



Universidad de las Ciencias Informáticas

Facultad 4

Juegos: Recuento de figuras geométricas y colores, El elevador y Cálculo con figuras geométricas por tipos y colores para la colección MundoClick

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor:

Mary Carmen Chacón Castellanos

Tutores:

Ing. Yasirys Terry González

Ing. Arisbel Hechavarría Rojas

La Habana, Junio 2013

“Año del 55 Aniversario de la Revolución”

Declaración de autoría

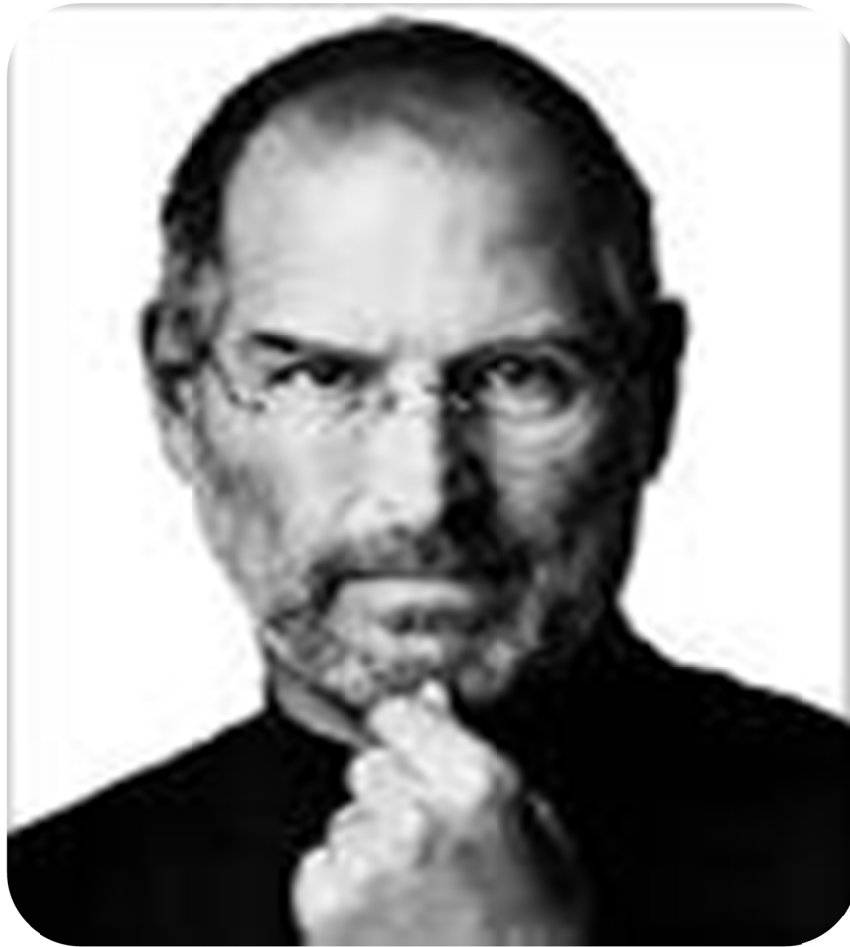
Declaro ser autora del presente trabajo de diploma y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmamos el presente a los días 26 del mes de Junio del año 2013.

Firma de la diplomante
Mary Carmen Chacón Castellanos

Firma del tutor
Ing. Yasirys Terry González

Firma del tutor
Ing. Arisbel Hechavarría Rojas



“El único modo de hacer un gran trabajo es amar lo que haces. Si no lo has encontrado todavía, sigue buscando. No te acomodes. Como con todo lo que es propio del corazón, lo sabrás cuando lo encuentres”

Steve Jobs

Dedico este trabajo:

A mis padres por ser lo máspreciado de este mundo, por guiarme por el buen camino, por hacer de mí una persona de bien y por educarme de la mejor forma posible.

A todos aquellos que de una forma u otra han contribuido con el desarrollo de este trabajo.

Agradecimientos

A mis padres por apoyarme en todo momento, por haber hecho lo posible y hasta lo imposible para enseñarme las cosas de la vida y darme una buena educación.

A Teresa y Liuvia porque son personas muy especiales para mí, las cuales me han brindado todo el amor del mundo, todo su apoyo y confianza.

A toda mi familia y vecinos por estar pendiente de mi todo este tiempo, en especial a María, Beba y familia.

A Leonardo por estar al pendiente de todo el transcurso de la tesis, siempre estuvo a mi lado para cualquier cosa sin mirar las consecuencias, pasó malas noches junto a mí ayudándome, me ayudó tanto, que aún no se de que forma le puedo agradecer el gesto tan lindo que tuvo conmigo.

A mis tutores, en especial a Yasirys que estuvo día a día revisando los avances de la tesis, siempre estuvo a disposición para lo que me hiciera falta, muchas gracias por todo.

Las personas sin las cuales no hubiera podido realizar esta tesis, Wilfredo, Yailín, Mileysis, Yoandys, gracias por su ayuda, entrega y por el tiempo dedicado.

A todos aquellos que me tendieron la mano y me ayudaron a seguir adelante desde que comencé la carrera hasta el día de hoy Yusdel, Yuneiry, Antonio Jesús, Antonio (Tony) y Yunior.

A mis compañeros de universidad, los de mi grupo en primer año a pesar de que algunos no pudieron continuar, a los del grupo 4305 los cuales me brindaron mucho apoyo de igual forma que el 4401.

Las tías que me han atendido en los edificios en los cuales he vivido y las que me han brindado su mano cuando las he necesitado, en especial Luzvila que ha sido como mi madre aquí en la universidad.

Resumen

El juego como actividad lúdica es tan antiguo como la propia existencia del hombre. Su utilización en el proceso de enseñanza-aprendizaje, ha permitido que se aprovechen sus características en pos de la formación de habilidades en el individuo de una forma motivadora e interesante. El surgimiento y desarrollo de las TIC propició que fuese posible recrear los juegos tradicionales en productos de software, los cuales lograron con rapidez insertarse dentro de la Educación. La construcción de colecciones de juegos educativos que facilitan la consolidación del conocimiento adquirido en determinadas asignaturas, permiten afianzar determinadas habilidades que por lo general son utilizadas dentro del contexto de la enseñanza-aprendizaje. El proyecto Multisaber-El navegante se propone como objetivo la construcción de una colección de juegos educativos, encaminados a niños en edades escolares entre 6 y 8 años. Para la construcción de dicha colección fue necesario determinar qué características se deben incorporar a los juegos, teniendo presente las peculiaridades de la audiencia y el contenido hacia el que esté encaminado. Se realiza un estudio del estado del arte y se utiliza el conjunto de herramientas y tecnologías definidas por el equipo de proyecto para desarrollar la propuesta de solución, se utiliza además como guía para el proceso de desarrollo de la metodología RUP. En la etapa de pruebas correspondiente a RUP, se verifica que el producto construido esté acorde a los requisitos especificados.

Palabras claves: enseñanza-aprendizaje, educación, juego, software educativo.

Introducción	VIII
Capítulo I. Fundamentación teórica	1
1.1 El juego en el proceso de enseñanza-aprendizaje	1
1.2 Software educativo, definiciones y características	3
1.3 Los juegos como software educativo	6
1.4 Habilidades matemáticas	8
1.5 Características iniciales de la colección MundoClick	10
1.6 Análisis de soluciones similares	10
1.7 Herramientas y tecnologías a utilizar	14
Conclusiones del capítulo	23
Capítulo II. Características del sistema	24
Introducción	24
2.1 Modelo conceptual	24
2.2 Descripción de la solución	25
2.3 Requisitos del sistema	27
2.4 Modelo de casos de uso	32
Conclusiones del capítulo	42
Capítulo III. Diseño del sistema	43
Introducción	43
3.1 Patrón arquitectónico Modelo-Vista-Controlador	43
3.2 Diagrama de clases del diseño	46
3.3 Modelo de diseño	44
3.4 Diagrama de interacción	48

3.5 Estándar de codificación	50
Conclusiones del capítulo	51
Capítulo IV. Implementación y prueba	52
Introducción.....	52
4.1 Modelo de implementación.....	52
4.2 Pruebas	55
Conclusiones del capítulo	58
Conclusiones	59
Recomendación	60
Referencias bibliográficas.....	61
Glosario de términos.....	69
Anexo 1: Guión del juego El elevador.....	70
Anexo 2: Guión del juego Cálculo con figuras geométricas por tipos y colores	74
Anexo 3: Guión del juego Recuento de figuras geométricas y colores	76

Introducción

El avance científico-técnico producido en los últimos 60 años, resumido en la revolución tecnológica que ha experimentado el mundo, ha impactado en muchas esferas de la formación y la vida social. Uno de los aspectos más relevantes de este avance está asociado al surgimiento de las denominadas Tecnologías de la Información y las Comunicaciones (de aquí en lo adelante TIC), que tuvo como impulso fundamental el surgimiento de la computadora.

La revolución tecnológica influyó en todos los campos de la ciencia, esto permitió desarrollar nuevas investigaciones, al mismo tiempo que fortaleció y transformó determinados procesos a fin de buscar nuevas formas de ejecución de los mismos. A este grupo de transformaciones no escapó la educación, ciencia que asumió las mejoras propuestas de las TIC para enriquecer el proceso de enseñanza-aprendizaje.

El uso de las TIC en la educación ha servido para hacer evidentes muchas de sus teorías. El diseño de materiales didácticos teniendo como soporte la tecnología ha contribuido a desarrollar nuevas metodologías de trabajo y también para recuperar viejas propuestas que en su momento no encontraron los medios o el contexto social propicio en el cual desarrollarse (1). Según Salvat, entre los diferentes enfoques teóricos sobre el proceso de enseñanza-aprendizaje, las aportaciones constructivistas han sido las más ricas en investigaciones, mostrando las TIC como herramientas mediadoras del aprendizaje y no únicamente como la fuente del mismo (1).

Entre estas herramientas mediadoras del aprendizaje se incorporó rápidamente el videojuego, el cual cambiaba de contexto con el simple hecho de ser utilizado con un propósito esencialmente educativo, esto propició denominar una nueva tipología de videojuegos: el videojuego (de aquí en adelante juego) educativo (2).

Todos estos avances e incorporaciones tecnológicas a la educación permitieron el auge del aprendizaje basado en computadoras, sobre todo aquel clasificado como aprendizaje basado en juegos, una de las corrientes pedagógicas más aplicadas en la última década (3).

En Cuba el aprendizaje basado en juegos ha permitido incorporar a las distintas enseñanzas (Primaria, Secundaria, Media Superior y Superior) nuevas formas de hacer llegar el conocimiento y las habilidades

en sus distintas variantes a los educandos (4). Consecuente de esta práctica, se ha desarrollado una incipiente industria nacional de software educativo para asistir el aprendizaje, siendo pionera de esta industria el equipo de desarrollo del Instituto Superior Pedagógico Enrique José Varona con sus aportes durante el desarrollo de las colecciones A jugar, Multisaber, El Navegante y Futuro (5).

La Universidad de las Ciencias Informáticas (UCI), fundada en el año 2002, se ha consolidado en la producción de software educativo. Sus primeros trabajos en el área estuvieron encaminados a migrar las colecciones antes mencionadas a entornos web, de conjunto con las instituciones del Ministerio de Educación (MINED).

El proyecto Multisaber–El Navegante¹, perteneciente al Centro de Tecnologías para la Formación (FORTES), a partir de las experiencias acumuladas y las expectativas comerciales existentes en el sector, se ha dado la tarea de construir una colección llamada MundoClick². Esta colección debe contener juegos orientados a la formación de habilidades, en niños cuyas edades estén comprendidas entre 6 y 8 años; entre las cuales se encuentran lógicas asociadas al desarrollo del cálculo matemático. El grupo de desarrollo de este proyecto cuenta además con las siguientes dificultades:

- Inexperiencia en el desarrollo de juegos educativos encaminados a formar habilidades de cálculo y agilidad mental.
- El código implementado en las colecciones anteriormente mencionadas sobre las cuales ya este proyecto trabajó, no puede ser utilizado porque no está acorde con la arquitectura definida para la plataforma de desarrollo Pointer³.

Esto trae como consecuencia que se plantee el siguiente **problema a resolver**: ¿Cómo contribuir al desarrollo de habilidades de cálculo y agilidad mental desde la colección MundoClick?

El problema anterior tiene concebido como **objeto de estudio**: Juegos educativos para niños.

¹ Multisaber–El Navegante proyecto que desarrolla las colecciones de software educativo, perteneciente al Centro FORTES adscrito a la Facultad 4. Universidad de las Ciencias Informáticas, La Habana.

² MundoClick es una colección de juegos educativos para niños.

³ Pointer es una plataforma de desarrollo orientada a la implementación y despliegue de juegos.

Esto permite que se plantee como **objetivo general**: Desarrollar los juegos: Recuento de figuras geométricas y colores, El elevador y Cálculo con figuras geométricas por tipos y colores para contribuir al desarrollo de habilidades de cálculo y agilidad mental para la colección MundoClick y se enmarca en el siguiente **campo de acción**: Juegos educativos que crean las habilidades de cálculo y agilidad mental en los niños.

Idea a defender

Si se desarrollan los juegos: Recuento de figuras geométricas y colores, El elevador y Cálculo con figuras geométricas por tipos y colores para MundoClick entonces se contará con una herramienta que permita contribuir a la formación de habilidades para realizar el cálculo mental en la enseñanza primaria.

Objetivos específicos:

- Elaborar el marco teórico de la investigación.
- Realizar el diseño de la propuesta de solución obtenida a partir del estudio de las soluciones similares.
- Implementar la propuesta de solución obtenida.
- Validar que el producto de software obtenido esté acorde con las funcionalidades realizadas.

Permitiéndose dar cumplimiento a los mismos a través de las siguientes **tareas de investigación**:

- Valoración crítica de los conceptos, tecnologías, metodologías y herramientas relacionadas con el objeto de estudio de la investigación.
- Levantamiento de los requisitos funcionales y no funcionales que formarán parte de la propuesta de solución asociada a la presente investigación.
- Modelado de los juegos: Recuento de figuras geométricas y colores, El elevador y Cálculo con figuras geométricas por tipos y colores acorde a la arquitectura de la plataforma Pointer.
- Implementación de los juegos: Recuento de figuras geométricas y colores, El elevador y Cálculo con figuras geométricas por tipos y colores acorde a la arquitectura de la plataforma Pointer.
- Integración de los juegos: Recuento de figuras geométricas y colores, El elevador y Cálculo con figuras geométricas por tipos y colores acorde a la arquitectura de la plataforma Pointer.

- Realización de pruebas funcionales a los juegos: Recuento de figuras geométricas y colores, El elevador y Cálculo con figuras geométricas por tipos y colores acorde a la arquitectura de la plataforma Pointer.

Durante la investigación se utilizaron métodos de la investigación científica entre los que se pueden nombrar métodos teóricos y empíricos.

Entre los **teóricos** se utilizaron:

Analítico-Sintético: Se utilizó durante la revisión de la documentación científico-técnica disponible asociada al objeto de la investigación, con el fin de realizar un análisis coherente de la problemática a resolver, y con el objetivo de sintetizar ideas que permitan fundamentar la propuesta de solución.

Histórico-Lógico: Se utilizó durante el análisis de la evolución de soluciones similares, sobre todo en la evolución de los conceptos asociadas a la informática educativa, específicamente en el campo de los juegos educativos.

Entre los **empíricos** se utilizaron:

Entrevista: Se utilizó para la obtención de información que permita ampliar los referentes teóricos sobre los que se fundamenta la propuesta de solución. Fue posible a través de ella obtener características y principios de diseño sobre las cuales elaborar una solución asociada al uso de los juegos educativos en la formación de habilidades en la enseñanza primaria.

La investigación en su desarrollo se ha dividido en cuatro capítulos:

Capítulo 1: Fundamentación teórica. Se realiza un análisis de los conceptos relacionados con el objeto de estudio, asumiendo los referentes teóricos fundamentales que permiten realizar una propuesta de solución.

Capítulo 2: Características del sistema. Se describen las características del sistema, se identifican los requisitos funcionales y los no funcionales, se elabora el modelo de casos de uso y las descripciones de cada uno de ellos.

Capítulo 3: Diseño del sistema. Se describe la solución que se propone, a partir de los diferentes modelos y diagramas correspondientes al diseño de un software.

Capítulo 4: Implementación y prueba. Se describe el sistema en función de componentes de software, se selecciona el método y la técnica de prueba que se va a utilizar, además se describen los casos de pruebas por los cuales se verificará el cumplimiento de los requerimientos especificados para el producto.

Capítulo I. Fundamentación teórica

1.1 El juego en el proceso de enseñanza-aprendizaje

El juego como actividad lúdica de ocio es tan antiguo como la propia existencia del hombre, e igualmente ha ido evolucionando de conjunto con la sociedad. Intrínsecamente el juego posee un carácter formador de la conducta social, permite no solamente que se establezcan rivalidades sino que estas rivalidades conduzcan a la formación de habilidades tanto físicas como mentales. Jugando el niño desarrolla su imaginación, el razonamiento, la observación, la asociación y comparación, su capacidad de comprensión y expresión, contribuyendo así a su formación integral (6) (7).

Varios autores han definido el término juego, Nielsen *et al.* lo definen como “un sistema de reglas, con unos objetivos marcados, en el que las acciones del jugador son evaluadas y, sobre todo, con una experiencia de juego diseñada” (8). Jiménez Rodríguez lo define como el “modo de interactuar con la realidad, propio de la infancia, que se caracteriza por su universalidad, regularidad y consistencia, siendo al mismo tiempo huella de la herencia biológica del hombre y producto de su capacidad creadora de cultura. Todo juego se desarrolla dentro de un marco psicológico que le da sentido y evoluciona con la edad reflejando en cada momento el modo en que el niño concibe el mundo y las relaciones que tiene con él” (9), Huizinga lo define como “una acción libre ejecutada, que puede absorber por completo al jugador, sin que haya en ella ningún interés material ni se obtenga en ella provecho alguno, que se ejecuta dentro de un determinado tiempo y un determinado espacio, que se desarrolla en un orden sometido a reglas y que da origen a asociaciones que propenden a rodearse de misterio o a disfrazarse para destacarse del mundo habitual” (10).

De acuerdo con las definiciones anteriores la autora de la presente investigación considera que la definición dada por Nielsen *et al.* posee puntos coincidentes con los demás conceptos asociados al término juego que fueron analizados, de igual forma contemplan enfoques sociales que permiten complementar y aportan útiles elementos al desarrollo de la investigación entre los que se pueden encontrar que los mismos se ejecutan en un tiempo determinado y tienen reglas que deben ser cumplidas.

Tal y como se expone en los conceptos anteriores el juego como actividad lúdica posee determinadas características, entre ellas se pueden nombrar:

- Es libre, espontáneo, no condicionado por refuerzos o acontecimientos externos.
- Requiere de determinadas actitudes y escenarios.
- Es un modo de expresión. Permite al jugador mostrar con mayor claridad sus motivaciones (11).
- Tiene reglas que los jugadores tienen que cumplir.
- Organiza las acciones de un modo específico (12).

