



Universidad de las Ciencias Informáticas

Facultad 4

*Juegos Habitando mi isla, Rescatando el tesoro y
Siguiendo la pista para la colección de
juegos educativos MundoClick*

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

Autor:

Mileisys Chacón Aroche

Tutores:

Ing. Ana María Álvarez Valdés

Ing. Nersa Doraines Acosta Labrada

Co-Tutor:

Ing. Abel Domínguez Marrero

La Habana, junio 2013

“Año del 55 Aniversario de la Revolución”

Declaración de autoría

Declaro ser autora del trabajo titulado: “Juegos Habitando mi isla, Rescatando el tesoro, y Siguiendo la pista para la colección de juegos educativos MundoClick” y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmo la presente a los 27 días del mes de junio del año 2013.

Firma del Autor
Mileisys Chacón Aroche

Firma del Tutor
Ing. Ana María Álvarez Valdés

Firma del Tutor
Ing. Nersa Doraines Acosta Labrada

Firma del Co-tutor
Ing. Abel Domínguez Marrero

Pensamiento

“Las ciencias tienen las raíces amargas, pero muy dulces los frutos.”

Aristóteles

Agradecimientos

A mis padres, mi hermano y mis adorados sobrinos por ser la verdadera razón por la que he realizado todos mis estudios, por ser mi fuente de inspiración, pero especialmente a mi mamá y mi papá por todo lo que me han enseñado, por todo lo que han luchado contra viento y marea por mí y por mi hermano para que podamos ser las personas que hoy somos, por nunca cansarse y seguir luchando, pero sobre todo por ser el verdadero motivo por el que yo he llegado a ser quien soy. A mi segundo papá Ricardo por depositar toda la confianza en mí y por tratarme como su chiquitica. A mi novio Abel por todo su amor, dedicación y paciencia, a ti muchas gracias. A mi familia por ser cómplices de todo el desarrollo de mi vida personal y profesional. A mi otra familia Dayamí, Gilbertico, Davisito y Lola, por estar a cada segundo al pendiente de mí y tratarme como un miembro más de su familia, además de brindarme toda la confianza. A Inés por ayudarme en todo lo que podía además de acogerme como su hija hembra mayor. A mi hermano, el negro Leonardo, por ser testigo de todo con respecto a mi persona desde el primer año, por preocuparse constantemente por lo que me sucediera y por apoyarme, guiarme, alarme las orejas y aconsejarme en muchas ocasiones, además de ser parte de mi escuela de la vida que me enseñó a como ser mejor tanto personal como profesionalmente, a ti muchas gracias. A la gorda, Mary Carmen, por ser la hermana hembra que nunca tuve, por estar cuando la necesitaba, y acompañarme en los momentos buenos y malos en la Universidad, muchas gracias. A Yailin y Mislaidis por ocupar un espacio importante en mi vida y por estar siempre pendientes de los que me sucediera y estar cuando me hicieron falta, a ustedes muchas gracias. A Richar por tratar siempre de ayudarme contra mi estrés y brindarme su ayuda en todo momento y por hacerme sentir que podía contar siempre con él. A mis compañeros de estudio, especialmente a los del 8105 (la vieja guardia) por siempre mantenerse de la misma manera, alegres y al pendiente de todo, a todos muchas gracias. Al antiguo 4305 y actual 4501 que me brindó la posibilidad de conocer a excelentes personas, a todos muchas gracias. A todos mis compañeros en estos años de tristezas, alegrías y sacrificios, a los que de una forma u otra han hecho posible este momento, a los que me han apoyado y los que pensaron que no llegaría hasta el final: para todos ellos es todo esto. A las tías por ser tan especiales conmigo. A las tutoras y al tribunal por guiarme por el buen camino, a todos ustedes muchas gracias.

Dedicatoria

A mis padres Vidalia y Roberto que me dieron la vida. Ellos representan todo para mí, son ejemplo, guía e inspiración. Todos y cada uno de mis logros como estudiante han sido para corresponder a los sacrificios que han hecho para hacer mis sueños realidad. Papi y mami me he formado como una Ingeniera (igual que ustedes) y mi única expectativa es ser tan buen profesional como lo han sido ustedes a lo largo de sus años de trabajo.

A mi niño, pues gracias a él todos los recuerdos que me llevo de la universidad están matizados con nuestro amor. Estoy feliz de estar a su lado por el modo en que nos involucramos el uno en las cosas del otro y la espontaneidad con que compartimos todo.

A Ibis María Marrero Alemán, porque la vida no le permitió llegar hasta aquí, porque si la magia de este instante me diera un único deseo, pediría que volviera a nacer.

Resumen

El juego como actividad lúdica es tan antiguo como la propia existencia del hombre. La utilización y 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. Debido a la utilización de los juegos en el proceso enseñanza-aprendizaje el proyecto Multisaber-El Navegante se propuso como objetivo la construcción de una colección de juegos educativos, encaminados a niños en edades escolares entre 6 y 12 años.

Para la construcción de dicha colección es necesario determinar qué características se deben incorporar a los juegos, teniendo presente el contenido a utilizar y las peculiaridades de la audiencia hacia el que esté encaminado. Con tal fin se realiza un estudio del estado del arte y se selecciona el conjunto de herramientas y tecnologías necesarias para desarrollar la propuesta de solución, se utiliza además como guía para el proceso de desarrollo la metodología RUP. En la etapa de pruebas correspondiente a RUP, se verifica que el producto construido esté acorde a las características y requisitos especificados.

Palabras claves: enseñanza-aprendizaje, educación, habilidad, juegos educativos.

Índice

Introducción	1
Capítulo I: Fundamentación teórica	6
1. <i>Introducción</i>	6
1.1. <i>Clasificación del software educativo.....</i>	6
1.1.1. <i>Juegos educativos.....</i>	7
1.2. <i>Habilidades en el uso del teclado.....</i>	11
1.3. <i>Características de la colección MundoClick.....</i>	12
1.4. <i>Análisis de soluciones similares.....</i>	13
1.5. <i>Tecnologías y herramientas.....</i>	16
1.5.1. <i>Metodología de desarrollo.....</i>	16
1.5.2. <i>Lenguaje Unificado de Modelado (UML).....</i>	17
1.5.3. <i>Lenguajes de desarrollo.....</i>	19
1.5.4. <i>Framework de desarrollo.....</i>	22
1.6. <i>Conclusiones.....</i>	25
Capítulo II: Características del sistema	26
2. <i>Introducción</i>	26
2.1. <i>Definición de modelo conceptual</i>	26
2.1.1. <i>Descripción de los conceptos</i>	26
2.1.2. <i>Modelo conceptual.....</i>	27
2.2. <i>Descripción del sistema propuesto.....</i>	28
2.3. <i>Especificación de los requisitos del sistema</i>	31

