeChild(alert_labe
l);
                                        self.reset();
                                        this.scene.removeChild(this);
                                    });
                                }
                            }

                            if(this.obstacles[i].name=='water'){
                                this.collision = true;
                                var alert = new
ImageObject(board,500,400,110,40,this.scene);
                                var alert_label = new enchant.Label();
                                alert_label.text = "Caver cayo al agua";
                                alert_label.width = 400;
                                alert_label.height = 64;
                                alert_label.font = "24px 'StoneHinge'";
                                alert_label.x = 260;
                                alert_label.y = 70;
                                this.scene.addChild(alert_label);
                                var accept = new
ButtonObject(accept_button,259,65,230.5,360,this.scene);
```

```
                                accept.addEventListener(Event.TOUCH_END,f
unction(){
                                    this.scene.removeChild(alert);
                                    this.scene.removeChild(alert_label);
                                    self.reset();
                                    this.scene.removeChild(this);
                                });
                            }



                        }
                    }

                    if (this.fwd == true) {
                        if (this.x<(this.current_x+88)){
                            this.x+=3;
                        }else{
                            this.frame = 0;
                        }
                    }

                    if(this.jumpp == true){
                        if(this.x<(this.current_x+109)){
                            var y_temp = 0;
                            var parabola = 0;

                            parabola = ((-
1)*(this.jump_dist)*(this.jump_dist)) + (4*(this.jump_dist)) + 5;

                            y_temp = ((54/9)*(parabola))+(1);

                            this.x += 3.67;
                            this.y = this.y + ((this.init_y-y_temp)-
this.y);
                            this.jump_dist += 0.2;
                        }else{
                            this.y = this.init_y;
                            this.x = this.current_x+110;
                            this.frame = 0;
                        }
                    }

                    if(this.upp == true){
                        if(this.up_count <15){
                            this.y -= 3;
                        }
                        if(this.up_count == 15){
                            if(this.intersect(this.obstacles[2])){
```

```
                                this.liana = true;
                            }else{
                                this.y -=3;
                            }
                        }
                        if(this.up_count > 15 && this.up_count < 30){
                            if(this.liana==true){
                                if(this.up_count == 16){
                                    this.obstacles[2].frame =
[1,2,3,4,5,6,7,8,9,10,11,12,13,14,14,14,14,14];
                                    this.frame = 0;
                                }
                                var y_temp = 0;
                                var parabola = 0;
                                parabola = ((-
1)*(this.jump_dist)*(this.jump_dist)) + (4*(this.jump_dist)) + 5;
                                y_temp = ((14/9)*(parabola))+(1);
                                this.x += 14.5;
                                this.jump_dist+=0.4;
                                this.y = this.y + (((this.init_y-
45)+y_temp)-this.y);
                            }else{
                                this.y +=3;
                            }
                        }
                        if(this.up_count >= 30 && this.up_count <= 35){
                            this.obstacles[2].frame = 14;
                            this.y += 7.05;
                        }

                        if(this.up_count == 36){
                            this.y = this.init_y;
                            this.frame = 0;
                        }
                        this.up_count++;
                    }

                    if(this.hit == true){
                        if(this.up_count <= 3){
                            if(this.up_count == 3){
                                this.frame = 1;
                            }
                            this.x += 3;
                        }
                        if(this.up_count == 3){
                            this.frame = 4;
                        }
                        if(this.up_count == 20){
                            this.frame = 0;
                            this.hit = false;
```

```
                }
                this.up_count++;
            }
        }

    });
    scene.addChild(this);
},
forward: function(){
    this.jumpp = false;
    this.upp = false;
    this.hit = false;
    this.frame = 1;
    this.current_x = this.x;
    this.fwd = true;
    //console.log("forward");
},
up: function(){
    this.fwd = false;
    this.jumpp = false;
    this.hit = false;
    this.frame = 2;
    this.jump_dist = -1;
    this.up_count = 1;
    this.upp = true;
    //console.log("up");
},
jump: function(){
    this.upp = false;
    this.fwd = false;
    this.hit = false;
    this.frame = 3;
    this.jump_dist = -1;
    this.current_x = this.x;
    this.current_y = this.y;
    this.jumpp = true;

},
bat: function(){
    this.upp = false;
    this.jumpp = false;
    this.fwd = false;
    this.up_count = 1;
    this.hit = true;
    //console.log("bat");
},
reset: function(){
    this.x = this.init_x;
    this.y = this.init_y;
    this.checked = false;
```