Los juegos deben poseer determinados requisitos para que sean considerados como actividad para la enseñanza, generadora de aprendizaje, estos son (13):

- Requiere la participación activa de los jugadores.
- Tiene una finalidad que los usuarios entienden y pueden tratar de conseguir.
- Estimula el razonamiento de los niños (Se juega mejor si se piensa cómo hacerlo).
- Fomenta la comunicación entre iguales.
- Es susceptible de progresar en él, de jugar cada vez mejor.
- Puede ser jugado de manera autónoma.
- Su práctica satisface a los jugadores (los niños se divierten jugando).

Si se hace posible que un juego cuente con la mayoría de las características antes planteadas, al aplicarlo en el contexto educativo se fomenta la creación de valores y habilidades en los niños; razones suficientes para que en la enseñanza primaria en Cuba la actividad lúdica didácticamente bien diseñada ocupe un lugar dentro del proceso de formación.

La actividad lúdica objetiva y didácticamente diseñada permite crear climas donde el niño aprende y puede poner en práctica el resultado de su aprendizaje (14). Si se aprovechan todas las características que posee el juego y se lleva al contexto de la enseñanza asistida por computadora a través del uso de las TIC, entonces se puede decir que se está en presencia de software que sirve de apoyo al proceso de enseñanza–aprendizaje con un fin evidentemente educativo.

1.2 Software educativo, definiciones y características

El software educativo según Pere Marqués es “programas para ordenador creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y de aprendizaje” (15). Por otra parte, Galvis Panqueva denomina al software educativo como “programas que permiten cumplir o apoyar funciones educativas” (16). El proyecto Huascarán del Ministerio de Educación de Perú lo define como "cualquier tipo de programa o recurso informático que interviene en el proceso educativo produciendo: intervención activa en el proceso de enseñanza–aprendizaje, materiales educativos e intervención positiva en la gestión del proceso educativo" (17).

Begoña Gross define el término software educativo como "programas educativos pensados para ser utilizados en un proceso formal de aprendizaje y por ese motivo se establece un diseño específico a través del cual se adquieran unos conocimientos, unas habilidades, unos procedimientos, en definitiva, para que un estudiante aprenda" (18).

Luego de analizar detenidamente los conceptos anteriores, teniendo en cuenta la naturaleza de la presente investigación, se tomó como guía para el desarrollo de la misma, la definición de software educativo siguiente:

Un software educativo es una aplicación o programa informático diseñado con el objetivo principal de apoyar el proceso de enseñanza–aprendizaje.

Características del software educativo

- Permite la interactividad con los estudiantes, retroalimentándolos y evaluando lo aprendido.
- Incide en el desarrollo de las habilidades a través de la ejercitación.
- Permite simular procesos complejos.
- Facilita el trabajo independiente y a la vez un tratamiento individual de las diferencias (19).
- Son interactivos.
- Contestan inmediatamente las acciones de los estudiantes y permiten un diálogo y un intercambio de informaciones entre el ordenador y estos.
- Son fáciles de utilizar (20).

- Toma en cuenta las características de la población destinataria.
- Se adecua a los niveles de dominio diferenciado de los usuarios (21).

Tipos de software educativo

Algorítmicos: son aquellos donde la mayor parte del aprendizaje que obtiene el estudiante se realiza con la transmisión del conocimiento, pues el rol del alumno es asimilar el máximo de lo que se le transmite.

Considerando la función educativa se pueden clasificar en (22):

- **Sistemas Tutoriales:** Sistema basado en el diálogo con el estudiante, adecuado para presentar información objetiva, tiene en cuenta las características del alumno, siguiendo una estrategia pedagógica para la transmisión de conocimientos.
- **Sistemas Entrenadores:** Se parte de que los estudiantes cuentan con los conceptos y destrezas que van a practicar, por lo que su propósito es contribuir al desarrollo de una determinada habilidad, intelectual, manual o motora, profundizando en las dos fases finales del aprendizaje: aplicación y retroalimentación.
- **Libros Electrónicos:** Su objetivo es presentar información al estudiante a partir del uso de texto, gráficos, animaciones, videos, pero con un nivel de interactividad y motivación que le facilite las acciones que realiza.

Heurísticos: donde el estudiante descubre el conocimiento interactuando con el ambiente de aprendizaje que le permita llegar a él. Considerando la función educativa se pueden clasificar en (22):

- **Simuladores:** Su objetivo es apoyar el proceso de enseñanza-aprendizaje, semejando la realidad de forma entretenida.
- **Juegos Educativos:** Su objetivo es llegar a situaciones entretenidas, sin dejar en ocasiones de simular la realidad.
- **Sistemas Expertos:** Programa de conocimiento intensivo que resuelve problemas que normalmente requieren de la pericia humana. Ejecuta muchas funciones secundarias de manera análoga a un experto, por ejemplo, preguntar aspectos importantes y explicar razonamientos.

Funciones del software educativo

El software educativo, tal y como se enunció anteriormente, tiene un conjunto de funciones que son de obligatoria consulta cuando se construye este tipo de producto. Estas funciones no son excluyentes, lo que posibilita que se puedan combinar con un propósito específico. A continuación se exponen algunas de esas funciones (23):

- **Función informativa:** Presentan contenidos que proporcionan una información estructuradora de la realidad a los estudiantes. Como todos los medios didácticos, estos materiales representan la realidad y la ordenan como es el ejemplo de los tutoriales, simuladores y las bases de datos.
- **Función instructiva:** Todos los programas educativos orientan y regulan el aprendizaje de los estudiantes ya que, explícita o implícitamente, promueven determinadas actuaciones de los mismos encaminadas a facilitar el logro de unos objetivos educativos específicos. Los programas tutoriales son los que realizan de manera más explícita esta función instructiva, porque dirigen las actividades de los estudiantes en función de sus respuestas y progresos.
- **Función motivadora:** Generalmente los estudiantes se sienten interesados por todo el software educativo, ya que los programas suelen incluir elementos para captar la atención de los alumnos, mantener su interés y cuando sea necesario focalizarlo hacia los aspectos más importantes de las actividades.
- **Función evaluadora:** La interactividad propia de estos materiales, les permite responder inmediatamente a las respuestas y acciones de los estudiantes, los hace especialmente adecuados para evaluar el trabajo que se va realizando con ellos.
- **Función investigadora:** Los programas no directivos, especialmente las bases de datos, simuladores y programas constructores, ofrecen a los estudiantes interesantes entornos donde investigar: buscar determinadas informaciones, cambiar los valores de las variables de un sistema, entre otros.
- **Función lúdica:** Trabajar con los ordenadores en la realización de actividades educativas es una labor que a menudo tiene unas connotaciones lúdicas y festivas para los estudiantes. Además, algunos programas refuerzan su atractivo mediante la inclusión de determinados elementos lúdicos, con lo que potencian aún más esta función.

- **Función innovadora:** Aunque no siempre sus planteamientos pedagógicos resulten innovadores, los programas educativos se pueden considerar materiales didácticos con esta función ya que utilizan una tecnología incorporada a los centros educativos y, en general, suelen permitir muy diversas formas de uso.

Según todo lo expuesto en este y en los epígrafes anteriores existe una forma muy peculiar de realizar y consolidar el proceso enseñanza–aprendizaje usando programas computacionales detalladamente pensados. También es posible dentro de estos programas incorporar elementos lúdicos que conlleven a una forma distinta del juego, es decir, llevarlo de la forma tradicional a la forma virtual en un software, lo que no significa que la forma tradicional desaparezca, sino que surja un nuevo modo de jugar que permita complementar la ya existente, y aprovechar sus potencialidades por parte de docentes y educandos. De acuerdo con esto, es posible decir que los juegos, didácticamente diseñados, con una estructura acorde con la audiencia y un correcto guión, constituyen una importante herramienta mediadora del aprendizaje.

1.3 Los juegos como software educativo

En este epígrafe el término juego educativo que se utiliza hace referencia a la palabra videojuego educativo que es la utilizada en la bibliografía analizada. El término juego educativo se define como: un juego electrónico que cuenta como parte sustancial de su desarrollo con un dispositivo audiovisual, y donde generalmente un dispositivo físico de interfaz (teclado, mouse, gamepad) transfiere las acciones del jugador a la pantalla y las transforma o traduce en alguna acción (24).

Según Ana Sedeño, destacada investigadora en este campo, existen varios tipos de juegos según el uso educativo de cada uno, entre los que se pueden citar (25):

- **Acción:** Proponen actividades para causar una respuesta precisa, determinada y rápida al jugador. Son juegos en los que no hay que planificar ninguna acción, sino interactuar con el entorno lo más rápidamente posible mediante ejecuciones simples.
- **Estrategia:** En ellos se hace hincapié en la necesidad de planificar y establecer estrategias para poder avanzar en el juego, con lo que se desarrollan especialmente el pensamiento lógico y la resolución de problemas.

- **Aventura:** Incorporan una alta interactividad, la necesidad de tomar decisiones de forma constante y escenarios complejos.
- **Rol:** Desarrollan el cálculo mental, el vocabulario y estimulan la creatividad, además de ciertas actitudes o valores de socialización como la empatía, la tolerancia, la conciencia y la responsabilidad, unidos al trabajo en equipo.

La autora considera, que se ajusta especialmente la clasificación de juegos de rol, debido a las características antes expuestas.

Para la presente investigación, se hace referencia fundamentalmente al término habilidad matemática que se explica con mayor claridad en el epígrafe siguiente, varios autores opinan que estas son reconocidas como aquellas que se forman durante la ejecución de las acciones y operaciones que tienen un carácter esencialmente matemático. El desarrollo de habilidades de reconocimiento, cálculo lógico o comparación se encuentran entre las principales que se deben formar en los niños en edades tempranas (de 5 a 12 años), correspondientes a la enseñanza primaria, sobre todo en el primer ciclo escolar debido a que estas habilidades constituyen la base del aprendizaje (26).

Autores como Félix Etxeberría afirman que existen determinadas características que hacen de los juegos educativos un medio de aprendizaje más atractivo y efectivo debido a las siguientes características (27):

- Facilitan el acceso a «otros mundos» y el intercambio de unos a otros a través de los gráficos, contrastando de manera evidente con las aulas convencionales y estáticas.
- Favorecen la repetición instantánea y el intentarlo otra vez, en un ambiente sin peligro.
- Permiten el dominio de habilidades. Aunque sea difícil, los niños pueden repetir las acciones, hasta llegar a dominarlas, adquiriendo sensación de control.
- Facilitan la interacción con otros amigos, además, de una manera no jerárquica, al contrario de lo que ocurre en el aula.
- Hay claridad en los objetivos. Habitualmente, el niño no sabe qué es lo que está estudiando en matemáticas, ciencias sociales o alguna otra asignatura, pero cuando juega sabe que hay una tarea clara y concreta: abrir una puerta, rescatar a alguien, hallar un tesoro o cualquier otra acción, lo cual proporciona un alto nivel de motivación.

- Favorece un aumento de la atención y del autocontrol, apoyando la noción de que cambiando el entorno, no el niño, se puede favorecer al éxito individual.

Debido a las peculiaridades del juego, tal y como se expresaba anteriormente, es considerado como parte importante dentro del programa y orientaciones metodológicas para la enseñanza primaria (28), es por ello que no es difícil introducir de igual forma el juego educativo con la finalidad de obtener resultados semejantes.

En las investigaciones realizadas por el Dr. Kawashima asociadas a los juegos educativos y la actividad cerebral se hace especial énfasis en aquellos que propician el cálculo mental, reconocimiento o comparación y cálculo aritmético. Este tipo de juegos educativos según experimentos, propician un lento envejecimiento del cerebro además de que permite que sean obtenidas habilidades que aseguran la posibilidad de aumentar la agilidad con la que se realiza el cálculo matemático de forma mental (29).

Por esta razón, los juegos educativos que promueven la consolidación de estas habilidades, contribuyen decisivamente en el aprendizaje activo de los niños comprendidos en esta enseñanza. En especial para esta investigación importan aquellos capaces de lograr la consolidación de habilidades relacionadas con el cálculo y otros contenidos asociados con la Matemática.

Basados en la existencia de la formación en la enseñanza primaria en Cuba (28), surge la necesidad de que las características de los juegos educativos asociados a una misma disciplina varíen en complejidad según vaya pasando de nivel, esto hace comprensible que exista, cierto nivel de habilidad, conocimientos y aprendizaje.

1.4 Habilidades matemáticas

En la presente investigación se hace uso del término habilidad. Según el diccionario de la Real Academia Española (RAE) la palabra habilidad hace referencia a: "cada una de las cosas que una persona ejecuta con gracia y destreza" (30). La Dra. Aleida Márquez lo define como: "formaciones psicológicas mediante las cuales el sujeto manifiesta en forma concreta la dinámica de la actividad con el objetivo de elaborar, transformar, crear objetos, resolver situaciones y problemas, actuar sobre sí mismo" (31). Álvarez de Zayas manifiesta al respecto que: "La habilidad es el componente del contenido que refleja las realizaciones del hombre, en una rama del saber propia de la cultura. Es, desde el punto de vista

psicológico, el sistema de acciones y operaciones dominados por el sujeto y que responden a un objetivo" (32).

Desde el punto de vista de la autora de la presente investigación el concepto aportado por (32) es el más completo, debido a que permite que se distinga la estructura de una habilidad.

Es válido también resaltar que no es posible lograr o desarrollar una habilidad si no se tiene un conocimiento precedente o dominio de la actividad que ejecuta (33).

Existen varios tipos de habilidades asociadas a distintas ramas del saber, pero es interés abordar solo aquellas que influyen en el campo de acción correspondiente a la problemática a tratar, en este caso las habilidades de cálculo y agilidad mental, comprendidas dentro de las habilidades matemáticas.

En el caso de las habilidades antes mencionadas, es necesario señalar que dependen de la formación de habilidades básicas tales como: identificación, comparación y clasificación, explicadas con mayor detalle en (33). Es posible adquirir estos dos tipos de habilidades teniendo en cuenta fundamentalmente dos modos de actuación: fijación del modo de acción a través de la repetición y aplicación de las habilidades básicas a situaciones más complejas desde el punto de vista del contenido (34).

En el desarrollo de habilidades de cálculo y agilidad mental, la enseñanza de las matemáticas es fundamental, en variadas regiones del planeta (Europa, América del Norte, América del Sur), las acciones y operaciones dirigidas a tal fin en la enseñanza primaria, están enfocadas hacia tres ejes fundamentales: dominio de números y operaciones, espacio y forma (geometría) y tratamiento de la información (35).

De forma muy especial en la enseñanza de las matemáticas en Cuba, en el primer ciclo de la enseñanza primaria, se hace énfasis en el trabajo con los números y sus operaciones. Esto involucra procesos de contar, leer, escribir números, memorizar resultados y procedimientos operacionales. Otro apartado que recibe igual atención es el estudio de la geometría, donde se incluye la identificación y descripción de propiedades de figuras geométricas planas (36).

Analizando lo anteriormente expuesto y favoreciendo la actuación según los modos analizados, es posible deducir que su aplicación contribuye al desarrollo y perfeccionamiento de las habilidades que se analizan. Razón por la cual la autora de la presente investigación, teniendo en cuenta el estudio realizado en este epígrafe, decide involucrar todos estos elementos en la propuesta de solución con el objetivo de lograr y perfeccionar habilidades de cálculo y agilidad mental.

1.5 Características iniciales de la colección MundoClick

MundoClick es una colección de juegos educativos para niños. Cuenta con un total de 27 juegos que fomentan y ejercitan varias habilidades entre las que se encuentran las mencionadas en el epígrafe anterior, esta colección está basada en la plataforma de desarrollo Pointer que su objetivo principal es apoyar el desarrollo de los juegos creando una abstracción de varias funcionalidades comunes entre ellos; esto permitirá acortar el tiempo de desarrollo, estandarizar gran parte del código y ofrecer un ambiente controlado para el funcionamiento de los mismos.

Los juegos que se obtendrán en la propuesta de solución deben estar acordes con la arquitectura de la plataforma la cual está basada en el patrón Modelo Vista Controlador (MVC) y fue definida por el grupo de desarrollo de la misma para así poder lograr una mejor integración de los juegos con esta. (37)

El resultado de la presente investigación está dirigido a los estudiantes de la enseñanza Primaria. Esto conlleva a que el diseño y estructura de los juegos sea pensado desde el punto de vista pedagógico y didáctico, atendiendo a las necesidades de aprendizaje que poseen dichos estudiantes. De forma general la audiencia se caracteriza por (38):

- Estar en un rango de edades que comprende desde 5 hasta 9 años.
- Su nivel de habilidad se encuentra dado por el conocimiento que han adquirido en este período escolar.
- Los contenidos que más se han desarrollado están asociados a la Matemática y al desarrollo de habilidades comunicativas.
- Se encuentra en el período de formación donde se adquieren los conocimientos principales para el desenvolvimiento como estudiante.

1.6 Análisis de soluciones similares

El camino de Hexamano

El camino de Hexamano es una aplicación construida utilizando el software Macromedia Flash (herramienta privativa), donde antes de poder acompañar a Hexamano (personaje principal) en su

aventura espacial el jugador debe de probar su habilidad resolviendo 3 problemas o juegos matemáticos (39).

- **El primer juego:** Ventanas, tiene como objetivo que el jugador cierre la ventana que sobra cuando la suma de las 3 restantes se corresponde con un número que se muestra en pantalla. Es necesario tener más de 10 aciertos para que se tome el problema como resuelto y poder pasar a realizar el próximo.
- **El segundo juego:** Sumas, basa su lógica en seleccionar de 4 opciones posibles la que se ajusta al resultado de la suma que se muestra en la parte superior, en este caso hay que tener más de 12 respuestas correctas para poder pasar la prueba y tener derecho a la próxima.
- **El tercer juego:** Monedas, consiste en seleccionar la cantidad de monedas necesarias dada una distribución para poder comprar un artículo del cual se muestra el precio. El jugador debe tener al menos 9 aciertos para que se considere como pasada la prueba.

Una vez que el jugador pasa estas 3 pruebas tiene derecho a acompañar a Hexamano en su aventura que trata de un pequeño juego donde se deberán destruir naves enemigas, donde la velocidad de las naves aumenta con el transcurrir del tiempo. El objetivo es derribar la mayor cantidad de naves y así obtener mayor puntuación.

Gbrainy

Gbrainy es un juego de preguntas lógicas, para divertirse y mantener el cerebro bien entrenado. El juego permite seleccionar diferentes categorías en función de las preferencias de los jugadores. Se enfrenta a una mezcla de todos ellos o a una categoría en particular. Los juegos que presenta esta aplicación se dividen en cuatro categorías (40):

- **Juegos lógicos** - Juegos destinados a retar la capacidad del razonamiento.
- **Cálculo mental** - Juegos basados en operaciones aritméticas para probar la capacidad de cálculo mental.
- **Entrenadores de memoria** - Juegos diseñados para retar la memoria a corto plazo.
- **Analogías verbales** - Juegos que retan la aptitud verbal.

Algunas de las características básicas del juego es que permite seleccionar entre tres niveles de dificultad: fácil, medio y avanzado, porque es necesario incrementar la complejidad de los ejercicios cuando el entrenamiento va avanzando, además de configurar el tiempo empleado para memorizar o el número de partidas utilizadas (40).