2.3.1.	Requisitos funcionales	31
2.3.2.	Requisitos no funcionales.....	35
2.4.	Modelo de casos de uso del sistema.....	35
2.4.1.	Actores del sistema	36
2.4.2.	Diagrama de casos de uso del sistema	36
2.4.3.	Descripción de los casos de uso del sistema.....	37
2.5.	Conclusiones	48
Capítulo III: Diseño del sistema		48
3.	Introducción	49
3.1.	Patrón arquitectónico Modelo Vista Controlador	49
3.2.	Modelo de diseño.....	49
3.2.1.	Patrones de diseño.....	50
3.2.2.	Diagrama de interacción.....	52
3.2.3.	Diagrama de clases del diseño.....	54
3.3.	Estándares de codificación	56
3.4.	Conclusiones	57
Capítulo IV: Implementación y pruebas		58
4.	Introducción	58
4.1.	Modelo de implementación	58
4.1.1.	Diagrama de componentes	58
4.1.2.	Diagrama de despliegue.....	61
4.2.	Pruebas de software	62
4.2.1.	Análisis de los resultados	63

4.3. Conclusiones.....	64
Conclusiones generales	65
Recomendaciones	66
Referencias bibliográficas.....	67
Bibliografía	72
Glosario de términos.....	73
Anexos	74

Introducción

El desarrollo acelerado de las Tecnologías de la Información y las Comunicaciones (TIC), ha provocado un impacto significativo en la sociedad actual. Su carácter transformador y creador, tiene día a día una mayor presencia en todos los aspectos de la vida laboral y personal, ofreciendo un nuevo espacio de innovación en ámbitos como la industria, los servicios, la salud, la administración, el comercio y la educación.

Los sistemas educativos en el mundo se enfrentan al reto de utilizar las TIC, con el objetivo de proveer a los estudiantes herramientas y conocimientos necesarios en el siglo XXI. Actualmente la educación cubana se encuentra inmersa en un proceso de grandes transformaciones que involucra este tipo de tecnologías. Algunas de las posibilidades de interacción que estos sistemas ofrecen son la búsqueda y replanteamiento continuo de contenidos y procedimientos así como el aumento de la implicación del alumno en sus tareas y desarrollo de iniciativa de los mismos. Las mejoras que brindan las TIC al proceso de enseñanza aprendizaje obligan al escolar a tomar pequeñas decisiones, a escoger, seleccionar y a filtrar información. (1)

El *software* educativo ha permitido el aumento de la calidad en el proceso de enseñanza-aprendizaje, permitiendo de esta manera adaptabilidad y atención a las diferencias individuales ya que tiene gran interactividad por el empleo de los recursos multimedia como son los sonidos, fotografías y videos. Estos recursos multimedia son utilizados principalmente para el desarrollo de la imaginación, la creatividad y facilita la transmisión de conocimientos de cada individuo de una forma más amena, integradora, diferenciada y activa que el resto de los medios de enseñanza, constituyendo un efectivo instrumento para el desarrollo educacional del hombre. (2)

Un tipo de *software* educativo son los juegos educativos que con el paulatino crecimiento en su desarrollo, estos desempeñan un gran papel en la percepción del aprendizaje por parte de los estudiantes de todos los niveles educacionales y sistemas de formación. (3) Los juegos digitales incluyen elementos de diversión que permiten captar de forma más sencilla la atención de los educandos, facilitando de esta manera la construcción de conocimientos sólidos y permanentes partiendo de lo aprendido en ellos.

La Universidad de las Ciencias Informáticas (UCI), institución surgida al calor de la batalla de ideas, entre sus misiones asume impulsar el desarrollo de la industria del *software* en Cuba. Con el objetivo de

organizar la labor productiva en la misma fueron creados varios centros de desarrollo, entre los que se encuentra el Centro de Tecnologías para la Formación (FORTES). Este centro tiene como misión desarrollar tecnologías que permitan ofrecer servicios y productos, para brindar soluciones de formación aplicando las TIC, a todo tipo de instituciones con diferentes modelos de formación y condiciones tecnológicas.

Debido a la creciente utilización de los juegos en el proceso de enseñanza–aprendizaje el equipo de trabajo del proyecto Multisaber y El Navegante perteneciente al centro FORTES decidió desarrollar una colección de juegos educativos llamado MundoClick¹. Esta colección tiene como principal propósito ser comercial y a su vez ser capaz de albergar en él un conjunto de juegos educativos para niños de todas las edades y que permita desarrollar en ellos diferentes habilidades, además de ampliar la comercialización de productos del centro FORTES. Para el desarrollo de dicha colección los profesores del Ministerio de Educación que colaboran con el Centro, realizaron un estudio en el que determinaron que debían fomentarse en los niños de 6 a 12 años de edad habilidades relacionadas con el uso del mouse y el teclado, la memoria, la agilidad mental y el conocimiento de variadas temáticas (tales como las matemáticas, el español, las ciencias, la geografía y la historia); por lo que los juegos pertenecientes a dicha colección deben permitirle a los niños aprender de manera práctica, sencilla e interactiva, despertando el interés por el contenido y desarrollando en ellos habilidades cognitivas.

El proyecto Multisaber–El Navegante ha desarrollado juegos educativos para las colecciones La caja mágica y El Navegante para la República Bolivariana de Venezuela. Estos juegos ayudan al desarrollo de las habilidades en el uso del teclado pero no pueden ser incluidos en la nueva colección ya que están contextualizados a un país en específico. Otras de las deficiencias que existen en estos juegos es que:

- ✚ Se encuentran inmersos en las colecciones antes mencionadas lo que impide la integración con MundoClick.
- ✚ La arquitectura sobre la que fueron desarrollados los juegos, no permite que se acoplen a la nueva colección.
- ✚ No motivan al jugador a llegar hasta el final ya que carecen de una historia.

¹ MundoClick es una colección de juegos educativos para niños, cuenta con un total de 27 que fomentan y ejercitan varias habilidades.

- ✚ La complejidad de los juegos no es gradual y siempre se debe comenzar desde el principio ya que los estados o niveles no son almacenados.

Teniendo en cuenta lo anteriormente planteado el **problema de investigación** queda definido de la siguiente manera: ¿Cómo contribuir al desarrollo de habilidades en el uso del teclado en los niños de las edades entre 6 y 12 años?