```
            this.fwd = false;
            this.upp = false;
            this.jumpp = false;
            this.hit = false;
            this.frame = 0;
            this.collision = false;
        }
    });

//-----------------------------------------------------------------------
---
//      FIN CLASES
//-----------------------------------------------------------------------
---

//***********************************************************************
***

//-----------------------------------------------------------------------
---
//      SCENES
//-----------------------------------------------------------------------
---
        var main = new Scene;
        main.name = "main";
        var stage_selection = new Scene;
        stage_selection.name = "stage_selection";
        var stage1 = new Scene;
        stage1.name = "stage1";
        stage1.unlocked = true;
        stage1.finished = false;
        var stage2 = new Scene;
        stage2.name = "stage2";
        stage2.unlocked = true;
        stage2.finished = false;
        var stage3 = new Scene;
        stage3.name = "stage3";
        stage3.unlocked = true;
        stage2.finished = false;
//-----------------------------------------------------------------------
---
//      FIN SCENES
//-----------------------------------------------------------------------
---

//***********************************************************************
***

//-----------------------------------------------------------------------
---
```

```
//      MAIN
//-----------------------------------------------------------------------
---
        var background_main = new
ImageObject(main_background,720,480,0,0,main);
        main.addEventListener(Event.ENTER_FRAME,function(){
            //console.log("aqui");
            if(background_main.frame == 0){
                background_main.frame =
[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19];
            }else if(background_main.frame == 19){
                var jugar = new
ButtonObject(boton_jugar,200,150,500,310,main);
                jugar.addEventListener(Event.TOUCH_END,function(){
                    game.pushScene(stage_selection);
                });
                background_main.frame =
[20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,
44];
            }
        });

        game.pushScene(main);

//-----------------------------------------------------------------------
---
//      FIN MAIN
//-----------------------------------------------------------------------
---

//***********************************************************************
***

//-----------------------------------------------------------------------
---
//      STAGE SELECTION
//-----------------------------------------------------------------------
---
        var background = new
ImageObject(stage_selection_background,720,480,0,0,stage_selection);
        var back = new
ButtonObject(back_button,104,107,5,5,stage_selection);
        back.addEventListener(Event.TOUCH_END,function(){
            game.pushScene(main);
        });
        var level1 = new
LevelButtonObject(level1_stage_selection_button,200,125,100,100,stage_sel
ection,stage1);
```

```
        var level2 = new
LevelButtonObject(level1_stage_selection_button,200,125,380,200,stage_sel
ection,stage2);
        var level3 = new
LevelButtonObject(level1_stage_selection_button,200,120,100,300,stage_sel
ection,stage3);
//-----------------------------------------------------------------------
---
//      FIN STAGE SELECTION
//-----------------------------------------------------------------------
---

//***********************************************************************
***

//-----------------------------------------------------------------------
---
//      STAGE 1
//-----------------------------------------------------------------------
---
        var background = new
ImageObject(stage1_background,720,480,0,0,stage1);
        var menu = new Menu(home_button,58,58,10,5,stage1);
        var winter_dropzone = new
DropzoneObject(stage1_winter_dropzone,346,151,14,65,"winter",stage1,'wint
er');
        var summer_dropzone = new
DropzoneObject(stage1_summer_dropzone,346,151,360,65,"summer",stage1,'sum
mer');
        var spring_dropzone = new
DropzoneObject(stage1_spring_dropzone,346,151,14,216,"spring",stage1,'spr
ing');
        var autum_dropzone = new
DropzoneObject(stage1_autum_dropzone,346,151,360,216,"autum",stage1,'autu
m');
        var dropzone_array_stage1 = new
Array(winter_dropzone,spring_dropzone,autum_dropzone,summer_dropzone);

        var tree_panel = new
ImageObject(stage1_tree_panel,614,138,0,342,stage1);
        var summer_tree = new
DragableObjectStage1(stage1_summer_tree,127.5,121,16,350,"summer",stage1,
dropzone_array_stage1);
        var autum_tree = new
DragableObjectStage1(stage1_autum_tree,127.5,121,163,350,"autum",stage1,d
ropzone_array_stage1);
        var winter_tree = new
DragableObjectStage1(stage1_winter_tree,127.5,121,311,350,"winter",stage1
,dropzone_array_stage1);
```