La parte central de la aplicación muestra la pregunta y la imagen o imágenes asociadas, dependiendo de la categoría, y en la parte inferior aparece un cuadro de texto que permite se introduzca el valor considerado como respuesta, así como opciones para solicitar una pista adicional o para saltar a la pregunta siguiente. La interfaz es muy intuitiva y fácil, de forma que el usuario puede manejar perfectamente el juego con solo realizar unas pocas partidas (40).

Brain Juice

La colección Brain Juice cuyo proveedor es Digital Chocolate, cuenta con un total de 12 juegos que promueven la actividad cerebral y están fundamentalmente diseñados para su uso desde teléfonos celulares o tecnología móvil. Los juegos son fáciles de instalar, se encuentran en un paquete (.jar) y solo requieren de aplicaciones de java para que puedan ser instalados en teléfonos móviles.

Los juegos principales que posee la colección son (41):

- **Recuento de colores:** se produce una serie de preguntas asociadas a la pantalla que se mostrará, para que el usuario entre a través del teclado el resultado. Se combinan en las preguntas colores y figuras geométricas planas tales como: el círculo, triángulo y el cuadrado, además es necesario aplicar operaciones aritméticas como la suma para poder responder.
- **Aritmética (42):** el juego consiste en un grupo de ejercicios que individualmente combinan la suma, resta, multiplicación y división, tiene como objetivo que el usuario entre la mayor cantidad de respuestas en el menor tiempo posible, siempre que se completan 8 ejercicios sin cometer error el juego regala una bonificación de tiempo.
- **Números ascendentes:** su lógica está basada en que el usuario debe apretar los números en orden ascendente según los que se muestran en la pantalla, cada 4 ejercicios, igual que el anterior, ofrece una bonificación de tiempo.

- **Ascensor:** trata sobre observar la cantidad de personas y llevar el conteo de las que entran y salen del elevador, el ejercicio consiste que cada cierta cantidad de entradas y salidas en el ascensor se pregunte la cantidad de personas que se encuentran dentro, de forma que el usuario pueda llevar la cuenta correcta.

La lógica de la colección de juegos incita a jugar diariamente, pues inicialmente y a medida que alcancen mejores resultados se irán habilitando el resto de los juegos. Posee un sistema de gráficas que permiten observar si has tenido algún progreso al igual que puedes observar los mejores resultados obtenidos (41).

La colección permite que se activen sonidos durante el desarrollo de los juegos, los cuales se ejecutan contra un tiempo determinado, una vez terminado este se le ofrece el resultado al usuario y además se trata de incentivar a mejorar a través de la comparación con 5 tipos de animales (41):

- **Caracol:** cuando el jugador responde de forma muy lenta una pregunta.
- **Tortuga:** cuando el jugador responde de forma lenta una pregunta.
- **Liebre:** cuando el resultado se encuentra por encima del nivel bajo pero que aún no es bueno.
- **Galgo:** cuando la agilidad en dar las respuestas es relativamente bueno.
- **Guepardo:** es cuando la agilidad con que se responde es relativamente alta, equivalente a un excelente.

De forma general en esta colección se abordan elementos generales asociados a la aplicación de la matemática, el uso de las operaciones de cálculo fundamentales y de figuras geométricas planas. De forma muy amena constituye una forma de ejercitación y consolidación de habilidades asociadas al cálculo y la agilidad con que este se realiza (41).

Sintetizando el análisis realizado por la autora de la presente investigación de las colecciones de juegos antes analizados es posible concluir:

- Todos los juegos se realizan contra reloj.
- Los juegos utilizan diseños fáciles de lograr y no constituyen parte importante en la lógica del mismo, además de no estar orientado con las características de la enseñanza primaria.
- El uso de figuras geométricas conjugadas con números logra una forma especial de presentar el ejercicio, lo que permite que se aprendan también contenidos relacionados a la Geometría plana.

- En todos los casos los juegos poseen niveles de dificultad, en el caso de la colección Brain Juice, la complejidad de los juegos es sumamente elevada, lo que no permite que sea apto para todos los niños en la enseñanza primaria.
- Contribuyen al reconocimiento de algunos principios básicos que deben cumplir los juegos a desarrollar en la presente investigación, lo que permite identificar de forma general requisitos funcionales y no funcionales para conformar la propuesta de solución.
- Presentan problemas de compatibilidad con la arquitectura de la plataforma Pointer debido a que fueron desarrollados sobre otras herramientas por lo que no son aptos para integrarlos en ella.

1.7 Herramientas y tecnologías a utilizar

Las herramientas y tecnologías a utilizar para la plataforma de desarrollo Pointer fueron rigurosamente analizadas por el equipo de trabajo del proyecto Multisaber–El Navegante (37). La propuesta de solución formará parte de la plataforma antes mencionada, por lo que se decidió continuar con la utilización de las mismas herramientas para garantizar una uniformidad dentro del sistema.

Estas herramientas, marcos de trabajo y tecnologías, de forma general constituyen prácticas emergentes en la industria de software, probadas y que cuentan con el respaldo de grandes comunidades científicas y académicas.

Metodología de desarrollo

Una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar un software con calidad (43).

Define las etapas para el desarrollo de un software, las tareas que se deben llevar a cabo en cada etapa, las restricciones que deben aplicarse, las técnicas y herramientas que deben emplearse, y la forma de controlar y gestionar un proyecto (43).

RUP (*Rational Unified Process*)

RUP divide en cuatro fases el desarrollo del software, estas son: Inicio, la que define el modelo del negocio, el alcance y los límites del proyecto; Elaboración, en la cual se define, valida y cimenta la

arquitectura; Construcción, donde se desarrolla el producto y Transición, la encargada de poner el producto en manos de los usuarios (44).

Por cada una de estas fases se transita a través de nueve flujos de trabajo, para los que RUP propone una serie de artefactos, actividades y roles que varían de un flujo de trabajo a otro. Su ciclo de vida está caracterizado por ser (45):

- Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos.
- Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.
- Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque hace hincapié en algunos más que en otros de acuerdo a la fase.

Otras de las características que hacen idónea esta metodología son (43):

- Unifica los mejores elementos de metodologías anteriores.
- No necesariamente el cliente debe tomar un papel activo en el desarrollo del producto.
- Asociado al desarrollo se genera una gran cantidad de documentación que pueden ayudar a entender y mantener el sistema.

Lenguaje de Modelado

UML (*Unified Modeling Language*) versión 2.4.1

UML es un lenguaje de modelado que se ha convertido en un estándar para representar y modelar la información con la que se trabaja en las fases de análisis y especialmente en la de diseño. UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático. Como método formal para el modelado permite obtener ciertas ventajas, orientadas

a obtener un mayor rigor en la especificación, realizar la verificación y validación del modelo realizado y automatizar procesos en cuanto a la generación de código a partir de los modelos y viceversa (46).

Al igual que cualquier lenguaje, UML posee un vocabulario y reglas que hacen posible la comunicación de aspectos para definir y comprender la descripción gráfica de un sistema, razones que permiten asociar UML con cuatro objetivos fundamentales (46):

- Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Para poder cumplir los objetivos anteriormente planteados UML está compuesto por tres clases de bloques de construcción (46):

- Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos y acciones).
- Relaciones: relacionan los elementos entre sí.
- Diagramas: Son colecciones de elementos con sus relaciones.

UML comprende un interesante grupo de diagramas entre ellos en la presente investigación solo se utilizarán un subconjunto de los mismos, los cuales permiten tener una representación a mediana escala de las vistas estática y dinámica de un sistema de software determinado y que permiten que solo usando estos, se pueda conducir el ciclo de desarrollo de un producto particular: Diagrama de casos de uso, Diagrama de clases, Diagrama de colaboración y Diagrama de componentes (47).

Herramienta CASE

Las herramientas CASE (*Computer Aided Software Engineering* traducido al español Ingeniería de Software Asistida por Computación) permiten a los analistas tener más tiempo para el análisis y diseño y minimizar el tiempo para codificar y probar (48).

Estas herramientas pueden proveer muchos beneficios en todas las etapas del proceso de desarrollo de software, algunas de ellas son (48):

- Verificar el uso de todos los elementos en el sistema diseñado.
- Automatizar el dibujo de diagramas.
- Ayudar en la documentación del sistema.
- Generar estructuras de código.

Algunas de las ventajas de estas herramientas (48):

- Facilidad para la revisión de aplicaciones.
- Soporte para el desarrollo de prototipos de sistemas.
- Generación de código.
- Mejora en la habilidad para satisfacer los requerimientos del usuario.
- Soporte interactivo para el proceso de desarrollo.

Visual Paradigm for UML versión 8.0

Existe gran diversidad de herramientas para el modelado de software que usan UML como notación. Entre ellas existen muchas similitudes y a la vez diferencias, las cuales están basadas en la facilidad que estas proveen a los usuarios y equipos de desarrollo para el modelado de un software determinado y el entendimiento de los artefactos que a través de ellas se generan. El equipo de desarrollo de la plataforma Pointer, luego de un estudio de las herramientas CASE disponibles, decide utilizar Visual Paradigm 8.0 for UML. Esta herramienta tiene un gran número de prestaciones que soportan el ciclo de vida completo de un proyecto de desarrollo, es posible integrar con distintos entornos de desarrollos integrado (IDE: Netbeans, Eclipse). En el análisis se tuvieron en cuenta las siguientes características (49):

- Soporte del ciclo de desarrollo basado en RUP.
- Posibilidad de integración con IDE.
- La navegación intuitiva entre el código y el modelo visual.
- Poderoso generador de informes PDF/HTML.

- Un ambiente modelador visual superior.
- Sofisticado diseño de diagramas.

Lenguajes de desarrollo

Los lenguajes de desarrollo son aquellos elementos dentro de la informática que permiten crear programas mediante un conjunto de instrucciones, con una cierta base sintáctica y semántica, que se ponen a disposición del programador para que se pueda comunicar con los dispositivos de hardware y software existentes (50).

CSS3 (*Cascade Style Sheet*)

CSS3 (*Cascade Style Sheet*, traducido al español como: hojas de estilo en cascada) constituye la forma común en español, para nombrar una tecnología que permite definir el formato de presentación de los elementos HTML que coexisten en una página web. Esta tecnología ha evolucionado, en cada nueva especificación se ha tratado de ayudar a solventar dificultades que afrontaban los desarrolladores del software web.

La última especificación del CSS3 es que ofrece nuevas herramientas que permiten lograr una mejor presentación del código HTML y de elementos contenidos en documentos XML. CSS3 no solo contiene todo lo especificado en sus anteriores versiones, sino que incorpora nuevas características para ayudar a los desarrolladores en la solución de varios problemas sin la necesidad de usar marcado semántico, complejas secuencias de comandos o gran cantidad de imágenes para lograr determinados efectos de presentación.

CSS3 incluye nuevos selectores que permiten un mejor manejo de los distintos elementos HTML, características que permiten ofrecer nuevas formas para mostrar efectos tales como: sombras, esquinas redondeadas, fondos múltiples, animaciones, transparencia en los objetos e imágenes (51).

Durante el análisis arquitectónico de la plataforma, se pudo constatar que el uso de este lenguaje, posibilita crear mejores estilos de presentación, en menores tiempos de desarrollo, característica que permite abaratar costos de producción elevando el nivel de eficiencia, además de lograr mejor aspecto y calidad visual de las interfaces de usuario. Las características antes planteadas y el posible

aprovechamiento de las potencialidades que ofrece CSS3, hizo posible que confluyera entre el grupo de tecnologías a utilizar para el desarrollo de los juegos educativos que se pretenden con la realización de la presente investigación. Se tuvo en cuenta que asumir esta tecnología para determinados efectos de presentación determina que solo pueda ser utilizada para los navegadores Firefox, Opera, Safari y Google Chrome (52).

JavaScript versión 1.8

Fue diseñado para ser un lenguaje de elaboración de script que pudiera incrustarse en archivos HTML, creado para el desarrollo de aplicaciones cliente-servidor a través de Internet se utiliza principalmente para crear páginas web dinámicas, no es necesario compilar los programas para ejecutarlo, es multiplataforma, es muy utilizado por su fiabilidad y seguridad (53).

JavaScript es un lenguaje interpretado, usado para múltiples propósitos pero solo considerado como un complemento hasta ahora. Una de las innovaciones que ayudó a cambiar el modo en que se ve JavaScript fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento de código. La clave de los motores más exitosos fue transformar el código JavaScript en código máquina para lograr velocidades de ejecución similares a aquellas encontradas en aplicaciones de escritorio. Esta mejorada capacidad permitió superar viejas limitaciones de rendimiento y confirmar el lenguaje JavaScript como la mejor opción para la web (54).

Para aprovechar esta prometedora plataforma de trabajo ofrecida por los nuevos navegadores, JavaScript fue expandido en relación con portabilidad e integración. A la vez, interfaces de programación de aplicaciones (APIs) fueron incorporadas por defecto en cada navegador para asistir al lenguaje en funciones elementales. Estas nuevas APIs (como *web storage*, *canvas*, y otras) son interfaces para librerías incluidas en navegadores. La idea es hacer disponible poderosas funciones a través de técnicas de programación sencillas y estándares, expandiendo el alcance del lenguaje y facilitando la creación de programas útiles para la web (54).

JavaScript proporciona los medios para (55):

- Controlar las ventanas del navegador y el contenido que muestran.
- Programar páginas dinámicas simples sin tener que utilizar Java.

- Evitar depender del servidor Web para cálculos sencillos.
- Capturar los eventos generados por el usuario y responder a ellos.
- Simular el comportamiento de las macros CGI cuando no es posible usarlas.
- Comprobar los datos que el usuario introduce en un formulario antes de enviarlos.

HTML5 (*HyperText Markup Language*)

HTML5 (*HyperText Markup Language*, traducido al español como: lenguaje de marcado de hipertexto) provee básicamente tres características: estructura, estilo y funcionalidad. Nunca fue declarado oficialmente pero, incluso cuando algunas APIs (Interface de Programación de Aplicaciones) y la especificación de CSS3 por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, CSS y JavaScript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5 (56).

Permite definir la estructura de una página web a través de varios elementos como pueden ser títulos (de diferentes niveles), párrafos y listas (57). Es un estándar cada vez más soportado por los navegadores modernos Firefox, Opera, Safari y Google Chrome (58).

Algunas reglas establecidas para HTML5 (59):

- Las nuevas características se basan en HTML, CSS, DOM y JavaScript
- Reducir la necesidad de plugins externos (como Flash)
- Mejor manejo de errores
- Más marcado para reemplazar secuencias de comandos
- HTML5 es independiente del dispositivo
- El proceso de desarrollo debe ser visible para el público

JSON (*Java Script Object Notation*)

JSON (*JavaScript Object Notation*, traducido al español Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos, es un lenguaje independiente, es fácil de entender, utiliza la sintaxis de JavaScript para la descripción de objetos de datos, pero todavía Json es independiente de la plataforma y lenguaje (60).

JSON está constituido por dos estructuras (61):

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Framework de desarrollo

Un marco de trabajo o framework puede ser considerado de varias formas. Algunas definiciones de framework se muestran a continuación:

Conjunto de clases que cooperan para formar un diseño reutilizable en un dominio concreto de software (62).

Es un conjunto de clases e interfaces que cooperan para resolver un problema específico en el ámbito del software. Entre sus características podemos encontrar que está compuesto por un conjunto de clases o componentes, cada uno de ellos provee o representa una abstracción de un concepto en particular. Define cómo estas abstracciones trabajan juntas para resolver un problema. Todos los objetos de los frameworks son reutilizables (63).

Los frameworks simplifican el desarrollo de las aplicaciones mediante la automatización de las tareas comunes. Pueden incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Además de que proporcionan estructura al código fuente, forzando al programador a crear código más legible y más fácil de mantener.

jQuery versión 1.7

jQuery es un framework para el lenguaje JavaScript, que simplifica el tiempo y esfuerzo que se deriva de su uso. Cuando un desarrollador tiene que utilizar JavaScript, generalmente tiene que preocuparse por hacer scripts compatibles con varios navegadores y para ello tiene que incorporar mucho código que lo único que hace es detectar el navegador que utiliza el usuario. jQuery implementa una serie de clases (de

programación orientada a objetos) que permite programar sin que importe el navegador que utiliza el usuario, ya que funcionan de exacta forma en todas las plataformas (64).

Este framework, ofrece una infraestructura que ofrece facilidad para la creación de aplicaciones complejas del lado del cliente (en una arquitectura cliente - servidor). jQuery ofrece un excelente sistema de ayuda en la creación de interfaces de usuario y efectos dinámicos incluyendo aplicaciones que hacen uso de Ajax.

Node.js versión 0.8.11

Node.js es un entorno JavaScript, que utiliza un modelo asíncrono y dirigido por eventos, Node soporta protocolos TCP, DNS y HTTP (65). Una aplicación Node.js puede manejar miles de conexiones simultáneas con el mínimo uso de recursos, tiene la capacidad de compartir código entre el servidor y del lado del cliente (66).

IDE (*Integrated Development Environment*)

Un IDE (*Integrated Development Environment* traducido al español Ambiente Integrado de Desarrollo) es un entorno de programación, puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, puede utilizarse para varios. Por lo general un IDE está compuesto por un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (67).

Adicionalmente los IDE deben tener (68):

- Herramientas de automatización.
- Posibilidad de ofrecer un sistema de control de versiones.
- Factibilidad para ayudar en la construcción de interfaces gráficas de usuarios.

Netbeans versión 7.3

El Netbeans es uno de los IDE más utilizado en la actualidad. Es de código abierto, ha proporcionado a los desarrolladores mayor flexibilidad gracias a su integración con múltiples herramientas y lenguajes de programación (Java, C, C++, Python, PHP, JavaScript). Está disponible para Windows, Mac, Linux y Solaris. El proyecto de NetBeans está apoyado por una comunidad de desarrolladores y ofrece una amplia

documentación y recursos de capacitación. Teniendo como ventaja la capacidad de integrar los lenguajes de programación PHP y JavaScript (69).

Características del Netbeans (69):

- Apoyo a marcos de PHP múltiples.
- Está disponible en varios idiomas.
- Ofrece rendimiento significativo y la experiencia de codificación.

Conclusiones del capítulo

El estudio del arte permitió obtener los elementos teóricos necesarios para constatar que la utilización de los juegos educativos contribuye al desarrollo y consolidación de habilidades en el individuo en la enseñanza primaria.

A partir del análisis de soluciones similares, se considera que aunque ninguna de ellas podría integrarse a la Plataforma Pointer, sí permitieron la identificación de elementos que se deben tener en cuenta para la concepción de la propuesta de solución entre los que se encuentran: ser ejecutados contra reloj, poseer interfaces sencillas y ser fáciles de jugar.

Las herramientas definidas como parte de la arquitectura de la plataforma y caracterizadas en este capítulo, posibilitan concluir el ciclo de desarrollo del software y así obtener una correcta implementación de la propuesta de solución.

Capítulo II. Características del sistema

Introducción

En el presente capítulo se realiza la descripción de la propuesta de solución, donde se incluye el modelo conceptual, se identifican los requisitos funcionales y los no funcionales, se elabora el diagrama de casos de uso del sistema para reflejar las relaciones entre los actores del sistema con los casos de uso y se realiza la descripción de los casos de uso arquitectónicamente significativos.