Se precisó como **objeto de estudio**: Aprendizaje basado en juegos.

El **campo de acción** de la investigación está enmarcado en los Juegos educativos que desarrollan habilidades en el uso del teclado.

Se define como **objetivo general**: Desarrollar los juegos Habitando mi isla, Rescatando el tesoro y Siguiendo la pista para la colección de juegos educativos MundoClick que posibiliten desarrollar habilidades en el uso del teclado.

Idea a defender

Si se desarrollan los juegos Habitando mi isla, Rescatando el tesoro y Siguiendo la pista entonces se contará con una herramienta que permita consolidar la formación de habilidades en el uso del teclado.

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.

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: Habitando mi isla, Rescatando el tesoro y Siguiendo la pista acorde a la arquitectura de la plataforma Pointer.²
- ✚ Implementación de los juegos: Habitando mi isla, Rescatando el tesoro y Siguiendo la pista acorde a la arquitectura de la plataforma Pointer.
- ✚ Integración de los juegos: Habitando mi isla, Rescatando el tesoro y Siguiendo la pista acorde a la arquitectura de la plataforma Pointer.
- ✚ Realización de pruebas funcionales a los juegos: Habitando mi isla, Rescatando el tesoro y Siguiendo la pista acorde a la arquitectura de la plataforma Pointer.

Los métodos de investigación científica proporcionan la orientación y dirección adecuada al trabajo del investigador, ya que se convierte en el camino más corto para alcanzar los resultados esperados y condiciona los nuevos conocimientos. Existen dos tipos de métodos: los teóricos y los empíricos. Los métodos teóricos permiten abordar la realidad, estudiar la naturaleza, la sociedad y el pensamiento del objeto, con el propósito de descubrir su esencia y sus relaciones, así como comprenderlo en su desarrollo, historia y lógica. Los métodos empíricos permiten describir y explicar las características fenomenológicas del objeto y representan un nivel de investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional. (4)

Métodos teóricos

- ✚ **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 semejantes, sobre todo en la evolución de los conceptos y aplicaciones asociadas a la informática educativa, específicamente en el campo de los juegos educativos.

² Pointer es una plataforma de desarrollo orientada a la implementación y despliegue de juegos. 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.

- ✚ **Modelación:** Fue de gran importancia para la investigación puesto que se aplicó en la representación gráfica de las características y funcionalidades del *software* desarrollado.

Estructura capitular

Capítulo I: 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 II: 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 III: Diseño del sistema. Mediante los diagramas generados por la metodología de desarrollo utilizada se describe la estructura interna del sistema.

Capítulo IV: Implementación y prueba. Se describe el sistema en función de componentes de *software*, se describen los casos de pruebas por los cuales los desarrolladores verifican el producto, además se selecciona el método y la técnica de prueba que se va a utilizar.

Capítulo I: Fundamentación teórica

1. Introducción

En el presente capítulo se definen conceptos esenciales en torno al software educativo y a los juegos educativos como un tipo especial de dicho software. De los juegos educativos se precisa características, ventajas y desventajas. Se realiza un análisis de soluciones similares a los juegos que se requieren en la colección MundoClick. Se caracterizan las tecnologías y herramientas a emplear para el desarrollo de la propuesta de solución.

1.1. Clasificación del software educativo

Un software educativo es un programa didáctico que permite hacer llegar el conocimiento a una audiencia de manera más sencilla, además adapta el ritmo de trabajo y las necesidades de cada educando para lograr diversas alternativas de aprendizaje. (5) Interactúa con distintas materias de diferentes maneras y ofrecen un entorno de trabajo sensible en dependencia de las circunstancias de los alumnos. (6) Se caracterizan por ser interactivos, a partir de que se le añaden programas de entretenimientos y recursos multimedia: videos, sonidos, fotografías, ejercicios, diaporamas y juegos instructivos, que apoyan la evaluación y el diagnóstico del estudiante. (7)

El software educativo cuenta con características o aspectos que son comunes y que lo distinguen independientemente de la finalidad educativa: (8)

- ✚ Son materiales desarrollados con finalidad didáctica.
- ✚ Utilizan la computadora como medio para realizar sus actividades.
- ✚ Son interactivos, ya que dan respuestas inmediatas a las acciones de los estudiantes.
- ✚ Permiten el intercambio de información entre la computadora y el estudiante.
- ✚ Se adaptan al ritmo de trabajo de cada estudiante, mostrándole ejercicios según las habilidades demostradas.
- ✚ Son fáciles de usar, no se requiere conocimientos avanzados en informática para su uso.

Tipologías que permiten clasificarlos según el tipo de software: (9)

Software algorítmico: el aprendizaje se da por medio de la transmisión del conocimiento.

- ✚ Sistema tutoriales: transmiten los conocimientos basado en el diálogo. Se tienen en cuenta las características de los estudiantes.
- ✚ Sistemas entrenadores: el objetivo es contribuir al desarrollo de una habilidad intelectual o manual, profundizando en la aplicación y retroalimentación.
- ✚ Libros electrónicos: el propósito es presentar información a los estudiantes a partir del uso de textos, gráficos, animaciones, videos y sonidos, creando motivación y facilitándole las acciones que realizan.

Software heurístico: el diseño crea ambientes con situaciones que el estudiante debe explorar, debe crear sus propias interpretaciones y modelos de pensamiento, el alumno interactúa con situaciones que permiten crear un ambiente de aprendizaje interactivo que le permita llegar a adquirir los conocimientos de manera funcional.

- ✚ Simuladores: apoya el proceso de enseñanza-aprendizaje, simulando la vida real.
- ✚ Sistemas de expertos: el objetivo es resolver problemas que normalmente realiza un experto, dar respuestas y explicar sus razonamientos.
- ✚ Sistemas tutoriales de enseñanza: detectan errores y pueden explicar el por qué se producen. Favorecen la retroalimentación de los estudiantes.
- ✚ Juegos educativos: el propósito es crear situaciones entretenidas, sin dejar de simular la realidad.

De las anteriores clasificaciones del software educativo, en la presente investigación se hará énfasis en los juegos educativos.

1.1.1. Juegos educativos

El concepto de juego educativo involucra un producto digital que incorpora una aventura en un entorno audiovisual, donde el usuario debe resolver ciertos problemas que conforman la esencia del juego en cuestión en pro de su desarrollo. Se crea entonces una reciprocidad entre el ámbito educativo y el recreacional que fortalece aún más la accesibilidad constante del usuario al conocimiento y a su formación integral.