```
        var spring_tree = new
DragableObjectStage1(stage1_spring_tree,127.5,121,458,350,"spring",stage1
,dropzone_array_stage1);
        var objects_array_stage1 = new
Array(summer_tree,autum_tree,winter_tree,spring_tree);
        var stage1_run = new
ButtonObject(stage1_run_button,104,107,608,375,stage1);
        stage1_run.addEventListener(Event.TOUCH_END,function(){
            var sin_collision = 0;
            this.count +=1;

            for(i=0;i<objects_array_stage1.length;i++){
                if(objects_array_stage1[i].collision == false){
                    sin_collision +=1;
                }
            }

            if(sin_collision!=0){
                this.count += (0.2*sin_collision);
                var alert = new
ImageObject(missing_panel,500,400,110,40,stage1);
                var alert_label = new enchant.Label();
                alert_label.text = sin_collision;
                alert_label.width = 400;
                alert_label.height = 64;
                alert_label.font = "25px 'StoneHinge'";
                alert_label.color = "#EA6D3B";
                alert_label.x = 340;
                alert_label.y = 163;
                stage1.addChild(alert_label);
                var accept = new
ButtonObject(accept_button,259,65,230.5,360,stage1);
                accept.addEventListener(Event.TOUCH_END,function(){
                    this.scene.removeChild(alert);
                    this.scene.removeChild(alert_label);
                    this.scene.removeChild(this);
                });

            }else{
                summer_tree.run();
                autum_tree.run();
                winter_tree.run();
                spring_tree.run();
                if(summer_tree.check == true && autum_tree.check == true
&& spring_tree.check == true && winter_tree.check == true){
                    var alert = new
ImageObject(stage1_congratulations,500,400,110,40,stage1);
                    stage1.score = 2000 - ((this.count-1)*100);
                    var score_label = new enchant.Label();
                    score_label.text = stage1.score;
```

```
                    score_label.width = 400;
                    score_label.height = 64;
                    score_label.font = "36px 'StoneHinge'";
                    score_label.color = "#EA6D3B";
                    score_label.x = 315;
                    score_label.y = 241;
                    stage1.addChild(score_label);
                    var accept = new
ButtonObject(accept_button,259,65,230.5,360,stage1);
                    accept.addEventListener(Event.TOUCH_END,function(){
                        this.scene.removeChild(alert);
                        this.scene.removeChild(score_label);
                        this.scene.removeChild(this);
                        stage1.finished = true;
                        stage2.unlocked = true;
                        game.pushScene(stage_selection);
                    });
                }else{
                    var alert = new
ImageObject(stage1_wrong,500,400,110,40,stage1);
                    var accept = new
ButtonObject(accept_button,259,65,230.5,360,stage1);
                    accept.addEventListener(Event.TOUCH_END,function(){
                        this.scene.removeChild(alert);
                        this.scene.removeChild(this);
                    });
                }

            }



        });


//-----------------------------------------------------------------------
---
//      FIN STAGE 1
//-----------------------------------------------------------------------
---

//***********************************************************************
***

//-----------------------------------------------------------------------
---
//      STAGE 2
//-----------------------------------------------------------------------
---
```

```
        var background = new
ImageObject(stage2_background,720,480,0,0,stage2);
        var menu = new Menu(home_button,58,58,10,5,stage2);
        var caver_dropzone = new
DropzoneObject2(stage2_caver_dropzone,200,380,30,80,1,stage2);
        var nicolas_dropzone = new
DropzoneObject2(stage2_nicolas_dropzone,200,380,490,80,2,stage2);
        var dropzone_array_stage2 = new
Array(caver_dropzone,nicolas_dropzone);


        //var caver_1 = new
DragableObjectStage2(stage2_caver_1,20,20,270,90,1,stage2,dropzone_array_
stage2);
        var caver_fuego = new
DragableObjectStage2(stage2_caver_fuego,75,71,280,220,1,stage2,dropzone_a
rray_stage2);

        //var nicolas_1 = new
DragableObjectStage2(stage2_nicolas_2,20,20,370,130,2,stage2,dropzone_arr
ay_stage2);
        var nicolas_llanta = new
DragableObjectStage2(stage2_nicolas_llanta,75,71,280,110,2,stage2,dropzon
e_array_stage2);
        //var nicolas_2 = new
DragableObjectStage2(stage2_nicolas_2,20,20,370,90,2,stage2,dropzone_arra
y_stage2);
        var nicolas_camiseta = new
DragableObjectStage2(stage2_nicolas_camiseta,75,71,370,100,2,stage2,dropz
one_array_stage2);
        var nicolas_martillo = new
DragableObjectStage2(stage2_nicolas_martillo,75,71,370,200,2,stage2,dropz
one_array_stage2);
        var nicolas_vela = new
DragableObjectStage2(stage2_nicolas_vela,75,71,370,300,2,stage2,dropzone_
array_stage2);