2.1 Modelo conceptual

El modelo conceptual es una descripción del dominio de un problema, identificando los atributos y las asociaciones existentes entre ellos. A partir de este modelo el diseñador agrega detalles a la definición de las clases (70).

A continuación se muestra el modelo conceptual de los juegos **Recuento de figuras geométricas y colores**, **El elevador** y **Cálculo con figuras geométricas por tipos y colores**.

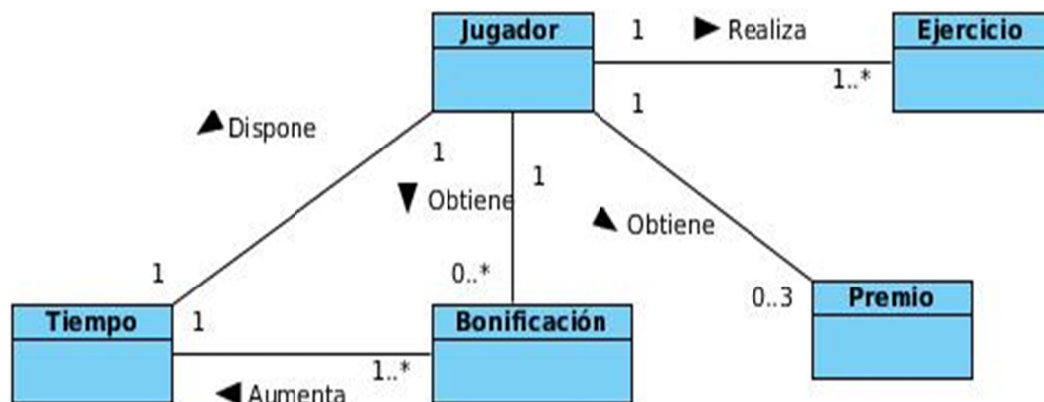


Figura 1. Modelo conceptual

Definición de los conceptos fundamentales:

Jugador: Persona que interactúa con la aplicación con el objetivo de aprender y vencer los contenidos educativos expuestos.

Ejercicio: Secuencia de imágenes, la cual está compuesta por el ejercicio y la pregunta a responder.

Tiempo: Delimita un intervalo de tiempo, durante el cual, el jugador puede responder las preguntas que muestre el sistema.

Bonificación: Se encarga de gestionar el tiempo extra, que se adiciona al tiempo total, la cantidad de tiempo que se adiciona por cada bonificación está en dependencia del nivel.

Premio: Gestiona los premios que se le entregan al jugador de acuerdo a la calidad de las respuestas del jugador, es decir en dependencia de las respuestas correctas, se otorgará un total de 3 premios, 1 por cada nivel del juego.

2.2 Descripción de la solución

El elevador

El juego contará con tres niveles. La dificultad de cada nivel estará vinculada con la cantidad de pisos que recorre el elevador, la cantidad de personas que suben y bajan del elevador, sumándole a esto la cantidad de ejercicios que se realizan por nivel.

Niveles del juego:

Nivel 1: Cuenta con 5 ejercicios, cada ejercicio con 3 pisos y solamente pueden entrar mujeres y hombres.

Nivel 2: Cuenta con 7 ejercicios, cada ejercicio con 4 pisos y solamente pueden entrar mujeres, hombres y niños.

Nivel 3: Cuenta con 9 ejercicios, cada ejercicio con 5 pisos y solamente pueden entrar mujeres, hombres y niños.

Un ejercicio del juego consiste en:

- Abrir las puertas del elevador.
- Subir n cantidad de hombres, mujeres y niños según este definido en para cada nivel.

- En todos los pisos se produce la entrada y salida de hombres, mujeres y niños excepto en el primer piso que solo se produce la entrada de personas.
- Al llegar al piso en el que concluye el ejercicio se preguntará al usuario ¿Cuántas personas (hombres, mujeres o niños) hay dentro del elevador?

El juego se complejiza cuando va pasando de nivel. Cuando se producen las respuestas correctas de forma consecutiva definidas para cada nivel se realizará una bonificación de tiempo. Al iniciar el juego el usuario dispone de 60 segundos para realizar las respuestas de los ejercicios propuestos para cada nivel, el cual es mostrado a través de una barra de tiempo. Se premia al jugador según la efectividad (efectividad = preguntas correctas/total de preguntas por cada nivel) obtenida en cada nivel. Al terminar el juego se muestran los resultados obtenidos por el jugador.

Recuento de figuras geométricas y colores y Cálculo con figuras geométricas por tipos y colores

Basan su lógica en el cálculo aritmético y el reconocimiento de figuras planas y colores. Para el desarrollo estos juegos contarán con 3 niveles, la dificultad de estos está dada por los factores anteriormente expuestos.

Un ejercicio del juego consiste en:

- Mostrar la pregunta del ejercicio.
- Mostrar la secuencia animada que contiene las distintas figuras geométricas, según la cual, se debe responder la pregunta realizada.
- Una vez entrada la respuesta el sistema verifica si es correcta o incorrecta.

La diferencia entre los juegos **Recuento de figuras geométricas y colores** y **Cálculo con figuras geométricas por tipos y colores** radica que en el primero se contarán las figuras geométricas atendiendo al tipo y color según se indica en la pregunta y el segundo juego realiza la operación aritmética suma entre los valores numéricos que contienen las figuras, el cálculo se complejiza al cambiar los valores numéricos, el color y tipo de la figura. Cuando se producen tres respuestas correctas de forma consecutiva se realizará una bonificación de tiempo. Al iniciar el juego el usuario dispone de 60 segundos para realizar los ejercicios propuestos para cada nivel, el cuál es mostrado a través de una barra de tiempo. Se premia al jugador según la efectividad (efectividad = preguntas correctas/total de preguntas por

cada nivel) obtenida en cada nivel. Al terminar el juego se muestran los resultados obtenidos por el jugador.

Con la descripción de solución expuesta y la ayuda de los guiones de los juegos ([Anexo1](#), [Anexo2](#), [Anexo3](#)) se pudo definir con mayor claridad los requisitos funcionales de estos.

2.3 Requisitos del sistema

El propósito fundamental del flujo de trabajo Requerimientos es guiar el desarrollo hacia la obtención del sistema correcto. Esto se consigue mediante una descripción de los requisitos del sistema suficientemente buena como para que pueda llegarse a un acuerdo entre el cliente y los desarrolladores sobre qué debe y qué no debe hacer el sistema (71). Los requisitos se clasifican en: funcionales y no funcionales. Los funcionales describen el funcionamiento del sistema. Son las condiciones y capacidades que necesita el usuario para resolver un problema o conseguir un objetivo determinado (71). Los no funcionales especifican propiedades o cualidades del sistema, como restricciones del entorno o de la implementación, facilidad de mantenimiento, fiabilidad, entre otras (72).

Requisitos funcionales del juego El elevador

RF1 Mostrar pregunta del ejercicio

Una vez que se han subido la cantidad de pisos definidas por cada nivel y hayan entrado y salido personas del elevador, se procederá a mostrar una pregunta donde se pedirá al jugador que entre el valor correspondiente a la solución de la pregunta que se realiza. La pregunta aumenta la complejidad a medida que el jugador aumenta pase de nivel.

RF2 Introducir respuesta del ejercicio

Se debe permitir que el jugador introduzca a través del teclado el valor de la respuesta de la pregunta realizada una vez que se han subido los pisos definidos en el nivel.

RF3 Verificar respuesta del ejercicio

Una vez introducida la respuesta de la pregunta, el sistema procederá a verificar si el valor entrado es correcto o no.

RF4 Mostrar resultado de la verificación de la respuesta del ejercicio

Cuando se verifique la respuesta introducida por el jugador se mostrará una imagen acorde con el resultado de la pregunta.

RF5 Subir piso del edificio

Cuando se cierran las puertas del elevador, luego de la entrada y salida de personas se procede a mostrar una secuencia de imágenes que indica que el elevador está subiendo 1 piso del edificio.

RF6 Realizar bonificación al tiempo

La realización de bonificaciones al tiempo del que dispone un jugador estará en correspondencia con la estrategia plasmada en el guión del juego ([Anexo 1](#)).

RF7 Disminuir tiempo

Al comenzar cada nivel se asigna al jugador un tiempo límite de 60 segundos. Una vez que comienza el juego el tiempo comienza a disminuir en el instante que se muestra al jugador la pregunta y hasta que este introduzca una respuesta. En el resto del procedimiento en el juego no se disminuye el tiempo.

RF8 Mostrar entrada de personas al elevador

El sistema deberá mostrar las personas entrando al elevador.

RF9 Mostrar salida de personas del elevador

El sistema deberá mostrar las personas saliendo del elevador.

RF10 Mostrar premios obtenidos

Siempre que se otorgue un premio a un jugador este se deberá mostrar en pantalla.

RF11 Mostrar nivel del juego

Se deberá mostrar en pantalla un bloque de información donde se exponga en todo momento el nivel donde se encuentra el jugador.

RF12 Mostrar resultados del juego

Siempre que se termine el juego o se agote el tiempo se deberán mostrar los resultados obtenidos por el jugador.

RF13 Otorgar premio

Los premios serán otorgados según la efectividad que obtiene el jugador en cada nivel.

RF14 Cambiar nivel

Cambia de nivel siempre que el jugador responda la cantidad de ejercicios definidos para cada nivel.

RF15 Terminar juego

Permite al jugador salir del juego.

RF 16 Múltiples jugadores en el mismo juego

Permite a varios usuarios jugar en el mismo juego.

Requisitos funcionales de los juegos Recuento de figuras geométricas y colores y Cálculo con figuras geométricas por tipos y colores.

RF1 Mostrar pregunta del ejercicio

Una vez comenzado el juego se debe mostrar una pregunta donde se pedirá al jugador que entre el valor correspondiente a la solución de la pregunta que se realiza. La pregunta aumenta la complejidad a medida que el jugador aumenta su nivel.

RF2 Introducir respuesta del ejercicio

Se debe permitir que el jugador introduzca a través del teclado el valor de la respuesta de la pregunta realizada una vez que se han mostrado las figuras del ejercicio.

RF3 Verificar respuesta del ejercicio

Una vez introducida la respuesta de la pregunta, el sistema procederá a verificar si el valor entrado es correcto o incorrecto.

RF4 Mostrar resultado de la verificación de la respuesta del ejercicio.

Cuando se verifique la respuesta introducida por el jugador se mostrará una imagen acorde con el resultado de la pregunta.

RF5 Realizar bonificación al tiempo

La realización de bonificaciones al tiempo del que dispone un jugador se realizará cada vez que el mismo responda tres preguntas de forma consecutiva cada una en un tiempo igual o menor a 6 segundos.

RF6 Disminuir tiempo

Al comenzar el juego se asigna al jugador un tiempo límite de 60 segundos. Una vez que comienza el juego, el tiempo comienza a disminuir en el instante que se le muestra al jugador la pregunta y hasta que este introduzca una respuesta. En el resto del procedimiento del juego no disminuye el tiempo.

RF7 Mostrar premios obtenidos

Siempre que se otorgue un premio a un jugador este se deberá mostrar en pantalla.

RF8 Mostrar nivel del juego

Se mostrará en pantalla un bloque de información donde se exponga en todo momento el nivel donde se encuentra el jugador.

RF9 Mostrar resultados del juego

Siempre que se termine el juego o se agote el tiempo se deberán mostrar los resultados obtenidos por el jugador.

RF10 Otorgar premio

Los premios serán otorgados según la efectividad que obtiene el jugador en cada nivel.

RF11 Cambiar nivel

Cambia de nivel siempre que el jugador responda la cantidad de ejercicios definidos para cada nivel.

RF12 Mostrar figuras del ejercicio

Una vez que se visualiza la pregunta se muestran las figuras correspondientes al ejercicio.

RF13 Terminar juego

Permite al jugador salir del juego.

RF14 Múltiples jugadores en el mismo juego

Permite a varios usuarios jugar en el mismo juego.

Requisitos no funcionales

RNF1 Requisito de Portabilidad

El sistema podrá ser utilizado bajo los sistemas operativos Windows XP o superior y Linux distribución Ubuntu 12.04 o superior.

RNF2 Requisitos de hardware

- Procesador Pentium 233 MHz (recomendado 500 MHz o mayor).
- 512 MB de RAM (recomendado 1 GB de RAM o mayor).
- Lector de CD-ROM.
- Soporte de video que admita resolución de al menos 1024x780px y 24 bits.
- Dispositivo de red de al menos 10 Mbits de velocidad de transmisión.

RNF3 Requisitos de interfaz

El sistema debe presentar una interfaz marcadamente infantil, deben primar las imágenes en lugar de los textos para transmitir mensajes.

RNF4 Requisito de soporte

Se realizará transferencia tecnológica del juego a los clientes.

RNF5 Requisito de usabilidad

El software será usado por los niños en las edades de 6 a 9 años.

RNF6 Restricciones de la implementación

- Lenguaje de desarrollo: JavaScript v1.8.
- Lenguaje de marcado: HTML 5.
- Hoja de estilo: CSS 3.
- Framework de desarrollo: jQuery v1.7 y Node.js v0.8.11.
- Entorno de desarrollo integrado: Netbeans v7.3.
- Herramienta CASE: Visual Paradigm 8.0 for UML.

2.4 Modelo de casos de uso

El modelo de casos de uso describe los requisitos funcionales de la aplicación. Los mismos se muestran en un diagrama de casos de uso (73).

Actores del sistema

Un actor puede ser un ser humano, o agente externo del sistema, que interactúa o participa en el caso de uso y se beneficia de esa interacción (74).

Tabla 1. Actores del sistema

Actores del sistema	Descripción
Jugador	Persona que interactúa con la aplicación con el objetivo de aprender y vencer los contenidos educativos expuestos.

Diagrama de casos de uso del sistema

El diagrama de casos de uso del sistema muestra gráficamente las relaciones existentes entre actores y casos de uso, estos constituyen una técnica de captura de requisitos (75).

A continuación se muestra el diagrama de casos de uso correspondiente a los juegos **Recuento de figuras geométricas y colores**, **El elevador** y **Cálculo con figuras geométricas por tipos y colores**.

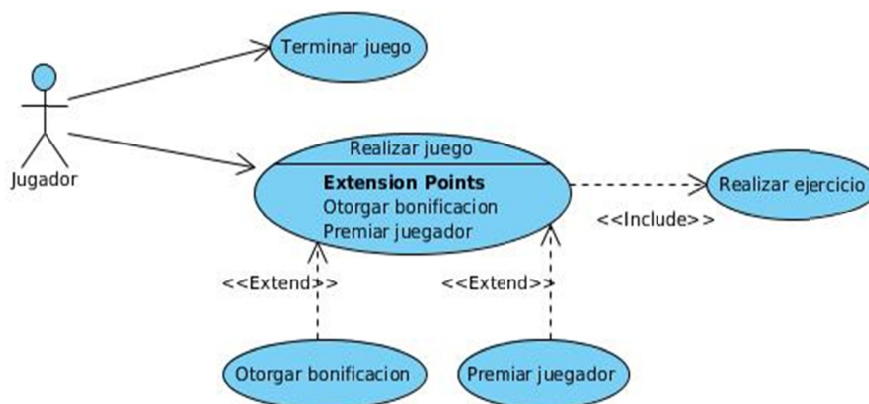


Figura 2. Diagrama de casos de uso

Descripciones de casos de uso

Las descripciones de casos de uso permiten comprenderlos mejor. A continuación se describen los casos de uso Realizar ejercicio, Otorgar bonificación, Realizar juego, Premiar jugador y Terminar juego siendo estos tres últimos la misma descripción para los tres juegos, estas servirán de guía para entender el funcionamiento de los mismos.

Tabla 2.Descripción del caso de uso: Realizar juego

Casos de uso	Realizar juego	
Actores	Jugador	
Resumen	El caso de uso inicia una vez que el jugador inicia el juego. Termina cuando el jugador sale del juego.	
Precondiciones	-	
Postcondiciones	-	
Prioridad	Crítico	
Flujo normal de eventos		
	Actor	Sistema
1.	Selecciona la opción Jugar.	1.1 Inicia la secuencia de preguntas que debe contestar el jugador: (Ver CU Realizar ejercicio) 1.2 Muestra al jugador el nivel en que se encuentra jugando. 1.3 Muestra al jugador el momento en que pasa de nivel. 1.4 Muestra cuando un jugador obtiene una bonificación: (Ver CU Realizar bonificación). 1.5 Muestra cuando un jugador obtiene un Premio: (Ver CU

		<p>Premiar jugador).</p> <p>1.6 Cuando se termina el tiempo del que dispone para realizar el ejercicio muestra los resultados generales del juego:</p> <ul style="list-style-type: none"> - Nivel - Preguntas correctas - Preguntas incorrectas - Bonificaciones obtenidas - Premios obtenidos <p>1.7 Permite salir del juego o volver a jugar.</p>
2.	Selecciona la opción Salir del juego.	<p>2.1 Almacena los datos del jugador una vez concluido el juego.</p> <p>2.2 Muestra la pantalla principal de la aplicación.</p> <p>2.3 Termina el caso de uso</p>
Flujos alternos		
1.7.a Volver a jugar		
	Actor	Sistema
1.	Selecciona la opción Volver a jugar.	1.1 Inicia nuevamente el flujo básico de eventos a partir del evento 2.
2.		2.1 Termina el caso de uso.
Relaciones	CU Incluidos	Realizar ejercicio.
	CU Extendidos	<p>Realizar bonificación Paso 1.4 del Flujo básico.</p> <p>Premiar jugador Paso 1.5 del Flujo básico.</p>
Requisitos no funcionales		

Asuntos pendientes	
---------------------------	--

Tabla 3. DCU: Premiar jugador

Caso de Uso	Premiar jugador	
Actores	Jugador	
Resumen	El caso de uso inicia cuando se calcula la efectividad del jugador para el nivel que se juega. Una vez calculada, se define en función del valor obtenido qué premio es el que se otorga al jugador en el nivel. Termina cuando se otorga al jugador el premio obtenido.	
Precondiciones	-	
Postcondiciones	Se ha otorgado el premio al jugador.	
Prioridad	Crítico	
Flujo normal de eventos		
	Acción Actor	Respuesta del Sistema
1.		<p>1.1 Otorga el premio de acuerdo a la efectividad con que haya respondido las preguntas del nivel. Los premios son:</p> <ul style="list-style-type: none"> - Reloj de oro: cuando la efectividad mayor que 0,8. - Reloj de plata: cuando la efectividad es mayor que 0,5 y menor o igual 0,8. - Reloj de bronce: cuando la efectividad es mayor o igual que 0,3 y menor o igual que 0,5. <p>1.2 La efectividad de las respuestas en un nivel se calcula:</p>

		Preguntas correctas/total de preguntas realizadas por el nivel. 1.3 Muestra los premios obtenidos en la parte izquierda inferior de la pantalla.
2.		2.1 Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	Premiar jugador en el CU Realizar juego.
Requisitos funcionales	no	
Asuntos pendientes		

Tabla 4. DCU: Terminar juego

Casos de uso	Terminar juego	
Actores	Jugador	
Resumen	El caso de uso inicia cuando el jugador selecciona la opción Salir del juego. Termina cuando se cierra el juego.	
Precondiciones	-	
Postcondiciones	Se termina el juego	
Prioridad	Crítico	
Flujo normal de eventos		
	Actor	Sistema
1.	Selecciona la opción Salir del juego.	1.1 Muestra un mensaje de confirmación de la acción "¿Está seguro que desea salir del juego?".