Los Juegos Educativos representan una herramienta pedagógica de muy alto valor ya que con ellos el usuario puede aplicar los conocimientos adquiridos en el aula o en el juego como instrumento para llegar hasta la victoria. Esto aumenta en gran medida el rendimiento del estudiante pues este se verá mucho más interesado en captar la mayor cantidad de conocimientos que le sea posible.

Los Juegos Educativos estimulan el desarrollo de habilidades en los estudiantes como la coordinación motriz y los reflejos, la capacidad observación, de concentración, la creatividad, la imaginación y la memoria. (10)

Las características que debe cumplir un juego son las siguientes: (10)

- ✚ Ser una actividad libre, comenzada y terminada a voluntad del usuario.
- ✚ Ser improductiva en generación de propiedad o riqueza, cuyo fin último es el juego en sí mismo.
- ✚ Está acotada, con límites de tiempo y de espacio.
- ✚ Está reglamentada, dentro de un orden específico propio de sí misma.
- ✚ Produce incertidumbre en el desarrollo y la finalización, dado que cada partida es distinta y depende de diversas ideas.

A las características generales sobre juegos, para que sea educativo, se debe añadir:

- ✚ Deben partir de una premisa a resolver.
- ✚ Debe tener siempre al menos una solución cierta.
- ✚ Que el usuario/jugador aprenda algo;
 - ✚ bien por introducción de nuevos conocimientos;
 - ✚ bien por fijación de conocimientos previamente adquiridos;
 - ✚ bien por ejercicio de habilidades;
 - ✚ bien por compartición de experiencias;
 - ✚ bien por descubrimiento de conceptos;
 - ✚ bien por desarrollo de creaciones. (10)

En la actualidad los juegos se utilizan para los niños como una herramienta muy beneficiosa en la educación, permitiendo de una forma diferente y divertida su desarrollo intelectual y cognitivo.

Los juegos educativos, han traído consigo múltiples ventajas, pues permiten:

- ✚ Potenciar el desarrollo de la actividad mental y física.
- ✚ Fomentar normas de relaciones sociales y morales.

- ✚ Descubrir valores y aptitudes.
- ✚ Entender diversos conceptos mediante representaciones simbólicas.
- ✚ Fomentar la interacción entre los jugadores, que expresan y confrontan sus puntos de vista y opiniones.
- ✚ Utilizar y aplicar conocimientos y habilidades en diversos contextos y situaciones.
- ✚ Ganar en confianza y vencer dificultades.

Sin embargo, los juegos educativos pueden considerarse una actividad de enseñanza-aprendizaje efectiva si:

- ✚ Requiere la participación activa de los jugadores.
- ✚ Exige el uso de diversas habilidades, destrezas y conocimientos.
- ✚ Su objetivo radica en que los niños se diviertan y aprendan simultáneamente.
- ✚ Estimula el razonamiento de los niños.
- ✚ Fomenta la comunicación.
- ✚ Es susceptible de progresar en él, de jugar cada vez mejor. (11)

La aplicación provechosa de los juegos educativos en la enseñanza no solo posibilita el desarrollo biológico, psicológico, social y espiritual del jugador como eje de la acción educativa; también promueve la diversión y el entretenimiento mientras adquieren cultura y conocimiento.

Ventajas y desventajas de los juegos educativos

A continuación se muestran las principales ventajas y desventajas que trae consigo el uso de juegos educativos:

Ventajas:

- ✚ **Motivación:** La motivación es uno de los motores del aprendizaje, ya que incita a la actividad y al pensamiento. Hace que los estudiantes dediquen más tiempo a trabajar y, por tanto, es probable que aprendan más.
- ✚ **Interacción:** Continua actividad intelectual. Los estudiantes están permanentemente activos al interactuar con el ordenador y mantienen un alto grado de implicación en el trabajo.
- ✚ **Desarrollo de la iniciativa:** Los estudiantes se ven obligados a tomar continuamente nuevas decisiones ante las respuestas del ordenador a sus acciones.

- ✚ Aprendizaje a partir de los errores: Los estudiantes pueden conocer sus errores justo en el momento en que se producen y generalmente el software les ofrece la oportunidad de ensayar nuevas respuestas o formas de actuar para superarlos.
- ✚ Los alumnos a menudo aprenden con menos tiempo.
- ✚ Permiten controlar las tareas docentes de forma individual o colectiva.
- ✚ Facilitan la autoevaluación del estudiante.
- ✚ Constituyen una nueva, atractiva, dinámica y rica fuente de conocimientos.
- ✚ Marca las posibilidades para una nueva clase más desarrolladora. (12)

Desventajas

- ✚ Adicción: El software educativo interactivo resulta motivador, pero un exceso de motivación puede provocar adicción.
- ✚ Distracción: Los alumnos a veces se dedican a jugar en vez de trabajar.
- ✚ Ansiedad: La continua interacción ante el ordenador puede provocar ansiedad en los estudiantes.
- ✚ Aprendizajes incompletos y superficiales: La libre interacción de los alumnos con estos materiales (no siempre de calidad) a menudo proporciona aprendizajes incompletos con visiones de la realidad simplista y poco profunda.
- ✚ Aislamiento: Los juegos educativos permiten al alumno aprender solo, hasta le animan a hacerlo, pero este trabajo individual, en exceso, puede acarrear problemas de sociabilidad.
- ✚ Cansancio visual y otros problemas físicos: Un exceso de tiempo trabajando ante el ordenador o malas posturas pueden provocar diversas dolencias.
- ✚ Problemas con los ordenadores: A veces los alumnos cambian la configuración o contaminan con virus los ordenadores. (12)

Considerando las desventajas que pueda implicar el uso de los juegos educativos, se imponen los beneficios que pueden aportar este tipo de software en el proceso de enseñanza–aprendizaje de los educandos.

1.2. Habilidades en el uso del teclado

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”. 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” (13). 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” (13).

Desde el punto de vista de la autora de la presente investigación el concepto aportado por (14) 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 (14).

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 en el uso del teclado.