        //var caver_2 = new
DragableObjectStage2(stage2_caver_1,20,20,270,130,1,stage2,dropzone_array
_stage2);
        var caver_martillo = new
DragableObjectStage2(stage2_caver_martillo,75,71,280,320,1,stage2,dropzon
e_array_stage2);
        var caver_rueda = new
DragableObjectStage2(stage2_caver_rueda,75,71,280,220,1,stage2,dropzone_a
rray_stage2);
        var caver_vestido = new
DragableObjectStage2(stage2_caver_vestido,75,71,280,220,1,stage2,dropzone
_array_stage2);
```

```
        var objects_array_stage2 = new
Array(caver_fuego,nicolas_llanta,caver_martillo,nicolas_camiseta,caver_ru
eda,nicolas_martillo,caver_vestido,nicolas_vela);

        var stage2_run = new
ButtonObject(stage2_run_button,77,71,623,0,stage2);
        stage2_run.addEventListener(Event.TOUCH_END,function(){
            this.count+=1;
            var caver_ready = caver_dropzone.check(objects_array_stage2);
            var nicolas_ready =
nicolas_dropzone.check(objects_array_stage2);
            var sin_collision = 0;
            for(i=0;i<objects_array_stage2.length;i++){
                //console.log(objects_array_stage2[i]);
                if(objects_array_stage2[i].collision ==false){
                    sin_collision++;
                }
            }
            if(sin_collision!=0){
                this.count += (0.2*sin_collision);
                var alert = new
ImageObject(missing_panel,500,400,110,40,stage2);
                var alert_label = new enchant.Label();
                alert_label.text = sin_collision;
                alert_label.width = 400;
                alert_label.height = 64;
                alert_label.font = "25px 'StoneHinge'";
                alert_label.color = "#EA6D3B";
                alert_label.x = 340;
                alert_label.y = 163;
                stage2.addChild(alert_label);
                var accept = new
ButtonObject(accept_button,259,65,230.5,360,stage2);
                accept.addEventListener(Event.TOUCH_END,function(){
                    this.scene.removeChild(alert);
                    this.scene.removeChild(alert_label);
                    this.scene.removeChild(this);
                });
            }else{
                if(caver_ready && nicolas_ready){
                    var alert = new
ImageObject(stage1_congratulations,500,400,110,40,stage2);
                    stage2.score = 2000 - ((this.count-1)*100);
                    var score_label = new enchant.Label();
                    score_label.text = stage2.score;
                    score_label.width = 400;
                    score_label.height = 64;
                    score_label.font = "24px 'StoneHinge'";
                    score_label.x = 315;
                    score_label.y = 241;
```

```
                    stage2.addChild(score_label);
                    var accept = new
ButtonObject(accept_button,259,65,230.5,360,stage2);
                    accept.addEventListener(Event.TOUCH_END,function(){
                        this.scene.removeChild(alert);
                        this.scene.removeChild(score_label);
                        this.scene.removeChild(this);
                        stage2.finished = true;
                        stage3.unlocked = true;
                        game.pushScene(stage_selection);
                    });
                }else{
                    var alert = new
ImageObject(stage1_wrong,500,400,110,40,stage2);
                    var accept = new
ButtonObject(accept_button,259,65,230.5,360,stage2);
                    accept.addEventListener(Event.TOUCH_END,function(){
                        this.scene.removeChild(alert);
                        this.scene.removeChild(this);
                    });
                }
            }
        });
//-----------------------------------------------------------------------
---
//      FIN STAGE 2
//-----------------------------------------------------------------------
---

//***********************************************************************
***

//-----------------------------------------------------------------------
---
//      STAGE 3
//-----------------------------------------------------------------------
---
        var background = new
ImageObject(stage3_background,720,480,0,0,stage3);
        var menu = new Menu(home_button,58,58,10,5,stage3);
        var dropzone_1 = new
DropzoneObject3(stage3_dropzone,88,50,52,430,stage3);
        var dropzone_2 = new
DropzoneObject3(stage3_dropzone,88,50,140,430,stage3);
        var dropzone_3 = new
DropzoneObject3(stage3_dropzone,88,50,228,430,stage3);
        var dropzone_4 = new
DropzoneObject3(stage3_dropzone,88,50,316,430,stage3);
        //var dropzone_5 = new
DropzoneObject3(stage3_dropzone,88,50,404,430,stage3);
```