		1.2 Guarda los resultados del jugador.
2.		2.1 Termina el caso de uso.
Flujos alternos		
1.1.a Cancelar salir del juego		
	Actor	Sistema
1.	Selecciona la opción "No" del mensaje mostrado.	1.1 Continúa en el juego.
2.		2.2 Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos funcionales	no	
Asuntos pendientes		

Especificaciones de casos de uso. Juego: El Elevador

Tabla 5. DCU: Realizar ejercicio

Caso de Uso	Realizar ejercicio
Actores	Jugador
Resumen	El caso de uso inicia cuando el jugador selecciona la opción jugar. Termina cuando haya respondido los ejercicios que propone el sistema.
Precondiciones	-
Postcondiciones	Se muestra si el ejercicio ha sido respondido de forma correcta o incorrectamente.
Prioridad	Crítico

Flujo normal de eventos		
	Acción Actor	Respuesta del Sistema
1.	Selecciona la opción Jugar.	<p>1.1 Muestra la secuencia inicial de cada ejercicio que consiste en:</p> <ul style="list-style-type: none"> - Abrir las puertas del elevador. - Mostrar de forma automática la entrada de personas al elevador en cada uno de los pisos por los que transita. - Mostrar de forma automática la salida de personas del elevador en cada uno de los pisos por los que transita, con excepción del primero. <p>1.2 Muestra la pregunta a responder una vez que se ha subido los pisos que propone el nivel para cada ejercicio y el campo para la entrada de la respuesta.</p> <p>1.3 Comienza a disminuir el tiempo del que dispone el jugador.</p>
2.	Introduce la respuesta a la pregunta realizada en el campo mostrado.	<p>2.1 Verifica respuesta dada por el jugador.</p> <p>2.2 Muestra resultado de la verificación "Correcto" o "Incorrecto" acompañado de una imagen de igual significado.</p> <p>2.3 Detiene la disminución del tiempo.</p> <p>2.4 Muestra próximo ejercicio y se comenzará a ejecutar nuevamente el flujo básico de eventos a partir del paso 1.1.</p>

3		3.1 Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Tabla 6. DCU: Otorgar bonificación

Caso de Uso	Otorgar bonificación	
Actores	Jugador	
Resumen	El caso de uso inicia cuando el jugador ha ganado una cantidad de ejercicios de forma consecutiva definidos por el nivel y termina cuando se le adiciona ese tiempo al tiempo del juego.	
Precondiciones	Realizar una cantidad de ejercicios de forma consecutiva en un tiempo definido para cada nivel.	
Postcondiciones	Se adiciona la bonificación al tiempo del juego.	
Prioridad	Crítico	
Flujo normal de eventos		
	Acción Actor	Respuesta del Sistema
1.		<p>1.1 Adiciona al tiempo del juego la bonificación obtenida definida por el nivel en que se encuentre.</p> <p>Nivel 1: 9 segundos.</p> <p>Nivel 2: 6 segundos.</p> <p>Nivel 3: 4 segundos.</p> <p>1.2 Permite que el jugador continúe</p>

		la realización de ejercicios. 1.3 Aumenta la cantidad de bonificaciones obtenidas hasta el momento.
2		3.1 Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos funcionales	no	
Asuntos pendientes		

Especificaciones de casos de uso. Juego: Recuento de figuras geométricas y colores y Cálculo con figuras geométricas por tipos y colores.

Tabla 7. DCU: Realizar ejercicio

Caso de Uso	Realizar ejercicio	
Actores	Jugador	
Resumen	El caso de uso inicia cuando el jugador selecciona la opción jugar y termina cuando haya respondido el ejercicio que muestra el sistema.	
Precondiciones	-	
Postcondiciones	Se muestra si el ejercicio ha sido respondido correcto o incorrectamente.	
Prioridad	Crítico	
Flujo normal de eventos		
	Acción Actor	Respuesta del Sistema

1	Selecciona la opción Jugar.	<p>1.1 Muestra la pregunta a responder una vez comenzado el juego.</p> <p>1.2 En el instante que muestra la pregunta comienza a disminuir el tiempo del que dispone el jugador.</p> <p>1.3 Muestra las figuras que complementan la pregunta para ofrecer una respuesta.</p>
2	Introduce la respuesta a la pregunta realizada.	<p>2.1 Verifica respuesta dada por el jugador.</p> <p>2.2 Muestra resultado de la verificación incrementando el contador de la pregunta correcta o incorrecta acompañado de una imagen de igual significado.</p> <p>2.3 Detiene la disminución del tiempo hasta que sea mostrada la próxima pregunta.</p> <p>2.4 Muestra próximo ejercicio y se comenzará a ejecutar nuevamente el flujo básico de eventos a partir del paso 1.1.</p>
3		4.1 Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Tabla 8. DCU: Otorgar bonificación

Caso de Uso	Otorgar bonificación
Actores	Jugador

Resumen	El caso de uso inicia cuando el jugador ha ganado una cantidad de ejercicios de forma consecutiva. Termina cuando se le adiciona ese tiempo al tiempo real del juego.	
Precondiciones	Realizar de forma consecutiva 3 ejercicios en un tiempo menor o igual a 6 segundos.	
Postcondiciones	Se adiciona a bonificación al tiempo.	
Prioridad	Crítico	
Flujo normal de eventos		
	Acción Actor	Respuesta del Sistema
1.		1.1 Adiciona al tiempo real del juego la bonificación obtenida. 1.2 Permite que el jugador continúe la realización de ejercicios. 1.3 Aumenta la cantidad de bonificaciones obtenidas hasta el momento.
2		2.1 Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Conclusiones del capítulo

El análisis de los guiones de los juegos propuestos permitió obtener un modelo conceptual correcto, que ilustra el dominio de interés asociado a la problemática.

Con la realización de la descripción de los requisitos funcionales y de casos de uso se logra un mejor entendimiento de las funcionalidades definidas para la propuesta de solución realizada.

Capítulo III. Diseño del sistema

Introducción

En este capítulo se expone la concepción general referente al diseño del sistema propuesto. El propósito de esta fase es adquirir una comprensión de los requisitos funcionales y no funcionales y restricciones relacionadas con los lenguajes de programación, descomponer los trabajos de implementación en partes más manejables que puedan ser llevados a cabo por diferentes equipos de desarrollo. Se describe la solución que se propone a partir de los diagramas de clases del diseño, que reflejan una vista interna del sistema.

3.1 Patrón arquitectónico Modelo-Vista-Controlador

Reconocido como estilo por Taylor y Medvidovic (76), en ocasiones es tomado más bien como un patrón de diseño o como una práctica recurrente, mientras Buschman y otros autores lo consideran un patrón correspondiente al estilo de los sistemas interactivos (77). Este estilo es muy usado en las aplicaciones web aunque no es exclusivo de estas. El estilo define el modelo (modelo de datos), la vista (visualización de la información) y la lógica de control (lógica del negocio) en 3 componentes distintos, los cuales interactúan entre sí para dar una salida final al usuario, donde cada componente conoce a los demás solo lo necesario. Por lo que una solución basada en este estilo es muy flexible. Permite soporte a vistas múltiples dado que la misma se encuentra separada del modelo y no hay una dependencia directa del modelo respecto a la vista, por lo que es posible que se puedan mostrar distintas vistas de los mismos datos simultáneamente, lo que evidencia la fácil adaptación a los cambios.

La plataforma de desarrollo Pointer entre sus características de arquitectura presenta la utilización de este patrón, esto implica que los juegos a desarrollar y posteriormente integrados a dicha plataforma apliquen de igual modo el MVC para lograr una coherencia desde el punto de vista arquitectónico, la aplicación del MVC permite caracterizar el modelo de diseño, logrando una estructura lógica que guía la construcción de los diagramas de clases del diseño.

3.2 Modelo de diseño

El Modelo de Diseño tiene como propósito especificar una solución funcional, que pueda ser fácilmente convertida en código fuente, construyendo una arquitectura simple y extensible, este modelo describe la realización física de los casos de uso, en este modelo los casos de usos son realizados por las clases del diseño y sus objetos (78).

Patrones de diseño

Un patrón es una descripción de un problema y la solución. Muchos patrones proporcionan guías sobre el modo en el que deberían asignarse las responsabilidades a los objetos, dada una categoría específica del problema. Todos los patrones tienen nombres sugerentes para que facilite la comunicación y apoye a la identificación del mismo (79).

Los patrones Grasp (*General Responsibility Assignment Software Patterns* traducido al español patrones generales de software para asignar responsabilidades) describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades, ayudan a entender el diseño del objeto esencial y aplican el razonamiento para el diseño de una forma sistemática, racional y explicable (80). Los que se evidencian en el diseño de la propuesta de solución son: Experto, Creador, Alta cohesión, Bajo acoplamiento y Controlador.

Experto: “Asignar una responsabilidad al experto en información la clase que tiene la información necesaria para realizar la responsabilidad” (79).

Se conserva el encapsulamiento de la información, ya que los objetos se valen de su propia información para realizar su tarea. En este caso, en la solución que se propone se hace uso de las buenas prácticas propuestas por el patrón. Evidencia de esto lo es la asignación de las operaciones a la clase Preguntas, en esta clase es donde se maneja únicamente toda la información referente a una pregunta, en ella se conforma la pregunta de acuerdo al nivel y la complejidad que debe poseer. De igual forma es observable en la clase C_Usuario, la cual contiene las responsabilidades y operaciones que competen al usuario así como sus características. Además es posible observar lo mismo en la realización de los diagramas de interacción (secuencia) expuesto en el próximo epígrafe.

Creador: “Asigna la responsabilidad al experto en la información, es decir, a la clase que cuenta con la información necesaria para cumplir con la responsabilidad” (79).

La intención básica del patrón Creador es encontrar un creador que necesite conectarse al objeto creado en alguna situación. Al aplicar este patrón estamos delegando a una clase la responsabilidad de crear objetos de otra o al menos a través de ella acceder a las operaciones de otra clase, en la solución que se propone la clase C_Usuario crea objetos de la clase Juego con el fin de poder acceder a algunos de sus eventos, tal y como se muestra a continuación:

```
juego = new Juego(nivel_actual);  
  
setInterval(juego.moverFiguras, 2000);  
  
setInterval(juego.pasarDeNivel, 600);  
  
setInterval(juego.marcadorDeTiempo, 1100);
```

De igual forma en la clase Juego se crea un objeto de la clase Preguntas a la hora de cargar en el juego los ejercicios que debe responder el jugador y como el experto en esa acción es la clase Preguntas solo ella puede crear las preguntas de acuerdo a los atributos necesarios.

Controlador: Es el responsable de controlar o manejar los eventos del sistema (79).

En el diseño de la propuesta de solución el uso de este patrón está asociado al manejo de los eventos asociados a las distintas funciones generales del juego, en este caso las clases RECUENTOFGC, CALCULOFGTC y ELEVADOR son las encargadas de manejar los eventos que permiten inicializar los juegos, almacenar los datos una vez que un jugador ha concluido el juego, controlan el flujo de transición de un jugador a otro cuando se juega en modo multijugador (multiplayer).

Bajo acoplamiento: “Asignar una responsabilidades de manera que exista bajo acoplamiento” (79).

El patrón Bajo Acoplamiento impulsa la asignación de responsabilidades de manera que su localización no incremente el acoplamiento hasta un nivel que nos lleve a los resultados negativos que puede producir un acoplamiento alto. En la investigación la asignación de responsabilidades que se realiza cuando se aplica el patrón Creador permite que se de soporte a la utilización de un bajo acoplamiento, en tal caso es

posible asociar que el acoplamiento global de una clase con otras es relativamente bajo. En la estructuración de las clases la creación de objetos y acceso a eventos de las clases se hace de forma secuencial, es decir la Clase C_Usuario accede a las operaciones de la clase Preguntas a través de un objeto de la clase Juego, esto permite que la clase C_Usuario solo esté lógicamente acoplada a la clase Juego. Otro de los factores influyentes en el bajo acoplamiento es la utilización del patrón arquitectónico MVC explicado al principio de este capítulo.

Alta cohesión: “Asignar una responsabilidad de modo que la cohesión siga siendo alta” (79).

Una clase con alta cohesión es ventajosa porque es relativamente fácil de mantener, entender y reutilizar, fundamentalmente porque realiza operaciones a fines con su objetivo. La utilización de los patrones antes mencionados y del patrón arquitectónico MVC, hacen posible que cada clase, realice las funciones únicas y exclusivas con su dominio, y el resto de las funciones sean delegadas en las clases expertas para su ejecución, tal y como se explicaba en el patrón anteriormente explicado.

3.3 Diagrama de clases del diseño

Este diagrama describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación, se realiza para tener en cuenta los detalles concretos de la implementación del sistema, en el diagrama la estructura de clases se especifica con relaciones existentes entre clases y objetos, además de modelar los conceptos relacionados con el dominio de la aplicación (81).

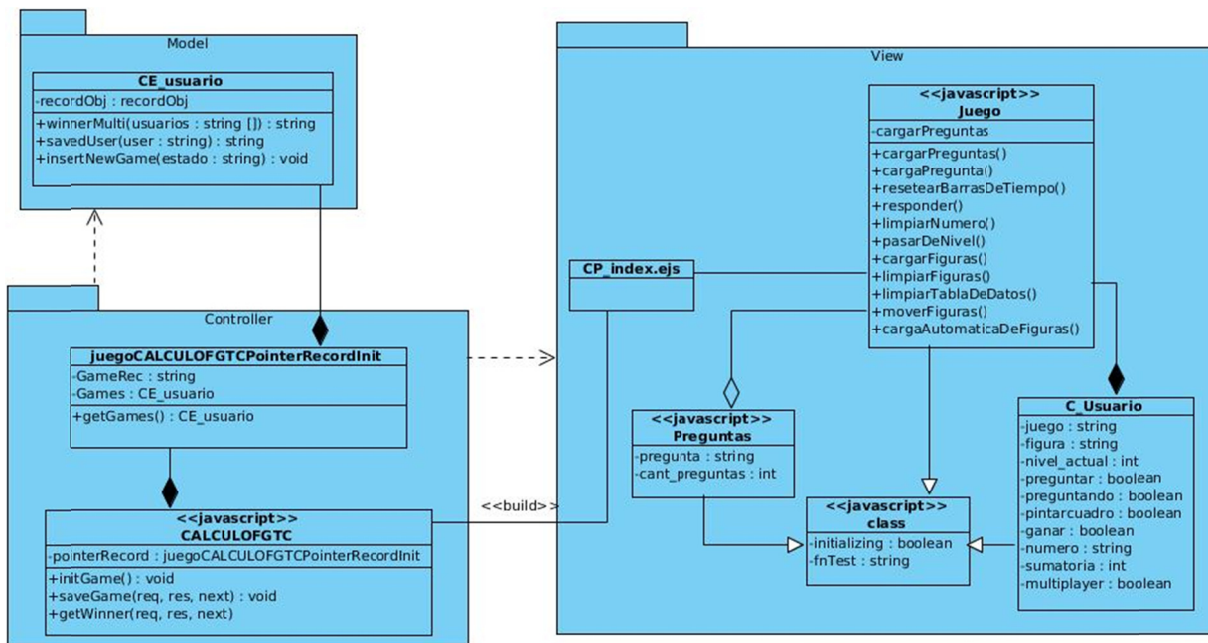


Figura 3. Diagrama de clases del diseño del juego Cálculo con figuras geométricas por tipos y colores

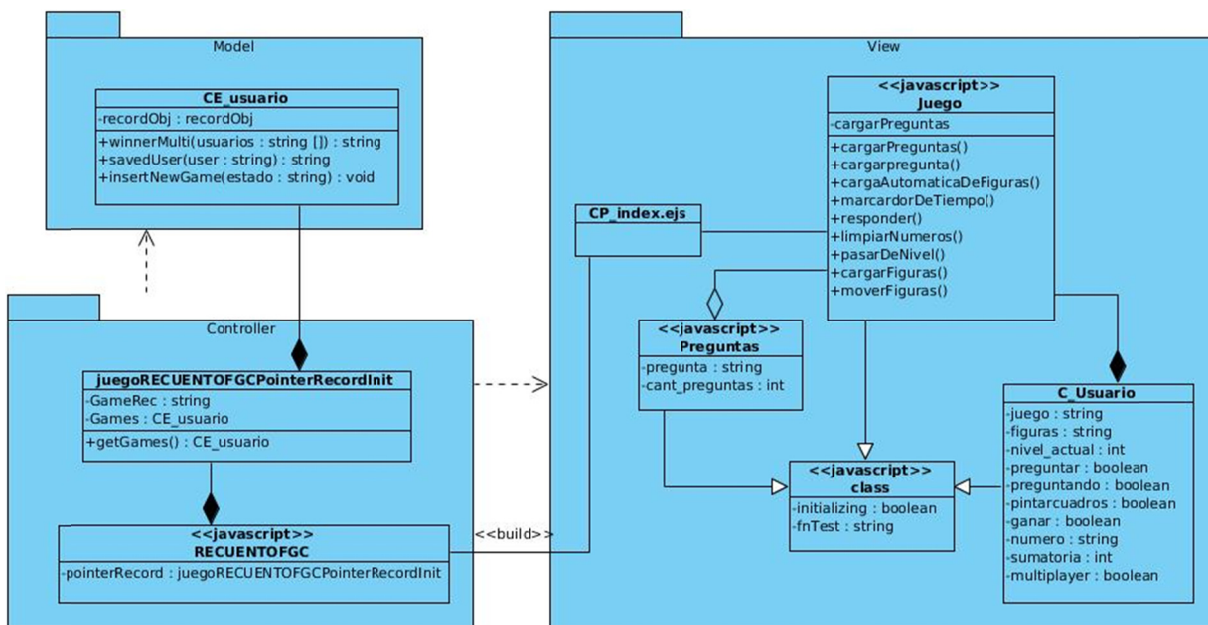


Figura 4. Diagrama de clases del diseño del juego Recuento de figuras geométricas y colores

Diagramas de secuencia

Muestra una interacción, que representa la secuencia de mensajes entre las instancias de clases, componentes o actores. En el diagrama el tiempo fluye hacia abajo y muestra el flujo de control de un participante a otro.

A continuación se muestra el diagrama de secuencia de los juegos **Recuento de figuras geométricas y colores**, **El elevador** y **Cálculo con figuras geométricas por tipos y colores**.