Sheila Corral distingue que entre las habilidades en tecnologías de la información y “aptitudes para el tratamiento de la información” existen un conjunto de habilidades fundamentales como es el caso el uso del teclado (15). El uso del teclado no solamente recupera competencias que pueden eventualmente haberse construido en relación con otra tecnología, como la máquina de escribir, sino que involucra además el conocimiento de diferentes elementos que son propios de este componente de la máquina. Efectivamente, la operación del teclado permite la activación de una multiplicidad de funciones y puede incluso hacer que el mouse resulte casi prescindible para el uso de determinadas aplicaciones (como es el caso del procesador de textos). Una vez más, cuanto mayor conocimiento se tenga sobre las posibilidades del teclado y sobre el modo de explotarlo, más fluida será la relación con el dispositivo y mayor el aprovechamiento de la tecnología para los fines de producción. (16)

El uso de la computadora se ha extendido a todas las enseñanzas (primaria, secundaria, media superior y superior), uno de los principios básicos de la utilización de esta, es conocer cómo utilizar los dispositivos de entrada, en este caso particular el teclado. De aquí la importancia de que aprender a utilizar el teclado a tempranas edades y adquirir habilidades con él será de gran utilidad para descubrir nuevas oportunidades de aprender y dominar el uso correcto de la computadora.

1.3. Características 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 cuál 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. (17)

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:

- ✚ Estar en un rango de edades que comprende desde 6 hasta 12 años.
- ✚ El nivel de habilidad se encuentra dado por el conocimiento que han adquirido en este período escolar.
- ✚ Período de formación donde se adquieren los conocimientos principales para el desenvolvimiento como estudiante.
- ✚ Las características psicopedagógicas constituyen un factor fundamental en el diseño de actividades lúdicas, con el fin de consolidar el proceso de aprendizaje.

1.4. Análisis de soluciones similares

En función de facilitar la concepción de los juegos educativos se realiza un estudio de soluciones similares existentes con el objetivo de verificar si existen juegos que contengan los requisitos que se necesitan en la colección MundoClick y por lo tanto que puedan ser integrados en ella. Los sistemas analizados son los siguientes:

Imagina y crea con pipo

Pipo es una colección de juegos educativos español en CD-ROM que capta rápidamente el interés del niño debido a su presentación y creatividad en el tratamiento de los diferentes temas. Imagina y crea con Pipo va dirigida principalmente a niños de entre 3 y 10 años, sin embargo este producto puede utilizarse en otras edades, dependiendo de las necesidades y capacidades de cada sujeto. Las principales áreas y habilidades que se trabajan son: creatividad, escritura, lectura, vocabulario, orientación espacial, comprensión lectora, memoria visual y la memoria a través del sistema auditivo.

Son juegos muy estimulantes, que captan el interés del niño, ya que él tiene la posibilidad de crear su propio juego, dibujo, o ser el propio autor de un cuento o de sus aventuras. (18)

El programa está estructurado en dos partes fundamentales:

1. El Taller de Cuentos: dedicado a la lectura y creación de cuentos, donde el niño tiene una gran variedad de opciones a la hora de crear sus historias, ya que puede elegir el escenario, el texto, los personajes u objetos animados, poner un sonido o melodía. (18)

2. Los Juegos Creativos: los niños pueden jugar y divertirse con Los globos, Vistiendo a Pipo y Cuca, Resolver puzles, Colorea las escenas, El Tangram de Pipo, Veo veo, El piano, Resuelve y Colorea y Las parejas. (18)

El juego de Los globos se juega de la siguiente manera:

- ✚ Cada tortuga esconde una letra, para seleccionar las letras se debe pulsar sobre una de las que aparecen en el abecedario de la pantalla o sobre las letras del teclado.
- ✚ Cuando aciertas la tortuga desaparece y en su lugar surge la letra acertada, así hasta que completes la palabra, si esto ocurriera “Cuca” llegaría sana y salva a su destino, la Isla. Si fallas se van explotando los globos de uno en uno, tienes 8 oportunidades, en caso de que explotara el octavo globo Cuca se caería al agua.

Existen 4 niveles de dificultad:

- ✚ **Nivel 1:** Las palabras son más largas y aparece la primera letra. Existe la posibilidad de pedir ayuda pulsando el botón salvavidas.
- ✚ **Nivel 2:** Las palabras son largas, pero no aparece la primera letra. Hay ayuda.
- ✚ **Nivel 3:** Las palabras son más cortas. Sigue habiendo ayuda.
- ✚ **Nivel 4:** Las palabras son cortas y no hay ayuda o botón salvavidas

Este juego presenta diferentes dificultades como:

- ✚ Repite las palabras una vez que se falla el intento impidiendo la diversidad de vocabulario del niño.
- ✚ Una vez que se desea salir, este no brinda la posibilidad de saber el nivel de aceptación que se obtuvo con las palabras.
- ✚ No guarda el estado en que se quedó el juego para un perfil determinado.

Colección “La caja mágica”

El módulo Juegos de la colección La caja mágica permite evaluar el conocimiento adquirido por el jugador a través de juegos interactivos que se correspondan con el ciclo de enseñanza y el área de conocimiento, aportando motivación y facilitando el aprendizaje de los temas propuestos en el sistema. (19)

Uno de los software que se pueden encontrar en esta colección es el de Amemos el ambiente, el mismo presenta conocimientos relacionados con las diferentes esferas de la Tierra, los objetos, fenómenos y procesos que ocurren en ella y sus relaciones, con un enfoque pedagógico acerca del cuidado del medio ambiente y la protección de los ecosistemas. En el módulo Juego de este software existen varios juegos entre los cuales se encuentra el de las Adivinanzas que consiste en que el jugador escoge la cantidad de adivinanzas a realizar y va respondiendo una seguida de la otra. Las principales dificultades que presenta dicho juego actualmente son:

- ✚ No existe un tiempo límite para responder la adivinanza.
- ✚ No guarda el estado en el que se quedó el juego al cerrar la aplicación.
- ✚ No está definido por niveles de dificultad por lo que el funcionamiento del juego nunca varía.
- ✚ No tiene definido ningún sistema de recompensa ni de penalización.
- ✚ No cuenta con un corrector ortográfico que permita guiar al jugador si la respuesta dada está correcta pero presenta alguna falta de ortografía.
- ✚ No le permite al jugador conocer cuál es la respuesta correcta si el mismo lo ha respondido mal.

El software el Ratón y la ventana proporciona información sobre el desarrollo y evolución de los equipos de cómputo y las características del *hardware* y *software*, además de abordar temas sobre el sistema operativo Linux. En el módulo Juego de este software se encuentran los juegos llamados Descubre la imagen II y Escribiendo con el teclado.

En el juego Descubre la imagen II se comienza escogiendo el tema del cual se prefieren las imágenes. Luego aparece la pregunta y un alfabeto para escoger las letras que forman la palabra. A la par que se adivina la palabra se van mostrando fragmentos de la imagen. No obstante presenta dificultades como son:

- ✚ No guarda el estado en el que se quedó el juego al cerrar la aplicación.
- ✚ No está definido por niveles de dificultad por lo que el funcionamiento del juego es siempre el mismo.
- ✚ No tiene definido ningún sistema de recompensa ni de penalización.
- ✚ No señala las letras ya seleccionadas por el usuario/jugador.