```
        var dropzone_5 = new
DropzoneObject3(stage3_dropzone,88,50,492,430,stage3);
        var dropzone_6 = new
DropzoneObject3(stage3_dropzone,88,50,580,430,stage3);
        var dropzone_array_stage3 = new
Array(dropzone_1,dropzone_2,dropzone_3,dropzone_4,dropzone_5,dropzone_6);

        var rock = new ImageObject(stage3_rock,50,50,159,300,stage3);
        rock.name = 'rock';
        var water = new ImageObject(stage3_water,150,10,355,340,stage3);
        water.name = 'water';
        var liana = new ImageObject(stage3_liana,180,242,327,0,stage3);
        liana.frame = 0;
        liana.name = 'liana';
        var stage3_obstacles_array = new Array(rock,water,liana);

        var forward = new
DragableObjectStage3(stage3_forward,65,39,648,10,'forward',stage3,dropzon
e_array_stage3);
        var up = new
DragableObjectStage3(stage3_up,65,39,648,57,'up',stage3,dropzone_array_st
age3);
        var jump = new
DragableObjectStage3(stage3_jump,65,39,648,104,'jump',stage3,dropzone_arr
ay_stage3);
        var bat = new
DragableObjectStage3(stage3_bat,65,39,648,151,'bat',stage3,dropzone_array
_stage3);


        var enemy = new ImageObject(dino,80,150,630,200,stage3);
        var player = new Player(stage3_player, 50, 100, 15, 250,
stage3,stage3_obstacles_array);

        var stage3_run = new
ButtonObject(stage1_run_button,104,107,360,0,stage3);
        stage3_run.addEventListener(Event.TOUCH_END,function(){
            this.count += 1;
            var sin_collision = 0;

            for(i=0;i<dropzone_array_stage3.length;i++){
                if(dropzone_array_stage3[i].data == null){
                    sin_collision ++;
                }
            }
            if(sin_collision!=0){
                this.count += (0.2*sin_collision);
                var alert = new ImageObject(board,500,400,110,40,stage3);
                var alert_label = new enchant.Label();
```

```
                alert_label.text = "faltan " + sin_collision.toString() +
" objeto(s) por organizar";
                alert_label.width = 400;
                alert_label.height = 64;
                alert_label.font = "24px 'StoneHinge'";
                alert_label.x = 260;
                alert_label.y = 70;
                stage3.addChild(alert_label);
                var accept = new
ButtonObject(accept_button,259,65,230.5,360,stage3);
                accept.addEventListener(Event.TOUCH_END,function(){
                    this.scene.removeChild(alert);
                    this.scene.removeChild(alert_label);
                    this.scene.removeChild(this);
                });
            }else{

                var delay = 0;
                var nextDelay = 0;

                for(i=0;i<dropzone_array_stage3.length;i++){

                    if(i != 0){
                        // var foo = (condition) ? true : false;
                        nextDelay = (i != dropzone_array_stage3.length -
1) ? player[dropzone_array_stage3[i+1].data+"_delay"] :
player[dropzone_array_stage3[i].data+"_delay"];
                        delay += (nextDelay > 0) ? nextDelay : 0;
                    }
                    //console.log("Ejecutando metodo:
"+player[dropzone_array_stage3[i].data]+" con un delay acomulado de:
"+delay);
                    check =
executeMethod(player,dropzone_array_stage3[i].data,'',delay);
                }
            }

        enemy.addEventListener(Event.ENTER_FRAME,function(){
            if(this.intersect(player) && player.hit==true){
                var winner = setTimeout(function(){

                    var alert = new
ImageObject(stage1_congratulations,500,400,110,40,stage3);
                    stage3.score = 2000 - ((stage3_run.count-1)*100);
                    var score_label = new enchant.Label();
                    score_label.text = stage3.score;
                    score_label.width = 400;
                    score_label.height = 64;
                    score_label.font = "24px 'StoneHinge'";
                    score_label.x = 315;
```