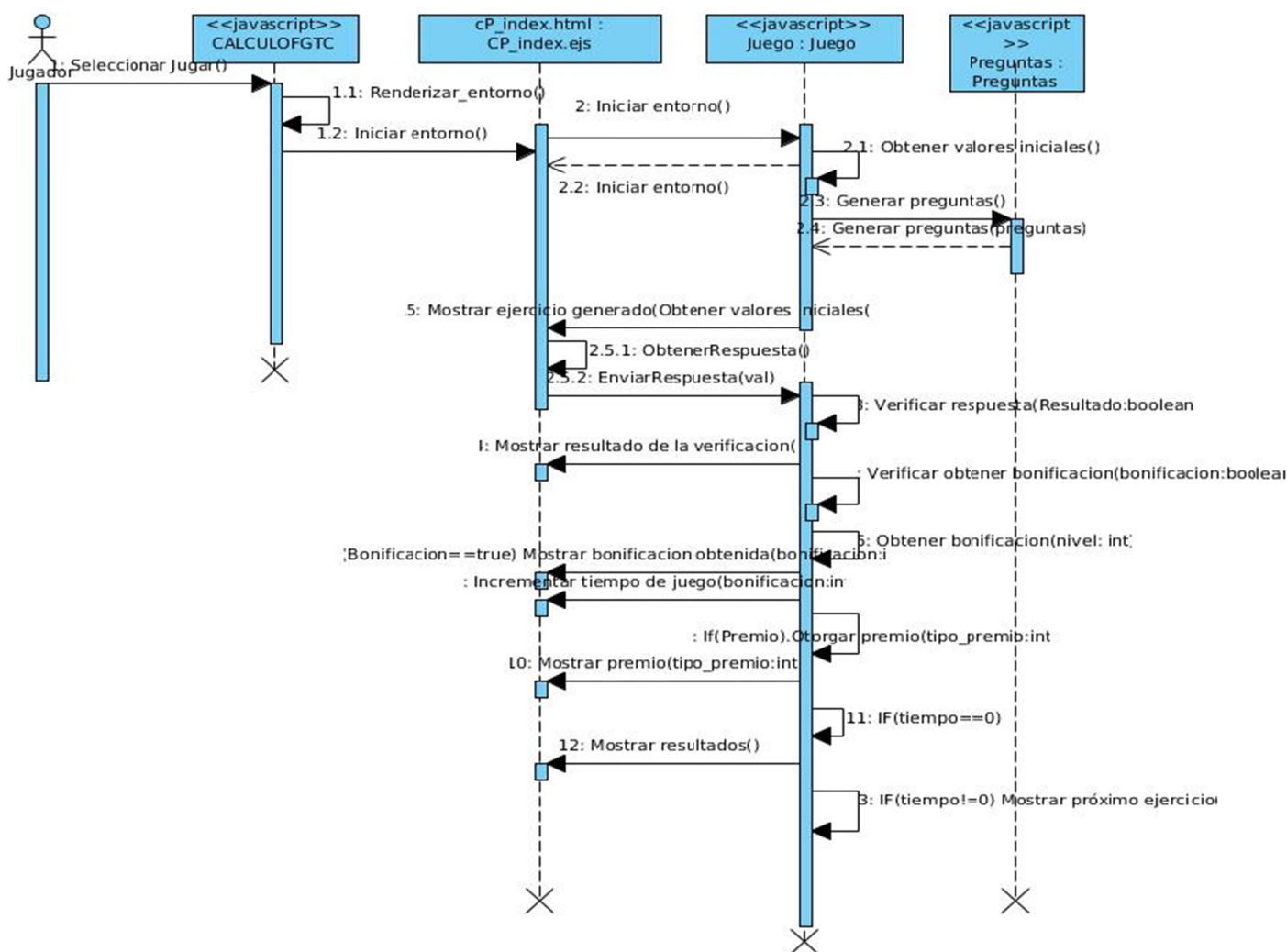


Figura 6. Diagrama de secuencia

3.5 Estándar de codificación

Los estilos de código o estándares de codificación son un conjunto de reglas o normas usadas para escribir código y que incluye una gran cantidad de aspectos dentro del proceso de codificación, El empleo de los mismo permite que los desarrolladores del sistema puedan interpretar de manera eficiente la escritura del código; asegurando que todos trabajen de forma coordinada y en un vocabulario común. (83).

A continuación se establecen las normas de codificación que serán utilizadas en el desarrollo de los juegos: **Recuento de figuras geométricas y colores**, **El elevador** y **Cálculo con figuras geométricas por tipos y colores**, las cuáles ya están definidas por la arquitectura de la plataforma de desarrollo Pointer, el estilo de código que se utilizará es:

CamelCase: es un estilo de escritura que se aplica a frases o palabras compuestas. El nombre se debe a que las mayúsculas a lo largo de una palabra en CamelCase se asemejan a las jorobas de un camello (84).

Nombres de las clases

Para los nombres de las clases se utilizará *UpperCamelCase*, o sea la primera letra de cada una de las palabras es mayúscula. Ejemplo: *EjemploDeUpperCamelCase*. Lo explicado anteriormente se muestra a continuación en algunos de los nombres de las clases de la propuesta de solución realizada.

```
var Juego = Class.extend ()
```

```
var Preguntas = Class.extend ()
```

Nombres de los métodos

Para el nombre de los métodos se utilizará *lowCamelCase*. La primera letra de cada una de las palabras es mayúscula a excepción de la primera. Ejemplo: *ejemploDeLowerCamelCase*. Lo explicado anteriormente se muestra a continuación en algunos de los nombres de los métodos de la propuesta de solución realizada.

```
cargarPregunta: function ()
```

cargaAutomaticaDeFiguras: function ()

marcadorDeTiempo: function ()

Conclusiones del capítulo

Durante el desarrollo del presente capítulo se obtuvieron una serie de artefactos ingenieriles necesarios en el ciclo de desarrollo de un software, tal y como especifica la metodología RUP. La estructuración del modelo de diseño, los patrones descritos y su aplicación en la elaboración de los diagramas de clases del diseño, permitieron refinar la línea base de la arquitectura, elemento de gran importancia y que proporciona una correcta guía a las actividades de implementación.

Capítulo IV. Implementación y prueba

Introducción

Luego de realizarse el diseño de cada juego a desarrollar, se cuenta con una vista más clara de cómo están estructurados los mismos. De esta manera queda todo listo para pasar a la etapa de implementación, donde se garantiza que cada funcionalidad cumpla con el diseño establecido. En el presente capítulo se describe el software en términos de componentes y la estrategia de prueba utilizada para verificar que el resultado de la implementación se corresponda con las especificaciones realizadas en los capítulos anteriores.

4.1 Modelo de implementación

El modelo de implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se podrá encontrar datos, archivos, ejecutables, código fuente y directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. Incluye suficiente información para construir el sistema. Debe incluir, no solamente la semántica lógica del sistema y los algoritmos, las estructuras de datos y los mecanismos que aseguran funcionamiento apropiado, sino también las decisiones de organización sobre los artefactos del sistema que son necesarios, permitiendo así el trabajo cooperativo de las personas y el procesamiento por parte de las herramientas (85).

La solución propuesta por la autora en esta investigación tiene como característica especial, que los tres juegos fueron desarrollados en entornos completamente web. Una vez que se concluyó la implementación de la propuesta de solución y las vistas o lógica de presentación, que la clase `index.html` se convirtió en `index.ejs` y en la clase controladora de cada una de las soluciones se modificó para que esta pudiera cargar la clase `index.ejs`, fue posible poder integrar la solución en la plataforma Pointer. Para lograr un correcto funcionamiento, teniendo en cuenta las características del patrón arquitectónico utilizado y de forma general las características de la arquitectura de Pointer. Lo antes expuesto tuvo como efecto que se realizara, tal y como se expone más adelante en este capítulo, dos rondas de pruebas.

El diagrama de componentes, que se expone a continuación, se realizó teniendo en cuenta la integración de la solución propuesta a la plataforma Pointer. Debido a esto se exponen características que no se tuvieron presentes en el entorno de desarrollo como lo es el componente index.ejs que se expone a continuación.

Diagrama de componentes

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño (86). El diagrama de componentes se utiliza para modelar la vista estática de un sistema. Se encarga de mostrar los componentes de software y las relaciones lógicas entre ellos en un sistema, además de estructurar el modelo de implementación en términos de subsistemas de implementación (87).

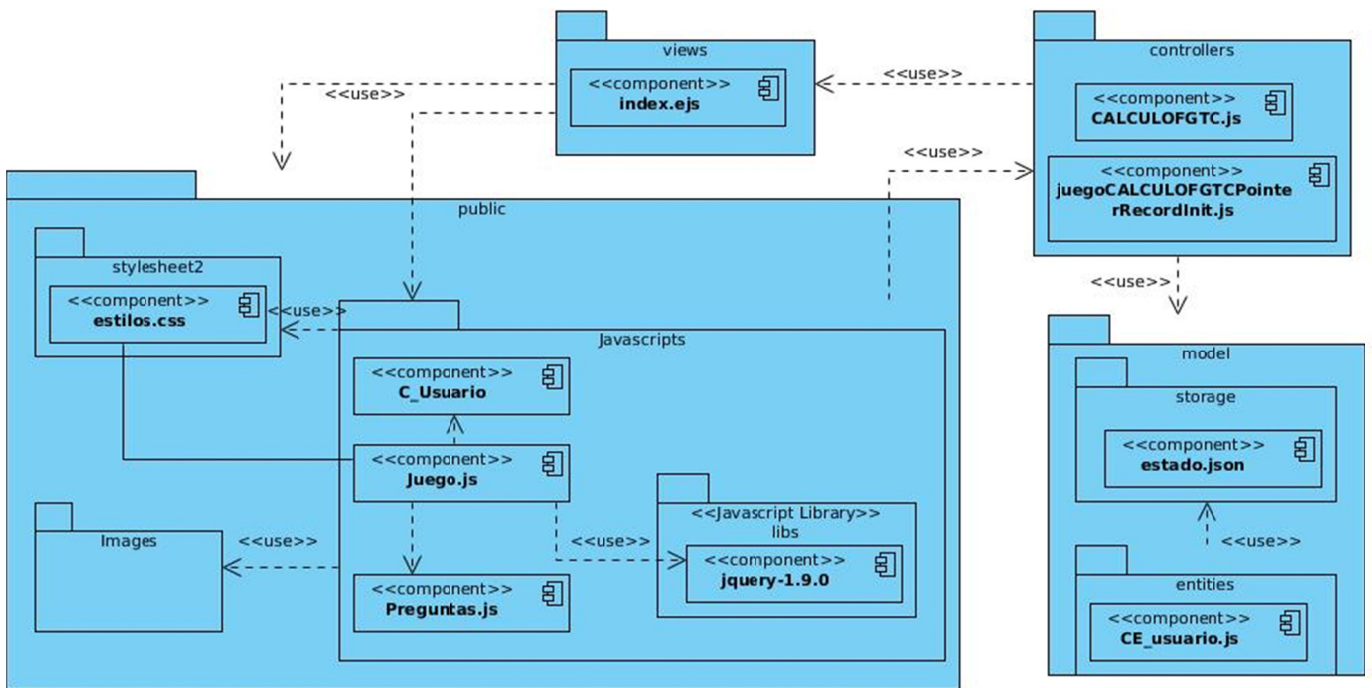


Figura 7. Diagrama de componente del juego Cálculo con figuras geométricas por tipos y colores

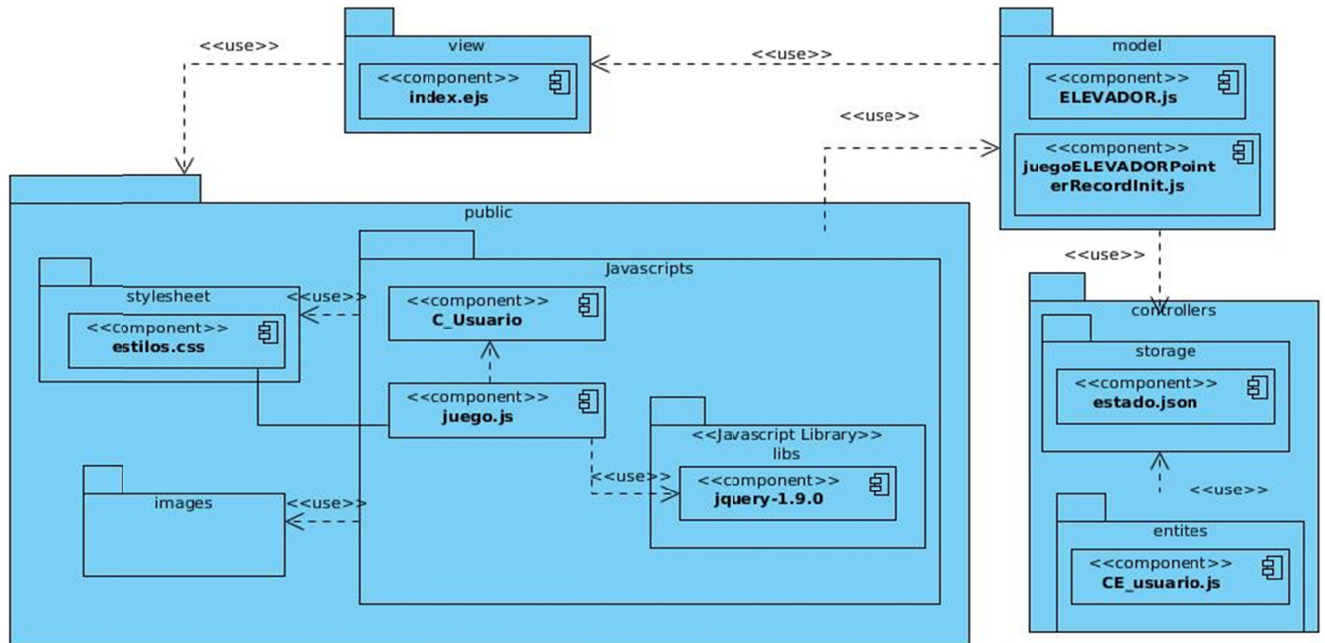


Figura 8. Diagrama de componente del juego El elevador

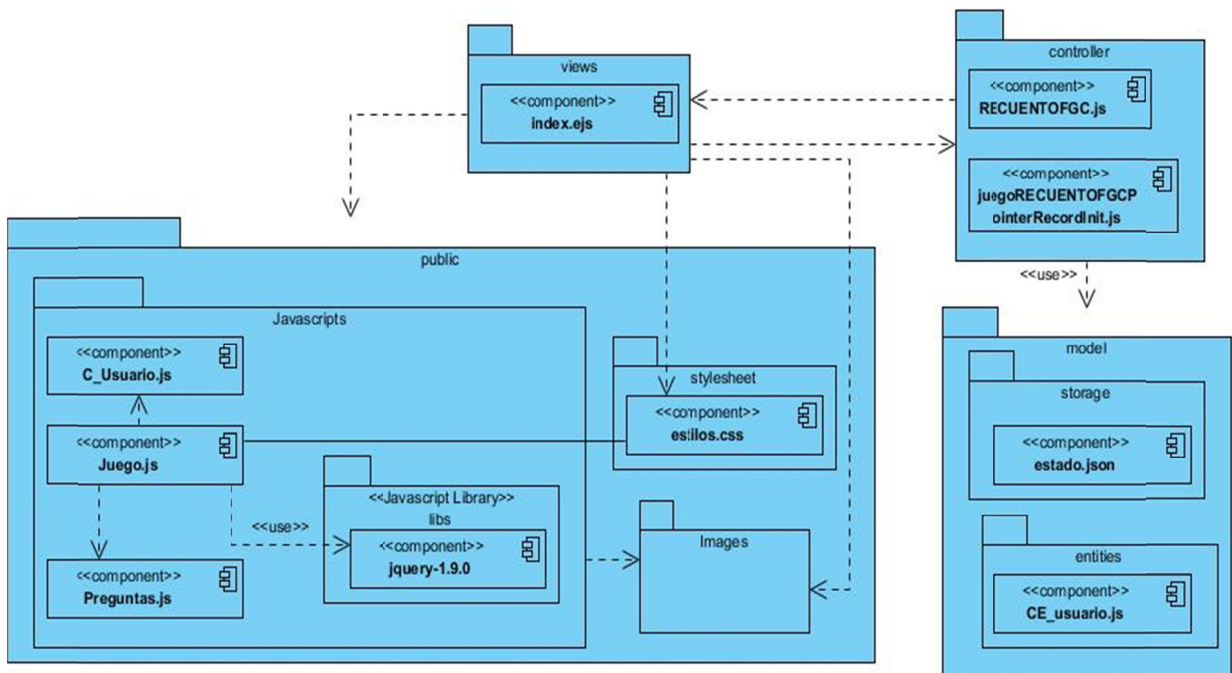


Figura 9. Diagrama de componente del juego Recuento de figuras geométricas y colores

Diagrama de despliegue

Muestra las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. Un nodo es un recurso de ejecución que puede ser un procesador o un dispositivo de hardware similar. Los componentes representan unidades de código en ejecución (87).

Se identificó el diagrama de despliegue que aparece a continuación para el modo de juego offline, el cual está compuesto por un nodo que tiene una computadora donde correrá la aplicación de manera local.



Figura 10. Diagrama de despliegue MundoClick

4.2 Pruebas

En el proceso de desarrollo de software se define una etapa de pruebas, con el objetivo de analizar si el producto final fue implementado correctamente. Las pruebas demuestran hasta qué punto las funciones del software parecen funcionar de acuerdo con las especificaciones (88). Existen dos métodos de pruebas: pruebas de caja blanca conocidas también como pruebas de caja de cristal, estas se centran en la estructura de control del programa. Otro método son las pruebas de caja negra que son diseñadas para validar los requisitos funcionales sin fijarse en el funcionamiento interno de un programa (89).

Para la realización de las pruebas a los productos obtenidos se establecieron una serie de pasos lógicos, con la finalidad de lograr una organización adecuada, ejecutar los procedimientos de pruebas de forma ordenada y garantizar encontrar la mayor cantidad de no conformidades (NC), así entregar un producto funcionalmente correcto. Estas series de pasos lógicos permitió definir una estrategia de pruebas basada fundamentalmente en evaluar el comportamiento de la propuesta de solución, para tal fin se ejecutaron pruebas al sistema, y pruebas de integración una vez que se incorporaron los juegos desarrollados en la plataforma Pointer. La realización de estas pruebas está guiada por el método de prueba de caja negra.

Pruebas de caja negra

Las pruebas de caja negra o también llamadas pruebas de comportamiento se centran en los requisitos funcionales del software, estas pruebas intentan encontrar errores de las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación (90).

Para la realización de las pruebas existen varias técnicas (90):

- Técnica de la partición de equivalencia: Divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Técnica del análisis de valores límites: Prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- Técnica de grafos de causa-efecto: Permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

En su gran mayoría las pruebas de validación y verificación de un producto de software están asociadas a las pruebas de caja negra, permiten reconocer si se ha construido el software correcto al contrastar los requisitos funcionales y no funcionales con el producto final.

Resultados de las pruebas

Las pruebas se realizaron en dos momentos fundamentales. Antes de integrar los productos a la plataforma Pointer y después de integrados. La razón fundamental de este procedimiento se basa en que los juegos fueron desarrollados en un entorno de desarrollo de aplicaciones web, sin tener en cuenta las características propias de la plataforma. Al ejecutar las tareas de integración con Pointer fue necesario realizar un conjunto de transformaciones mínimas para permitir que los juegos se ejecutaran a través de la plataforma Pointer.