Por otra parte se encuentra el juego de Escribiendo con el teclado que consiste en que el jugador escoge el nivel de complejidad y luego escribe el nombre de los objetos que le van apareciendo, uno seguido de otro. Las principales dificultades que presenta dicho juego son:

- ✚ No existe un tiempo límite para responder.
- ✚ No guarda el estado en el que se quedó el juego al cerrar la aplicación.
- ✚ A pesar de que da la posibilidad de escoger niveles de dificultad el funcionamiento nunca varía.
- ✚ No tiene definido ningún sistema de recompensa ni de penalización.
- ✚ No cuenta con un corrector ortográfico que permita corregir la palabra escrita por el jugador.
- ✚ No le permite al jugador conocer las respuestas correctas.

Luego de analizar estos juegos educativos se concluye que reúnen un conjunto de funcionalidades con un alto grado de creatividad además de estimular a los niños a desarrollar las habilidades necesarias para favorecer el conocimiento propio del jugador. Estos juegos a su vez presentan un conjunto de dificultades, entre las cuales se pueden encontrar que están desarrollados para un país en específico, carecen de una historia por lo que no motiva al jugador a llegar al final del juego, no cuentan con un sistema de recompensa ni de penalización, no es posible integrarlos a la colección de juegos educativos MundoClick, no guarda el estado o nivel en que se quedó el jugador por lo que se hace necesario implementarlos

nuevamente a fin de obtener mejores resultados en el desempeño intelectual de la audiencia y corregir el conjunto de deficiencias detectadas.

1.5. Tecnologías y herramientas

Las herramientas y tecnologías a utilizar para la plataforma de desarrollo Pointer fueron rigurosamente analizadas por el equipo de trabajo del proyecto Multisaber–El Navegante³ (17). 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.

1.5.1. Metodología de desarrollo

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software; (20) están divididas en dos grupos: metodologías tradicionales o robustas y metodologías ágiles o ligeras. La diferencia que existe entre ambos grupos es que las metodologías tradicionales son procesos muchos más controlados, con políticas y normas tratando de buscar la calidad del software a través del orden y la documentación, sin embargo, las metodologías ágiles son procesos menos controlados y con pocos principios, que tratan de buscar la calidad del software a partir de la comunicación inmediata y directa entre las personas que intervienen en el proceso de desarrollo.

La metodología que se utilizará para guiar el proceso de desarrollo de los juegos educativos *Habitando mi isla*, *Rescatando el tesoro* y *Siguiendo la pista* es RUP, esta es la metodología que está definida en la colección de juegos educativos *MundoClick* para la cual se implementarán dichos juegos, además de permitir obtener una completa documentación del proceso de desarrollo.

Proceso Unificado de Desarrollo (RUP)

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

³ Proyecto que desarrolla las colecciones de software educativo Multisaber-El Navegante. Centro FORTES adscrito a la Facultad 4. Universidad de las Ciencias Informáticas, La Habana.

arquitectura; Construcción, donde se desarrolla el producto y Transición, la encargada de poner el producto en manos de los usuarios. (21)

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 (21):

- ✚ 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 (22):

- ✚ 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.

1.5.2. Lenguaje Unificado de Modelado (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. (23)

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 (23):

- ✚ 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 (23):

- ✚ Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, 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. (24)

Herramienta CASE de modelado

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. (25)

Estas herramientas pueden proveer muchos beneficios en todas las etapas del proceso de desarrollo de software, algunas de ellas son (25):

- ✚ Verificar el uso de todos los elementos en el sistema diseñado.
- ✚ Automatizar el dibujo de diagramas.

- ✚ Ayudar en la documentación del sistema.
- ✚ Ayudar en la creación de relaciones en la Base de Datos.
- ✚ Generar estructuras de código.

Algunas de las ventajas de estas herramientas (25):

- ✚ 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 v8.0

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 en desarrollo, es posible integrar con distintos entornos de desarrollos integrado (IDE: Netbeans, Eclipse). En el análisis se tuvieron en cuenta las siguientes características (26):

- ✚ Soporte del ciclo de desarrollo basado en RUP.
- ✚ La navegación intuitiva entre el código y el modelo visual.
- ✚ Un ambiente modelador visual superior.
- ✚ Sofisticado diseño de diagramas.

1.5.3. Lenguajes de desarrollo

Los lenguajes de desarrollo intentan conservar una similitud con el lenguaje humano, con la finalidad de que sean más naturales a quienes los usan. Establecen un conjunto de reglas sintácticas y semánticas, las cuales rigen la estructura del programa de computación que se escribe o edita. De esta forma, permite a los desarrolladores, poder especificar de forma precisa los datos sobre los que se va a trabajar, su almacenamiento, transmisión y demás acciones a realizar bajo las distintas circunstancias consideradas.

(27) Además, permite al desarrollador comunicarse con el hardware y software existente mediante el uso de instrucciones, operadores y reglas de sintaxis.

HTML5

HTML5 (Hyper Text 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 hojas de estilo en cascada por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, hojas de estilo en cascada y JavaScript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5 (28).

Permite definir la estructura una página web a través de varios elementos como pueden ser títulos (de diferentes niveles), párrafos y listas. (29) Es un estándar cada vez más soportado por los navegadores modernos Firefox, Opera, Safari y Google Chrome (30).

Algunas reglas establecidas para HTML5 (31):

- ✚ 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 debe ser independiente del dispositivo
- ✚ El proceso de desarrollo debe ser visible para el público

CSS 3

CSS3 (Cascade Style Sheet, traducido al español como: hojas de estilo en cascada) constituyen 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 ido evolucionando, 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 (32).

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 (33).

JavaScript

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 ejecutarlos, es multiplataforma, es muy utilizado por su fiabilidad y seguridad (34).

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 es 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 (35).

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 (35).

JavaScript proporciona los medios para (36):

- ✚ 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.

Notación JSON o Notación de objetos JavaScript

JSON (JavaScript Object Notation, traducido al español Notación de Objetos 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 (37).

JSON está constituido por dos estructuras (38):

- ✚ 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.

1.5.4. Framework de desarrollo

El framework es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación, promueve buenas prácticas de desarrollo como el uso de patrones.

Permite la reutilización del código y facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. (39)

Estas estructuras simplifican el desarrollo de una aplicación y agilizan el desarrollo de sistemas web. Un framework asiste al trabajo de un programador en dos niveles: en el lado del cliente y en el lado del servidor.

jQuery v1.7

La herramienta jQuery según (40): 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.