```
                    score_label.y = 241;
                    stage3.addChild(score_label);
                    var accept = new
ButtonObject(accept_button,259,65,230.5,360,stage3);
                    accept.addEventListener(Event.TOUCH_END,function(){
                        this.scene.removeChild(alert);
                        this.scene.removeChild(score_label);
                        this.scene.removeChild(this);
                        stage3.finished = true;
                        //stage3.unlocked = true;
                        game.pushScene(stage_selection);
                    });
                },1000);
            }
        });

//-----------------------------------------------------------------------
---
//      FIN STAGE 3
//-----------------------------------------------------------------------
---


        });

        //FUNCTIONS
        function executeMethod(player,method,params,delayMS){

            //setTiemout(metodo,delay,parametros);
            var timer = setTimeout(function(){

                            //console.log("true");
                            if(player.collision == false){
                                player[method](params);
                            }else{
                                clearTimeout(timer);
                            }

                            return true;

                },delayMS);
        }


    };
    game.start();
};
```

Anexo C. Entrevistas a los niños

Con el único fin de conocer la experiencia de los jugadores se hicieron preguntas a los niños después de utilizar el videojuego.

Parámetros de respuesta: Usuario 1 (U1, U(n)). Sexo Masculino (M), Sexo Femenino (F)

1. **¿Te gustó el videojuego?**

U1 (F). R / Sí, me gustó pero le falta movimiento a los personajes

U2 (M). R / Sí, es chévere la historia.

U3 (M). R / Sí, es bueno.

U4 (F). R / Un poco, ese tipo de juegos no me gustan.

U5 (M). R / Me gusto la parte de ayudar a mi amigo en el juego, y si es interesante

U6 (F). R / Me encantó el juego, es muy divertido.

U7 (F). R / No me gustó el juego, es aburrido.

U8 (M). R / Me gustan otros juegos conocidos.

U9 (M). R/ Al principio es emocionante, pero en la última parte de los pasos es difícil.

U10 (M). R / Sí pero, me gustaría que tuviera más niveles, los que hay son pocos.

2. **¿Te gustaron los personajes de la historia?**

U1 (F). R / Sí, me son bonitos

U2 (M). R / Sí pero deberían ser más personajes y moverse más.

U3 (M). R / Los personajes son muy chéveres.

U4 (F). R / Los personajes con diferente ropa me gustaron mucho.

U5 (M). R / Sí, son dibujos, me agradan.

U6 (F). R / Los personajes si parecen de la historia, son chistosos.

U7 (F). R / No entendí los personajes hasta que la profesora nos explicó quienes eran.

U8 (M). R / No, los personajes deberían ser como Doki, Spider Man, y esos así.

U9 (M). R/ Sí, me gustó, porque parecía de la prehistoria.

U10 (M). R / No me gustaron, porque se veían feos.

3. ¿Te agradan este tipo de videojuegos?

U1 (F). R / Si, agrandan son divertidos

U2 (M). R / Sí. Jugar me gusta mucho, todos los videojuegos.

U3 (M). R / Sí son buenos para pensar.

U4 (F). R / No, me gustan más los de deportes y con movimiento.

U5 (M). R / No, prefiero los juegos de vestir los personajes.

U6 (F). R / Sí, aunque me gustan los juegos de curar las heridas más.

U7 (F). R / Sí me agradan, porque son divertidos.

U8 (M). R / No, son difíciles.

U9 (M). R/ Los videojuegos como estos no me gustan, me gustan más como Jet Pack.

U10 (M). R / Un poco, me gustan más los de cartas.

4. ¿Te dio curiosidad por jugar la historia en el videojuego?

U1 (F). R / Si, al principio me interesé por jugarlo.

U2 (M). R / Sí. Cuando nos dijeron que íbamos a jugar me dieron ganas de conocer el juego

U3 (M). R / Sí apenas vi el botón de iniciar quise empezar a jugar

U4 (F). R / Los videojuegos no me gustan tanto, no me dio tante curiosidad

U5 (M). R / Quise conocer los personajes y los niveles apenas me contaron la historia del juego.

U6 (F). R / Me dio tanta curiosidad que hundí muchas veces el botón para empezar

U7 (F). R / Si, me imagine los personajes antes de empezar y si eran como los había pensado.

U8 (M). R / No, son difíciles.

U9 (M). R/ Los videojuegos como estos no me gustan, me gustan más como Jet Pack Ride.