Las NC encontradas responden a tres categorías, según las clasificaciones que utiliza el Centro Calisoft en la realización de pruebas de liberación:

- Error de formato y redacción (NCFR): este tipo de NC está relacionado tal y como su nombre lo indica al formato y redacción de los artefactos utilizados en las pruebas, en este caso particular en los DCP y descripciones textuales de los casos de uso como artefactos de apoyo.
- Error de Aplicación (NCA): comportamiento no esperado que ocurre en el sistema, que no se encontraba especificado ni descrito en los DCP.
- No correspondencia entre artefacto DCP y producto de software obtenido (NCCAP): está asociado con los mensajes que se emiten durante la ejecución de los juegos, pues estos no coincidían con los especificados en los DCP.

Estas no conformidades fueron resueltas con diferentes niveles de prioridad:

- Prioridad Alta: NC asociadas a errores de aplicación.
- Prioridad Media: NC asociadas a: No correspondencia entre artefacto DCP y producto de software obtenido.
- Prioridad baja: NC asociadas a errores de formato y redacción de los DCP.

A continuación se tabulan los resultados de las pruebas realizadas, la primera y segunda iteración se realizaron a la solución en sus entornos de desarrollo, la 3ra iteración y prueba final se ejecutaron una vez que se realizó la integración en la plataforma Pointer. En la prueba final no se encontraron NC.

Tabla 9. Resultados de las pruebas

	Iteración 1	Iteración 2	Iteración 3	Prueba final
NCFR	16	7	1	0
NCA	5	2	3	0
NCCAP	8	3	1	0

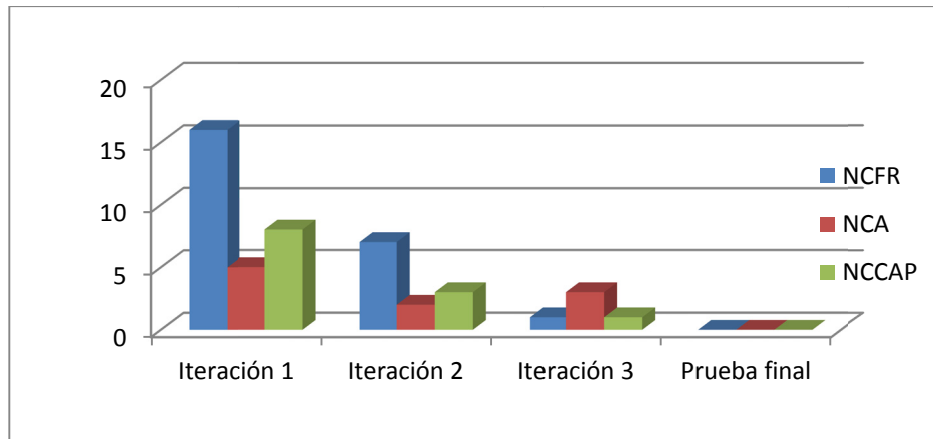


Figura 11. Resultado de las pruebas

Conclusiones del capítulo

En el presente capítulo quedó claramente reflejada la realización física de los elementos del diseño, uno de los resultados de este capítulo es el modelo de implementación. Las pruebas aplicadas al sistema dieron lugar a realizar mejoras sustanciales en la implementación de las capas de la lógica del negocio y presentación, logrando un mejor nivel de acabado del producto.

Conclusiones

El estudio del arte permitió obtener una fundamentación teórica que constata que la utilización de los juegos educativos contribuye al desarrollo y consolidación de habilidades en el individuo en la enseñanza Primaria.

El análisis de las soluciones similares permitió identificar características para incluir en la propuesta de solución que se realiza.

La obtención de los artefactos ingenieriles correspondientes a la metodología RUP permitió:

- Obtener una propuesta de solución acorde con las necesidades expuestas en la problemática y analizadas en el estudio del estado del arte.
- Propiciar una adecuada entrada a las actividades de implementación de la propuesta.

El tipo de prueba seleccionado y la realización de estas constataron un conjunto de insuficiencias no detectadas durante el proceso de desarrollo y permitieron verificar que la solución está acorde con los requerimientos especificados.

Recomendación

Se recomienda incorporar el diseño gráfico y los elementos pedagógicos definitivos a cada juego.

Referencias bibliográficas

1. *Nuevos medios para nuevas formas de aprendizaje: El uso de los videojuegos en la enseñanza*. **Gros Salvat, Begoña**. 3, Barcelona : Red digital: Revista de Tecnologías de la Información y Comunicación Educativas., 2002. 1696-0823.
2. *El uso de los videojuegos como recurso de aprendizaje en educación primaria y Teoría de la Comunicación*. **Morales Corral, Enrique**. 78, Madrid : REVISTA ACADÉMICA DE LA FEDERACIÓN LATINOAMERICANA DE FACULTADES DE COMUNICACIÓN SOCIAL, 2009. 1995 - 6630.
3. *Técnicas de la inteligencia artificial aplicadas a la educación*. **Sánchez Vila, Eduardo M y Lama Penín, Manuel**. 33, 2007, Revista Iberoamericana de inteligencia artificial, Vol. 11, págs. 7-12. 1137-3601.
4. **Rodríguez Rodríguez, Livan A**. *Módulo Resultados de la Colección Futuro de preuniversitario*. Instituto Superior Pedagógico "Félix Varela". 2005. online www.bvs.hn/cu-2007/ponencias/MUL/MUL074.pdf.
5. **Mined. Colectivo de autores**. *Uso del Software Educativo en la escuela cubana y su impacto en el aprendizaje de los estudiantes*. La Habana : Mined, 2009. online <http://mediateca.rimed.cu/media/document/4771.pdf>. 978-959-18-0448-8.
6. **Ulloa Reyes, Luis Gaspar, y otros**. *Los juegos educativos por computadora como apoyo al proceso de Enseñanza-Aprendizaje*. Camagüey : Instituto Superior Pedagógico "José Martí", 2008. pág. 3.
7. *El uso de los videojuegos como recurso de aprendizaje en educación primaria y Teoría de la Comunicación*. **Morales Corral, Enrique**. 78, Madrid : REVISTA ACADÉMICA DE LA FEDERACIÓN LATINOAMERICANA DE FACULTADES DE COMUNICACIÓN SOCIAL, 2009, págs. 1-2. 1995 - 6630.
8. **Egenfeldt-Nielsen, Simon, Heide-Smith, Jonas y Pajares Tosca, Susana**. *Understanding Videogames*. London : s.n., 2008. págs. 280-281.
9. *La importancia del juego*. **Jiménez Rodríguez, Esmeralda**. 26, Sevilla : Revista Digital "Investigación y Educación", 2006, Vol. III, págs. 3-4. 1696-7208.
10. **Huizinga, Johan**. *Homo Ludens: el elemento lúdico de la Cultura*. Madrid : Alianza, 2003.

11. *La importancia del juego*. **Jiménez Rodríguez, Esmeralda**. 26, Sevilla : Revista Digital "Investigación y Educación", 2006, Vol. III, págs. 1-2. 1696-7208.
12. Juegos educativos para niños. *Juegos educativos para niños*. [En línea] 28 de Marzo de 2012. [Citado el: 11 de Diciembre de 2012.] http://pascalnaybejuegoseducativos.blogspot.com/2012/03/cuales-son-las-caracteristicas-de-los_28.html.
13. *La importancia del juego*. **Jiménez Rodríguez, Esmeralda**. 26, Sevilla : Revista Digital "Investigación y Educación", 2006, Vol. III, págs. 10-11. 1696-7208.
14. —. **Jiménez Rodríguez, Esmeralda**. 26, Sevilla : Revista Digital "Investigación y Educación", 2006, Vol. III, págs. 9-10. 1696-7208.
15. **Marquès, Pere**. *El software educativo*. Barcelona : Universidad Autónoma de Barcelona, 1999. pág. 1.
16. **Galvis Panqueva, Alvaro H**. *Ingeniería de Software Educativo*. Colombia : Ediciones Uniandes, 2001. 958-9057-25-X.
17. Proyecto Huascarán. *Proyecto Huascarán*. [En línea] MINISTERIO DE EDUCACIÓN DEL PERÚ, 2006. [Citado el: 12 de Diciembre de 2012.] <http://www.huascarán.gob.pe>.
18. **Gros, Begoña**. *Del software educativo a educar con software*. Barcelona : Universidad de Barcelona. pág. 1.
19. **Núñez Rojas, Nemecio**. *LA WEBQUEST, EL AULA VIRTUAL Y EL DESARROLLO DE COMPETENCIAS PARA LA INVESTIGACIÓN EN LOS ESTUDIANTES DEL I CICLO DE EDUCACIÓN – USAT*. PERÚ : Universidad Católica de Santo Toribio de Mogrovejo, 2009. pág. 34.
20. Software Educativo. *Software Educativo*. [En línea] 18 de Marzo de 2007. [Citado el: 12 de Diciembre de 2012.] <http://tecno-educativa.blogspot.com/2007/03/software-definición-y-características.html>.
21. **González Neri, Ysauro, Carmona Martínez, Victoria y Espíritu Reyes, Sara**. *Evaluación de software educativo*. México : INSTITUTO LATINOAMERICANO DE LA COMUNICACIÓN EDUCATIVA Unidad de Investigación y Modelos Educativos, 1998.

22. **Núñez Rojas, Nemecio.** *LA WEBQUEST, EL AULA VIRTUAL Y EL DESARROLLO DE COMPETENCIAS PARA LA INVESTIGACIÓN EN LOS ESTUDIANTES DEL I CICLO DE EDUCACIÓN – USAT.* Perú : Universidad Católica Santo toribio de Mogrovejo, 2009. pág. 36.
23. **Marquès, Pere.** *El software educativo.* Barcelona : Universidad Autónoma de Barcelona, 1999. págs. 12-14.
24. **Pérez Latorre, Óliver.** *Análisis de la significación del videojuego: Fundamentos teóricos del juego, el mundo narrativo y la enunciación interactiva como perspectivas de estudio del discurso.* s.l. : Universitat Pompeu Fabra, 2010.
25. *Videojuegos como dispositivos culturales: las competencias espaciales en educación.* **Sedeño, Ana.** 34, Málaga : Revista Científica de Educomunicación, 2010, Vol. XVII. 1134-3478.
26. **Ferrer Vicente, Maribel.** *La resolución de problemas en la estructuración de un sistema de habilidades matemáticas en la escuela media cubana.* Santiago de Cuba : s.n., 2000.
27. **Etxeberría, Félix.** *Videojuegos y educación.* San Sebastián : Reflexiones, 1998.
28. **Álvarez Figueroa, Oneida.** *El sistema educativo cubano en los noventa.* La Habana : Centro de Investigaciones de Economía Internacional.Universidad de La Habana., 1997.
29. *Estimulación y relax mental a través del uso de video pasatiempos de última generación.* **Tapia Meléndez, Laura.** 8, Madrid : Universidad Complutense de Madrid, 2006. 1697 - 8293.
30. Real Academia Española. *Real Academia Española.* [En línea] [Citado el: 16 de Junio de 2013.] <http://lema.rae.es/drae/?val=habilidad>.
31. **Márquez Rodríguez, Dra Aleida.** *Habilidades: reflexiones y proposiciones para su evaluación.* Santiago de Cuba : Instituto Superior Pedagógico, 1993.
32. **Alvarez de Zayas, Carlos Manuel.** *La escuela de la vida. Colección Educación y Desarrollo.* La Habana : Felix Varela, 1992. 9590700152.

33. **Ramírez Uclés, Rafael.** *Habilidades de visualización de los alumnos con talento matemático.* Granada : Universidad de Granada, 2012.
34. **Colectivo de autores.** *Metodología de la enseñanza de la Matemática en la escuela primaria.* Ciudad de La Habana : Pueblo y Educación, 1992. pág. 88.
35. **Castillo Montes, Mayra Virginia.** *La educación matemática en el Primer Ciclo de la Educación Primaria.* Guatemala : s.n., 2010.
36. Oficina Internacional de Educación. *Oficina Internacional de Educación.* [En línea] [Citado el: 16 de Junio de 2013.] http://www.ibe.unesco.org/National_Reports/ICE_2008/cuba_NR08_sp.pdf.
37. **Rodríguez Rodríguez, Jorge Remberto.** *Arquitectura de software. Guía base. Proyecto MundoClick.* 2013.
38. **Araujo Pérez, Liana Isabel.** ¿Cuál es la audiencia de los juegos a desarrollar? *Arquitectura de la plataforma Pointer.* La Habana, 25 de Enero de 2013.
39. El camino de Hexamano: Matemáticas divertidas. *El camino de Hexamano: Matemáticas divertidas.* [En línea] 19 de Agosto de 2011. [Citado el: 10 de Enero de 2013.] <http://vecindarioclic.blogspot.com/2011/08/el-camino-de-hexamano.html>.
40. Análisis de aplicación:gbrainy-Centro de apoyo tecnológico de emprendedores. *Análisis de aplicación:gbrainy-Centro de apoyo tecnológico de emprendedores.* [En línea] [Citado el: 11 de Enero de 2013.] <http://www.bilib.es/noticias/noticia/articulo/analisis-de-aplicacion-gbrainy/>.
41. Brain Juice: Gimnasia para el cerebro. *Brain Juice: Gimnasia para el cerebro.* [En línea] Digital Chocolate. [Citado el: 10 de Enero de 2013.] http://www.ciao.es/Brain_Juice__Opinion_1169988.
42. **María Ramos Rodríguez.** Cálculo Mental. *Cálculo Mental.* [En línea] febrero de 2009. [Citado el: 10 de Enero de 2013.] http://www2.gobiernodecanarias.org/educacion/17/WebC/eltanque/todo_mate/calculo_m/calculomental_p_p.html.
43. **Pressman, Roger A.** *Ingeniería del Software: Un Enfoque Práctico.* 2005. 9701054733.

44. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software.* Madrid : Pearson Educación, S.A., 2000. págs. 10-12. 84-7829-036-2.
45. —. *El proceso unificado de desarrollo de software.* Madrid : Pearson Educación, S.A., 2000. págs. 5-8. 84-7829-036-2.
46. **Larman, Craig.** *UML y Patrones: Introducción al análisis y diseño orientado a objetos.* México : PRENTICE HALL, 1999. 970-17-0261-1.
47. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *El Lenguaje Unificado de Modelado.* California : Addison Wesley, 1998. págs. 3-4.
48. **López Pecho, Ramiro y Cesar Ballesteros, Julio.** Herramientas CASE. *Herramientas CASE.* [En línea] 29 de Septiembre de 2008. [Citado el: 7 de Diciembre de 2012.] <http://tpsis324.blogspot.com/>.
49. *Herramientas libres para modelar software.* **Callejas Cuervo, Mauro y Baquero Moreno, Oscar Yovany.** Colombia : Revista Facultad de Ingeniería, UPTC, 2005. 0121–1129.
50. **Sierra, Antonio J.** *Lenguajes de programación.* 2011.
51. **Goldstein, Alexis, Lazaris, Louis y EstelleWeyl.** *HTML5 y CSS3 for the real world.* USA La desventaja es que solamente los navegadores modernos soportan los nuevos efectos. : SitePointPty. Ltd, 2011. 978-0-9808469-0-4.
52. **Eguíluz Pérez, Javier.** CSS avanzada. *CSS avanzada.* [En línea] 2 de Enero de 2009. [Citado el: 3 de Diciembre de 2012.] <http://www.librosweb.es/css/>.
53. —. Introducción a JavaScript. *Introducción a JavaScript.* [En línea] 7 de junio de 2008. [Citado el: 3 de Diciembre de 2012.] <http://www.librosweb.es/javascript/>.
54. Tutorial de JavaScript. *Tutorial de JavaScript.* [En línea] Universidad de Malaga. España, 29 de Octubre de 2012. [Citado el: 5 de Diciembre de 2012.] <http://www.lcc.uma.es/~eat/services/html-js/manual13.html>.

55. **Diego Gauchat, Juan.** *El gran libro de HTML5, CSS3 y JavaScript.* Barcelona : Marcombo, S.A., 2012. 978-84-267-1782-5.
56. —. *El gran libro de HTML5, CSS3 y Javascript.* Barcelona : MARCOMBO, S.A., 2012. págs. 1-2. 978-84-267-1782-5.
57. **Luca, Damián De.** *CSS3 & HTML5 Los nuevos estándares para el diseño y el desarrollo web. CSS3 & HTML5 Los nuevos estándares para el diseño y el desarrollo web.* [En línea] 10 de 10 de 2010. [Citado el: 5 de Diciembre de 2012.] <http://css3html5.com.ar/que-es-html5/>.
58. **Sánchez-Heredero Pérez, Alberto.** *Accesibilidad a los contenidos audiovisuales en la Web a través de HTML5.* Leganés : Universidad Carlos III de Madrid, 2011.
59. w3schools.com. *w3schools.com.* [En línea] [Citado el: 6 de Diciembre de 2012.] http://www.w3schools.com/html/html5_intro.asp.
60. w3schools. *w3schools.* [En línea] [Citado el: 7 de Enero de 2013.] <http://www.w3schools.com/json/default.asp>.
61. Introducción a JSON. *Introducción a JSON.* [En línea] [Citado el: 7 de Enero de 2013.] <http://www.json.org/json-es.html>.
62. **Departamento de Ingeniería Civil.** *Examen Final de Programación Avanzada Tercer Curso.* Burgos : s.n., 2008.
63. **Sánchez, Zoraida Hidalgo.** *Diseño e implementacion de una BD de investigacion.* Barcelona : s.n., 2007. pág. 86.
64. DesarrolloWeb. *DesarrolloWeb.* [En línea] 19 de Septiembre de 2012. [Citado el: 5 de Diciembre de 2012.] <http://www.desarrolloweb.com/manuales/manual-jquery.html>.
65. Introducción a Node.js. *Introducción a Node.js.* [En línea] 1 de Mayo de 2011. [Citado el: 8 de Enero de 2013.] <http://www.rmuno.net/introduccion-a-node-js.html>.

66. Needume: Software,Diseño y Web. *Needume: Software,Diseño y Web*. [En línea] Needume Page. [Citado el: 9 de Enero de 2013.] <http://needume.blogspot.com/2013/01/que-es-nodejs-y-como-comenzar-en-el.html>.
67. **Jaime**. Entornos de Desarrollo Integrados. *Entornos de Desarrollo Integrados*. [En línea] 14 de Enero de 2009. [Citado el: 6 de 12 de Diciembre.] <http://petra.euitio.uniovi.es/~i1667065/HD/documentos/Entornos%20de%20Desarrollo%20Integrado.pdf>.
68. **Maldonado, Daniel**. El Codigo k:que son los IDE de programación. *El Codigo k:que son los IDE de programación*. [En línea] 7 de Septiembre de 2007. [Citado el: 6 de Diciembre de 2012.] <http://elcodigok.blogspot.com/2007/09/que-son-los-ide-de-programacin.html>.
69. Netbeans IDE. *Netbeans IDE*. [En línea] 2012. [Citado el: 6 de Diciembre de 2012.] <http://netbeans.org/features/index.html>.
70. **Larman, Craig**. *UML y Patrones: Introducción al análisis y diseño orientado a objetos*. México : PRENTICE HALL, 1999. pág. 11. 970-17-0261-1.
71. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady**. *El proceso unificado de desarrollo de software*. Madrid : Pearson Educación, S.A, 2000. págs. 106-107. 84-7829-036-2.
72. —. *El proceso unificado de desarrollo de software*. Madrid : Pearson Educación, S.A, 2000. págs. 110-111. 84-7829-036-2.
73. —. *El Lenguaje Unificado del Modelado*. California : Addison Wesley, 1998. págs. 365-366.
74. —. *El lenguaje unificado de modelado*. California : Addison Wesley, 1998. pág. 56.
75. **Larman, Craig**. *UML y Patrones: Introducción al análisis y diseiio orientado a objetos*. México : PRENTICE HALL, 1999. pág. 55. 970-17-0261-1.
76. **Schmidt, Douglas, et al**. *Pattern-Oriented Software Architecture, Patterns for Concurrent and Networked Objects*. London : John Wiley & Sons, Inc., 2000. 0-471-60695-2.
77. **CyberFicient Technologies Inc**. *Modell-View-Controller, Architecture Overview*. USA : s.n., 2001.

78. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software.* Madrid : Pearson Educación, S.A., 2000. pág. 208. 84-7829-036-2.
79. **Larman, Craig.** *UML y Patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado.* 2. s.l. : Pearson Prentice Hall, 2003. págs. 162-170. 9788420534381.
80. —. *UML y Patrones: Introducción al análisis y diseño orientado a objetos.* México : PRENTICE HALL, 1999. págs. 185-192. 970-17-0261-1.
81. **Schmuller, Joseph.** *Aprendiendo UML en 24 horas.* México : PEARSON EDUCACION,S.A, 2000. 968-444-463-X.
82. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *El Lenguaje Unificado de Modelado.* California : Addison Wesley, 1998. págs. 25-27.
83. **Arias Calleja, Manuel.** Carmen. Estándares de codificación. *Carmen. Estándares de codificación.* [En línea] [Citado el: 1 de Abril de 2013.] <http://www.cisiad.uned.es/carmen/estilo-codificacion.pdf>.
84. pedrormarcos.com. *pedrormarcos.com.* [En línea] 2012. [Citado el: 29 de Marzo de 2013.] <http://www.pedrormarcos.com/escritura-estilo-camelcase/>.
85. **Pressman, Roger A.** *Ingeniería de software: Un enfoque práctico.* 5. 2005. pág. 404. 970105473.
86. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *Proceso unificado de desarrollo de software.* Madrid : Pearson Educacion.SA, 2000. pág. 287. 84-7829-036-2.
87. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *El lenguaje unificado de modelado.* California : Addison Wesley, 1998. págs. 212-213.
88. **Pressman, Roger A.** *Ingeniería del Software:Un enfoque práctico.* 2005. págs. 281-284. 970105473.
89. —. *Ingeniería del Software:Un enfoque práctico.* 2005. págs. 284-293. 970105473.
90. —. *Ingeniería del Software:Un enfoque práctico.* 2005. págs. 294-302. 970105473.

Glosario de términos

Aprendizaje basado en juegos: trata de utilizar el poder de los juegos de ordenador para atraer y motivar a los estudiantes para conseguir que estos desarrollen nuevos conocimientos y habilidades. Este tipo de aprendizaje permite realizar tareas y experimentar situaciones que de otro modo serían imposibles de realizar por cuestiones de coste, tiempo, infraestructura.

Constructivismo: corriente pedagógica creada por Ernst von Glasersfeld, basándose en la teoría del conocimiento constructivista, que postula la necesidad de entregar al alumno herramientas (generar andamiajes) que le permitan crear sus propios procedimientos para resolver una situación problemática, lo cual implica que sus ideas se modifiquen y siga aprendiendo.

Proceso Enseñanza-Aprendizaje: el movimiento de la actividad cognoscitiva de los alumnos bajo la dirección del maestro, hacia el dominio de los conocimientos, las habilidades, los hábitos y la formación de una concepción científica del mundo.

Anexo 1 Guión del juego El elevador

Historia:

La historia se basa en un niño que gustaba de descubrir cosas y hacerse preguntas sobre todo lo que le rodeaba. Lo que lo conlleva a desarrollar el gusto por las ciencias.

*Alejandrino(a) era un niño(a) con mucha imaginación. Pasaba horas observando los objetos y eventos que ocurrían a su alrededor preguntándose y tratando de dar respuestas a todo cuanto veía y no entendía. Muchas veces observando el capullo de las mariposas llegó a pensar que si se encerraba por largo tiempo dentro de su cuarto podría convertirse en uno de los súper héroes que veía en la TV. Usualmente pasaba horas con su abuelo que le hacía historias de su niñez y casi siempre tenía una buena respuesta a las inocentes preguntas que Alejandrino(a) le hacía. Una vez preguntó cuál era la razón por la que llovía y su abuelo le explicó que las nubes eran como esponjas o algodones, que guardaban pequeñas gotitas de agua que se desprendían de los ríos y mares. Estas esponjas al llenarse dejaban escapar las gotitas y por eso llovía.

**Alejandrino(a) fue creciendo y entró a la escuela, poco a poco iba entendiendo y dando respuestas a aquellas cosas que de niño(a) respondía a su manera. Un día en su clase de matemáticas el maestro les enseñaba a sumar y restar objetos y sobre las miles maneras en las que usábamos los cálculos en nuestra vida en diferentes situaciones. Explicaba además que nuestro cerebro constantemente realizaba cálculos y que estos eran una buena manera de ejercitarlo. Ese día realizaron varios juegos de agilidad mental que les ayudaba a familiarizarse con las operaciones de cálculo. Como especie de tarea les recomendó que llevaran a la próxima clase un ejemplo de una de estas situaciones en forma de juego y que fuera premiada la tarea más creativa.

Ese día Alejandrino(a) llegó a su casa con un reto y pasó toda la tarde pensando en una manera ingeniosa de realizar su tarea. Pensaba y pensaba pero no se le ocurría nada creativo.

Casi llegada la noche venía de visitar a un amigo, Pedrito que vivía en el primer piso de su edificio y un tanto frustrado por no encontrar una buena idea se dispuso a usar el elevador, como todos los días, hasta el 5to piso donde se encontraba su apartamento. Junto a él subieron 3 personas más, Ramón del 3er piso, y la señora Alicia con su niño que iban hasta el 4to. Llegado al 2do piso se les unió Felipe, un joven que estaba de visita en un apartamento de su mismo piso. Alejandrino(a) pensó, ya somos 5 personas y en un

espacio tan reducido. Al tercer piso Ramón se baja del elevador y Alejandrino(a) repite el conteo observando que volvían a ser 4 personas. En el 4to piso se bajan la señora Alicia y su hijo, ya solo quedamos 2, pensó Alejandrino(a). Por fin el piso 5. Alejandrino(a) corre a casa y se dirigió a realizar su baño de todos los días aún molesto porque no había encontrado un buen ejercicio para su tarea y ya se le acababa el tiempo.

Llegó la hora de dormir y el/la niño(a) no se sentía conforme pues no había sido capaz de encontrar una situación para cumplir con su tarea. Casi llegando el sueño recordó su paso por el elevador y trató de recordar en cada piso quien subía y quien bajaba del mismo. Poco a poco se dio cuenta de que en ese momento estaba realizando inconscientemente operaciones de cálculo y apelaba a su memoria tratando de recordar quien entraba y salía en cada piso. -¡Genial!-, dijo dando un salto en la cama y corrió a escribir lo que había sucedido en su cuaderno.

Al otro día explicó en que consistía su juego y todos se encantaron con la brillante idea de Alejandrino(a). -¿Y como llamarás a tu juego?-, preguntó el maestro. -El elevador- respondió, después de todo fue aquí donde nació. Desde ese día Alejandrino(a) practicaba al llegar a casa y muchos de sus compañeros repetían el juego que surgió de una manera tan peculiar.

Descripción de los elementos gráficos:

- El ambiente principal del juego representará el espacio de un elevador ascendiendo por una cantidad dada de pisos dentro de un edificio.
- En la parte superior una barra de avance (o retroceso) o un reloj que muestre el paso del tiempo total disponible para resolver los ejercicios.
- En la parte inferior se ubicará otra barra con el campo de entrada donde se escribirá el resultado de los ejercicios.
- Las puertas del elevador se deben ilustrar de manera separada para permitir realizar las animaciones para abrir y cerrar las puertas.

Cada elemento X que se describe posteriormente será representado por una imagen.

La bonificación que se describe posteriormente será representada por la imagen de un reloj.

Los aciertos y desaciertos serán indicados en texto, imágenes o ambos.

Los premios serán representados por relojes de arena de: oro, plata y bronce.

El elevador:

El juego consiste en un elevador que debe ascender N pisos en los cuales se va llenando o vaciando de $X_1, X_2, X_3, \dots, X_n$ tipos de objetos. Al llegar al piso N se debe introducir la suma de los objetos X que quedaron en el elevador. La complejidad aumenta de acuerdo con la cantidad de pisos N , cantidad de objetos X_n y combinaciones de objetos en cada piso.

Ej. de una posible situación para $N = 3$, $X_1 =$ Hombres, $X_2 =$ Mujeres, $X_3 =$ Niños.

Piso 1: Se suben 2 mujeres.

Piso 2: Se sube 1 hombre y se baja 1 mujer.

Piso 3: Se suben 3 hombres, 2 niños y se baja 1 mujer.

Dada la situación anterior posible preguntas serían:

- Cuántos niños llegan al piso N .
- Cuántos hombres y mujeres llegan al piso N .

Si se quisiera aumentar la complejidad y poner a mayor prueba la memoria se podría preguntar cuántos objetos o suma de objetos X llegan a algún piso intermedio.

Tiempo de juego y niveles:

Inicialmente se cuenta con tiempo total $T = 1$ minuto (60s) para realizar la mayor cantidad de ejercicios E posibles.

Nivel 1:

- Como valor inicial cada situación contará con una cantidad de pisos $N = 3$.
- Los objetos X serán $X_1 =$ Hombres y $X_2 =$ Mujeres.
- Cuando el jugador logre realizar en este nivel 5 ejercicios o más se desbloquea el próximo nivel.

Nivel 2:

- Cuando el jugador logre realizar en este nivel 7 ejercicios o más se desbloquea el próximo nivel.
- Como valor inicial cada situación contará con una cantidad de pisos $N = 4$.

- Los objetos X serán $X_1 = \text{Hombres}$, $X_2 = \text{Mujeres}$ y $X_3 = \text{Niños}$.

Nivel 3:

- Cuando el jugador logre realizar en este nivel 9 ejercicios o más se desbloquea el próximo nivel.
- Como valor inicial cada situación contará con una cantidad de pisos $N = 5$.
- Los objetos X serán $X_1 = \text{Hombres}$, $X_2 = \text{Mujeres}$, $X_3 = \text{Niños}$.

Bonificaciones:

Las bonificaciones consisten en aumentar el tiempo de juego en el reloj, dependiendo del desempeño del jugador.

Para el 1er nivel

- Si es capaz de responder 3 ejercicios seguidos, cada uno en 6 segundos o menos aplicar una bonificación de 9 segundos al tiempo total.

Para el 2do nivel

- Si es capaz de responder 4 ejercicios seguidos, cada uno en 4 segundos o menos aplicar una bonificación de 6 segundos al tiempo total.

Para el 3do nivel

- Si es capaz de responder 5 ejercicios seguidos, cada uno en 3 segundos o menos aplicar una bonificación de 4 segundos al tiempo total.

La estrategia propuesta para determinar las bonificaciones consiste en dividir el tiempo (60s) en 5, 7 y 9 partes o ejercicios E mínimos para los niveles 1, 2 y 3 respectivamente. Con un periodo $P=T/E$ (cantidad de segundos mínimos por ejercicio) de 12, 8 y 6 s/e (segundos/ejercicio) respectivamente. Si responde el 60% de los ejercicios E, cada uno en un tiempo t_n menor o igual a la mitad del tiempo mínimo por ejercicio determinado en P ($t_n \leq 1/2P$) obtendrá una bonificación de dicho tiempo menos la cuarta parte de este ($B = P - 1/4P$).

Premios:

Los premios serán dados en base a la efectividad que se obtenga en cada nivel.

efectividad = cantidad de preguntas correctas / cantidad total de preguntas

Anexo 2 Guión del juego Cálculo con figuras geométricas por tipos y colores

Historia:

Descripción de los elementos gráficos:

- En la parte inferior a la izquierda una barra de avance (o retroceso) que muestre el paso del tiempo total disponible para resolver los ejercicios.
- En la parte derecha se ubicará un campo donde se visualizará el resultado de los ejercicios.
- Cada elemento X que se describe posteriormente será representado por una imagen.
- La bonificación que se describe posteriormente será representada por la imagen de un reloj donde se tiene la cantidad de bonificaciones obtenidas por el jugador.
- Los aciertos y desaciertos serán indicados en texto, imágenes o ambos.
- Los premios serán representados por relojes de arena de: oro, plata y bronce.

Cálculo con figuras geométricas por tipos y colores:

El juego consiste en realizar cálculos de las cifras asignadas a figuras geométricas mostradas aleatoriamente. El recuento se realiza dependiendo de los colores C y tipo de las figuras T. La complejidad aumenta de acuerdo con la cantidad de figuras, los tipos y el número contenido en estas.

Ej. de una posible situación para:

$C = 3$, $C_1 = \text{Azul}$, $C_2 = \text{Rojo}$, $C_3 = \text{Amarillo}$.

$T = 3$, $T_1 = \text{Cuadrado}$, $T_2 = \text{Círculo}$, $T_3 = \text{Triángulo}$.

Dada la situación anterior posible preguntas serían:

- Ejercicio 1: Suma de las cantidades contenidas en los triángulos.
- Ejercicio 2: Suma de las cantidades contenidas en las figuras de color azul.
- Ejercicio 3: Suma de las cantidades contenidas en los cuadrados amarillos.

Tiempo de juego y niveles:

Inicialmente se cuenta con tiempo total $T = 1$ minuto (60s) para realizar la mayor cantidad de ejercicios E posibles.

Nivel 1:

- Como valor inicial cada situación contará con una cantidad de 9 ejercicios.
- Los objetos T serán: $T_1 =$ Cuadrado y $T_2 =$ Círculo, $T_3 =$ Triángulo.
- Los colores C serán: $C_1 =$ Rojo y $C_2 =$ Azul, $C_3 =$ Amarillo, $C_4 =$ Verde.
- Cuando el jugador logre realizar en este nivel 9 ejercicios o más se desbloquea el próximo nivel.

Nivel 2:

- Como valor inicial cada situación contará con una cantidad de 15 ejercicios.
- Los objetos T serán: $T_1 =$ Cuadrado y $T_2 =$ Círculo, $T_3 =$ Triángulo
- Los colores C serán: $C_1 =$ Rojo y $C_2 =$ Azul, $C_3 =$ Amarillo, $C_4 =$ Verde
- Cuando el jugador logre realizar en este nivel 9 ejercicios o más se desbloquea el próximo nivel.

Nivel 3:

- Como valor inicial cada situación contará con una cantidad de 21 ejercicios.
- Los objetos T serán: $T_1 =$ Cuadrado y $T_2 =$ Círculo, $T_3 =$ Triángulo.
- Los colores C serán: $C_1 =$ Rojo y $C_2 =$ Azul, $C_3 =$ Amarillo, $C_4 =$ Verde.
- Cuando el jugador logre realizar en este nivel 9 ejercicios o más se desbloquea el próximo nivel.

Bonificaciones:

Las bonificaciones serán dadas cada vez que se respondan tres preguntas de forma consecutiva en un tiempo igual o menor a 6 segundos.

Premios:

Los premios serán dados en base a la efectividad que se obtenga en cada nivel.

Anexo 3 Guión del juego Recuento de figuras geométricas y colores

Historia:

Descripción de los elementos gráficos:

- En la parte superior una barra de avance (o retroceso) o un reloj que muestre el paso del tiempo total disponible para resolver los ejercicios.
- En la parte inferior se ubicará otra barra con el campo de entrada donde se escribirá el resultado de los ejercicios.
- Cada elemento X que se describe posteriormente será representado por una imagen.
- La bonificación que se describe posteriormente será representada por la imagen de un reloj de arena, que realizará una animación desde una de las esquinas inferiores hasta la barra que indica el tiempo.
- Los aciertos y desaciertos serán indicados en texto, imágenes o ambos.
- Los premios serán representados por relojes de arena de: oro, plata y bronce. Estos de tamaños diferentes (mayor el más valioso).

Recuento de figuras geométricas y colores

El juego consiste en realizar recuentos de figuras geométricas mostradas aleatoriamente. El recuento se realiza dependiendo de los colores C y tipo de las figuras T. La complejidad aumenta de acuerdo con la cantidad de figuras y los tipos.

- Ej. de una posible situación para:
- $C = 3$, $C_1 = \text{Azul}$, $C_2 = \text{Rojo}$, $C_3 = \text{Amarillo}$.
- $T = 3$, $T_1 = \text{Cuadrado}$, $T_2 = \text{Círculo}$, $T_3 = \text{Triángulo}$.

Dada la situación anterior posible preguntas serían:

- Ejercicio 1: Cuántos triángulos.
- Ejercicio 2: Cuántas figuras de color azul.
- Ejercicio 3: Cuántos cuadrados amarillos.

Tiempo de juego y niveles:

Inicialmente se cuenta con tiempo total $T = 1$ minuto (60s) para realizar la mayor cantidad de ejercicios E posibles.

Nivel 1:

- Como valor inicial cada situación contará con una cantidad de 9 ejercicios.
- Los objetos T serán: $T_1 =$ Cuadrado y $T_2 =$ Círculo, $T_3 =$ Triángulo.
- Los colores C serán: $C_1 =$ Rojo y $C_2 =$ Azul, $C_3 =$ Amarillo, $C_4 =$ Verde.
- Cuando el jugador logre realizar en este nivel 9 ejercicios o más se desbloquea el próximo nivel.

Nivel 2:

- Como valor inicial cada situación contará con una cantidad de 15 ejercicios.
- Los objetos T serán: $T_1 =$ Cuadrado y $T_2 =$ Círculo, $T_3 =$ Triángulo.
- Los colores C serán: $C_1 =$ Rojo y $C_2 =$ Azul, $C_3 =$ Amarillo, $C_4 =$ Verde.
- Cuando el jugador logre realizar en este nivel 9 ejercicios o más se desbloquea el próximo nivel.

Nivel 3:

- Como valor inicial cada situación contará con una cantidad de 21 ejercicios.
- Los objetos T serán: $T_1 =$ Cuadrado y $T_2 =$ Círculo, $T_3 =$ Triángulo.
- Los colores C serán: $C_1 =$ Rojo y $C_2 =$ Azul, $C_3 =$ Amarillo, $C_4 =$ Verde.
- Cuando el jugador logre realizar en este nivel 9 ejercicios o más se desbloquea el próximo nivel.

Bonificaciones:

Las bonificaciones serán dadas cada vez que se respondan tres preguntas de forma consecutiva en un tiempo igual o menor a 6 segundos.

Premios:

Los premios serán dados en base a la efectividad que se obtenga en cada nivel.