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, efectos dinámicos incluyendo aplicaciones que hacen uso de Ajax.

NodeJS v0.8.11

NodeJS es un framework de entrada y salida basado en JavaScript que busca escribir programas de red escalables similar a los servidores web. (39) Además se puede decir que es una nueva plataforma interesante para el desarrollo de aplicaciones web, servidores de aplicaciones, cualquier tipo de servidor de red o el cliente, y la programación de propósito general. Está diseñado para la máxima escalabilidad en las aplicaciones en red a través de una ingeniosa combinación de Entrada/Salida asíncrona, JavaScript del lado del servidor, el uso inteligente de las funciones de JavaScript anónimos, y una arquitectura de ejecución solo hilo de eventos rajado.

El modelo de nodo es muy diferente de plataformas comunes de aplicación de servidor que utilizan hilos de escala. La demanda es que, debido a la arquitectura orientada a eventos, la huella de memoria es baja, el rendimiento es alto, y el modelo de programación es más simple. (41)

Bootstrap v2.3.2

Bootstrap es un marco de trabajo que combina CSS y JavaScript haciendo más fácil el proceso de diseño a los desarrolladores. Cuenta por defecto con una serie de estilos que se ven bien visualmente y que se pueden personalizar para un estilo visual propio. (42)

Los componentes que utiliza se adaptan y escalan según las resoluciones de pantalla y dispositivos para proporcionar una experiencia consistente. Se emplea para el desarrollo de los juegos ya que es compatible con todos los navegadores, utiliza CSS3, HTML5 y permite la integración de la biblioteca JQuery. Además, provee una serie de plugins y componentes que facilitarán la implementación de los juegos.

Entorno de desarrollo integrado

Un IDE (Integrated Development Environment traducido al español Ambiente Integrado de Desarrollo) es un entorno de programación, puede dedicarse en exclusiva a un único 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 (43).

Adicionalmente los IDE deben tener (44):

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

NetBeans 7.2

NetBeans es un IDE de código abierto, gratuito, sin restricciones de uso, ofrece una amplia documentación y recursos de capacitación. Es una plataforma que permite a los desarrolladores la creación rápida de diferentes tipos de aplicaciones, esta herramienta sirve para escribir, compilar, depurar y ejecutar programas. Además ofrece un rendimiento significativamente mejorado y la experiencia de codificación, con las nuevas capacidades de análisis de código. (45):

Es muy importante para la implementación de los juegos educativos de la plataforma Pointer el NetBeans, ya que presenta soporte a JQuery, además para lenguajes como JavaScript, HTML, CSS y se integra con los sistemas de control de versiones tales como Subversion. (45):

1.6. Conclusiones

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. Las herramientas definidas como parte de la arquitectura y caracterizadas en este capítulo posibilitan concluir el ciclo de desarrollo del software y dar soporte completo al mismo.

Capítulo II: Características del sistema

2. 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. Definición de 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 (46).

2.1.1. Descripción de los conceptos

A continuación se identifican y describen los principales conceptos relacionados con el contexto del problema, los cuales son utilizados en el modelo conceptual.

Jugador: El jugador es un niño el cual interactúa con el software, puede jugar y de esa forma ejercitar las habilidades con el uso del teclado.

Imagen: Imagen que debe ser identificada por el jugador.

Bono: Premio que obtiene el jugador por descubrir una imagen que traía incluida un bono.

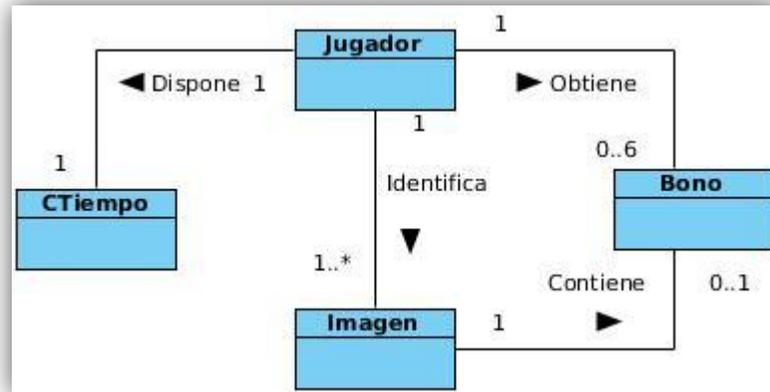
Tiempo: Intervalo de tiempo que demoró el jugador en terminar el juego.

Acertijo: Acertijo que debe ser identificado por el jugador.

Pista: Pista que debe ser descifrada por el jugador.

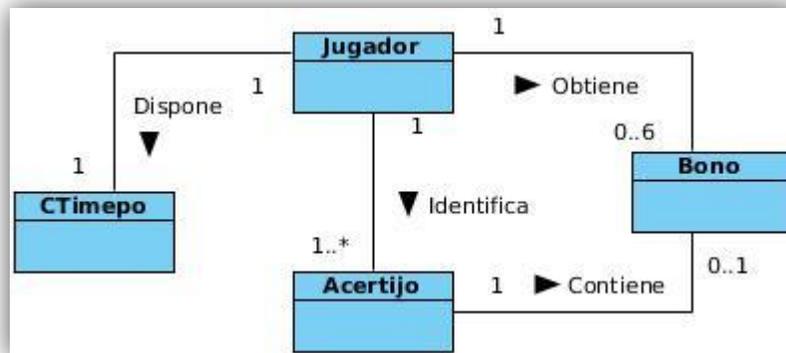
2.1.2. Modelo conceptual

Juego Habitando mi isla



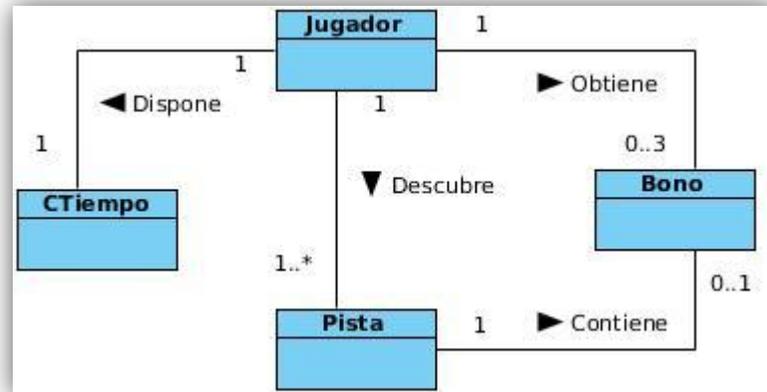
Figura# 1 Modelo conceptual del juego Habitando mi isla.