U10 (M). R / Un poco, me gustan más los de cartas.

5. ¿Te parecieron apropiados los botones y elementos del videojuego?

U1 (F). R / Me gustaron, me dejaron funcionar en el videojuego

U2 (M). R / Los botones son apropiados y similares a los de otras aplicaciones y por eso fue fácil reconocerlos

U3 (M). R / Sí, los botones significan lo que voy a hacer y los objetos son fáciles de arrastrar.

U4 (F). R / Sí, pero los 'pasos' del nivel 3 están un poco alejados de las zona de arrastre.

U5 (M). R / Sí aunque el botón de la cueva no lo reconocí fácilmente

U6 (F). R / No, algunos botones eran confusos.

U7 (F). R / Los colores de los botones son los indicados y hacen lo que quiero.

U8 (M). R / Después de que la profesora explicó los botones los entendí

U9 (M). R/ Me parece fácil que se utilice solo el mouse para arrastrar los objetos, los botones están bien.

U10 (M). R / Sí pero los botones del nivel 3 no los pude identificar, necesité ayuda de la profesora.

6. ¿Pudiste dar las instrucciones al personaje del tercer nivel?

U1 (F). R / Sí, para llegar donde el Dinosaurio debí utilizar todos los pasos, y me equivoqué en unos.

U2 (M). R / En el tercer nivel no pude, me parece que donde se ponen los pasos de las ordenes que se le da al personaje, debe tener una línea que muestre en que parte del escenario se aplica la acción.

U3 (M). R / Sí, con la ayuda de la profesora los pude seguir.

U4 (F). R / Sí, todos las instrucciones del tercer nivel las hice bien.

U5 (M). R / Sí, pero toco pensar mucho antes de poner los pasos.

U6 (F). R / Las instrucciones que le di al cavernícola fueron erradas, intenté muchas veces hasta que lo logré.

U7 (F). R / No pude darle las instrucciones solo, pero le pedí ayuda a la profesora y lo pude terminar.

U8 (M). R / Primero sí, me equivoqué en un solo paso pero al final lo logré, y salvé a Nícolas.

U9 (M). R/ No me gustó tanto el nivel 3 porque era un poco difícil. No pude darle correctamente las instrucciones. La profesora me ayudó.

U10 (M). R / Sí, pude arrastrar los pasos y darle las instrucciones. Aunque me equivoqué varias veces pude salvar a Nícolas.

7. ¿Crees que aprendiste jugando? ¿Qué aprendiste?

U1 (F). R / Sí, me parece que aprendí muchas cosas, como las casas de los cavernícolas y los atuendos de ellos. También a diferenciar entre los objetos de un niño de esta época y el del cavernícola. Además aprendí a darle instrucciones al computador para que el cavernícola hiciera lo que yo quisiera.

U2 (M). R / Sí, aprendí las figuras y formas de los cavernícolas. Yo le enseñé al cavernícola a pasar los obstáculos para salvar al dinosaurio.

U3 (M). R / Sí, aprendí porque me tocó pensar para poner las figuras donde eran. También para separar los objetos de cada uno. En el último nivel pude aprender a solucionar un problema.

U4 (F). R / Sí, aprendí a hacer un algoritmo que hacía que el cavernícola salvara al dinosaurio. Y a identificar los objetos de cada uno.

U5 (M). R / Sí, pero me divertí más. No aprendí nada nuevo, sólo de los cavernícolas.

U6 (F). R / Sí, aprendí y le enseñé al cavernícola todo.

U7 (F). R / Sí, aprendí a salvar al niño de los dinosaurios

U8 (M). R / Sí, aprendí porque seleccionar y diferenciar los objetos, a agruparlos y a juntarlos. También salvé a Nícolas, porque me inventé la solución con los pasos.

U9 (M). R/ No aprendí mucho del videojuego, porque no me explicaba nada.

U10 (M). R / Sí, aprendí a darle órdenes al programa para que realizara algo, como salvar al amigo del Cavernícola.

Anexo D. Encuestas para los docentes

Encuesta del video juego educativo
"Ayuda a Caver & Nicolas"

La presente encuesta tiene por objetivo evaluar el videojuego educativo **"Ayuda a Caver & Nicolas"**. Está dirigida a docentes de Tecnología, o relacionados al tema. Es importante que esta encuesta sea aplicada luego que Ud, haya explorado, interactuado y/o jugado detenidamente el videojuego.

Nota:
Es importante resaltar, que el videojuego es un producto actualmente en desarrollo. Usted ha probado una versión no totalmente terminada, que incluye las funciones principales.
Información general
Nombre del encuestado
(Opcional):_____
__
Marque con X las opciones que se adapten a su perfil

Ocupación	
	Profesor
	Estudiante
	Desarrollador de Software

Área	
	Tecnología
	Lógica
	Intro. Programación

Experiencia en el uso de video juegos	
	Nunca
	Ocasionalmente
	Una vez a la semana
	Varios días a la semana

Dispositivo favorito para el uso de videojuegos casuales	
	Ordenador
	Smartphone
	Tablet
	Otro _____

Instrucciones: Evalúe cada afirmación de las tablas usando el siguiente criterio:

1. **Muy de acuerdo**
2. **De acuerdo**
3. **Neutro**
4. **En desacuerdo**
5. **Muy en desacuerdo**

1. Con respecto a la Historia del videojuego. Marque con una X su selección.

Pregunta	1	2	3	4	5
El videojuego desarrolla su historia de forma clara y entendible para el jugado, con la explicación de la profesora					
El videojuego incita al jugador a conocer los temas básicos de la lógica y el proceso básico de la programación a través de situaciones cotidianas.					
El videojuego reta al jugador a resolver los problemas lógicos a través de la historia del juego.					
Los términos e instrucciones del juego son entendibles por el jugador.					

2. Con respecto al diseño del videojuego. Marque con una X su selección.

Pregunta	1	2	3	4	5
Los personajes del videojuego se asociados con la historia					
Las animaciones de las acciones son fluidas y semejantes a la realidad					
Los escenarios presentan los entornos de la historia					
El diseño en general, de los ítems y los botones es adecuado					

3. Con respecto a la interactividad del videojuego. Marque con una X su selección.

Pregunta	1	2	3	4	5
El jugador entiende de forma clara e intuitiva las acciones que debe realizar en el juego.					
El jugador reconoce el efecto de sus acciones en el videojuego.					
El jugador interactúa fácilmente con los objetos del videojuego.					
El jugador puede retroceder acciones para corregir sus decisiones.					
El jugador puede interactuar con los objetos fácilmente a través del ratón ó sus dedos.					
El videojuego es de fácil interacción usando el ratón ó los dedos.					
El jugador reconoce intuitivamente los objetos con los que puede interactuar.					

El jugador comprende con claridad como debe interactuar con los objetos del videojuego.					
Se proveen mecanismos que facilitan la interacción del usuario con el software.					

4. Con respecto al contenido relacionado con la lógica matemática y el proceso básico de la programación. Marque con una X su selección.

Pregunta	1	2	3	4	5
A través del videojuego se utilizan, manifiestan o demuestran conceptos de la lógica matemática y la algoritmia.					
Los conceptos utilizados son coherentes con el contenido de la lógica matemática y la algoritmia.					
Los conceptos de la lógica matemática y la algoritmia son coherentes con el flujo del videojuego.					
La metáfora “organizar secuencia de pasos”, utilizada en el videojuego se relaciona con algún concepto de algoritmia.					
El jugador requiere conocimientos previos en la lógica matemática y la algoritmia para desarrollar las acciones del videojuego.					
El jugador reconoce algún concepto de la lógica matemática y la algoritmia a través del videojuego.					
A través del videojuego se obtiene un mejor entendimiento de conceptos de lógica matemática y algoritmia.					
El uso de este videojuego puede ser útil para reforzar conocimientos de la lógica matemática y el proceso básico de la programación.					

5. Con respecto a la interfaz del videojuego. Marque con una X su selección.

Pregunta	1	2	3	4	5
Las interfaces usadas proveen mecanismos que facilitan la interacción con el videojuego.					
Las interfaces usadas proveen simbología estándar de videojuegos.					
La selección de colores para el videojuego es agradable visualmente.					
La presentación de los niveles del videojuego es adecuada, dando un correcto manejo del espacio de trabajo tanto para objetos del juego como para botones y menús.					

6. Otros aspectos. Marque con X su selección.

Pregunta	1	2	3	4	5
Se utiliza el audio adecuadamente en el videojuego.					
El tiempo de carga del videojuego es razonable.					
El tiempo de carga de los niveles del videojuego es razonable.					

Se ofrecen mecanismos de ayuda.					
Utiliza un lenguaje respetuoso y ameno para los usuarios.					
Influye en la formación de valores necesarios para nuestra realidad académica.					

Muchas gracias por su colaboración.