Juego Rescatando el tesoro



Figura# 2 Modelo conceptual del juego Rescatando el tesoro.

Juego Siguiendo la pista



Figura# 3 Modelo conceptual del juego Siguiendo la pista.

2.2. Descripción del sistema propuesto

El juego Habitando mi isla consiste en:

Se mostrarán imágenes y el jugador debe ir nombrando cada una de ellas. Algunas de estas imágenes contienen bonos los cuales se activarán en caso de que se nombre correctamente la imagen y en correspondencia del nivel en que se encuentre el jugador.

Niveles del juego:

Nivel 1: Se activará solo el bono "**teclado guía**", el mismo le permitirá al jugador ver en un teclado que aparecerá en la pantalla las letras que contiene el nombre de la imagen, que estarán resaltadas en otro color. El jugador debe identificar las imágenes correspondientes al nivel y se jugará en un tiempo de 8 minutos.

Nivel 2: Se activará el bono "**indicador cantidad de letras**", el mismo le permitirá al jugador saber la cantidad de letras que tiene la palabra. El jugador debe identificar las imágenes correspondientes al nivel y se jugará en un tiempo de 7 minutos.

Nivel 3: Se activará el bono "**indicador primera letra**", el mismo le permitirá al jugador saber la primera letra que tiene la palabra. El jugador debe identificar las imágenes correspondientes al nivel y se jugará en un tiempo de 6 minutos.

Nivel 4: Se activará el bono "**señalador ortográfico**", el mismo le permitirá al jugador saber en qué letra o letras de la palabra se encuentra el error. El jugador debe identificar las imágenes correspondientes al nivel y se jugará en un tiempo de 5 minutos.

Nivel 5: Se activará el bono "**congelar tiempo**", el mismo le permitirá al jugador detener la imagen por 5 segundos. El jugador debe identificar las imágenes correspondientes al nivel y se jugará en un tiempo de 4 minutos.

Nivel 6: Se activará el bono "**súper control remoto**", el mismo le permitirá al jugador saber el nombre de la imagen. El jugador debe identificar las imágenes correspondientes al nivel y se jugará en un tiempo de 3 minutos.

El juego Rescatando el tesoro consiste en:

Se mostrarán acertijos que el jugador deberá ir respondiendo. Algunos de estos acertijos contienen bonos los cuáles se activarán en caso de que se responda correctamente sin importar el nivel, los mismos pueden ser:

-  **Lupa:** resalta la primera letra de la respuesta.
-  **Súper lupa:** resalta todas las letras de la respuesta.
-  **Contador:** indica la cantidad de letras que tiene la respuesta.
-  **Lápiz mágico:** indica específicamente en que letra o letras de la palabra se encuentra el error ortográfico.
-  **Tiempo extra:** se asignan 20 segundos extras y se congela el tiempo del juego.
-  **Rompe acertijo:** muestra la respuesta del acertijo.

Niveles del juego:

Nivel 1: El jugador debe responder 7 acertijos en un tiempo de 8 minutos.

Nivel 2: El jugador debe responder 7 acertijos en un tiempo de 7 minutos.

Nivel 3: El jugador debe responder 7 acertijos en un tiempo de 6 minutos.

Nivel 4: El jugador debe responder 7 acertijos en un tiempo de 5 minutos.

Nivel 5: El jugador debe responder 7 acertijos en un tiempo de 4 minutos.

Nivel 6: El jugador debe responder 7 acertijos en un tiempo de 3 minutos.

El juego Siguiendo la pista consiste en:

Se mostrarán pistas que el jugador debe descifrar. Algunas de estas pistas contienen bonos los cuáles se activarán en caso de que se descifren correctamente sin importar el nivel.

Los mismos pueden ser:

-  **Agregar intento:** sirve para agregar otro intento cuando se está descubriendo una pista.
-  **Lupa:** muestra la imagen que se está descubriendo cuando se pasa el mouse por encima.
-  **Pista:** proporciona parte de la frase que se está descifrando y en conjunto descubre un fragmento de la imagen.

Niveles del juego:

Nivel 1: El jugador debe descubrir las pistas correspondientes en el nivel en un tiempo de 10 minutos y al finalizar el mismo se le asignará la distinción de Investigador.

Nivel 2: El jugador debe descubrir las pistas correspondientes en el nivel en un tiempo de 9 minutos y al finalizar el mismo se le asignará la distinción de Detective.

Nivel 3: El jugador debe descubrir las pistas correspondientes en el nivel en un tiempo de 8 minutos y al finalizar el mismo se le asignará la distinción de Capitán.

Nivel 4: El jugador debe descubrir las pistas correspondientes en el nivel en un tiempo de 7 minutos y al finalizar el mismo se le asignará la distinción de Mayor.

Nivel 5: El jugador debe descubrir las pistas correspondientes en el nivel en un tiempo de 6 minutos y al finalizar el mismo se le asignará la distinción de Teniente.

Nivel 6: El jugador debe descubrir las pistas correspondientes en el nivel en un tiempo de 5 minutos y al finalizar el mismo se le asignará la distinción de Coronel.

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. Especificación de los requisitos del sistema

Para que un proceso de software sea exitoso es imprescindible una descripción detallada, correcta y exhaustiva de los requisitos, ya que es necesario saber qué debe hacer el sistema. Los requisitos expresan lo que los usuarios quieren del producto, estos son especificados por escrito como parte de un documento formal. Los requisitos (o requerimientos) son condiciones que deben estar presentes en el *software*, estos se clasifican en: funcionales y no funcionales. Los requisitos funcionales definen las acciones que el sistema será capaz de realizar, son capacidades o condiciones que debe cumplir para satisfacer las necesidades funcionales del cliente. Los requisitos no funcionales describen aquellas propiedades o cualidades del sistema una vez que esté finalizado, como por ejemplo, el rendimiento, interfaces de usuario, fiabilidad, mantenimiento, seguridad, portabilidad y estándares. (47)

A continuación se muestran los requisitos funcionales y no funcionales de los juegos *Habitando mi isla*, *Siguiendo la pista* y *Rescatando el tesoro* para la colección de juegos educativos MundoClick.

2.3.1. Requisitos funcionales

Juego Habitando mi isla

RF 1: Mostrar imagen a identificar.

El sistema deberá mostrar imágenes cayendo y el jugador deberá escribir el nombre de ellas antes de que lleguen a tierra.

RF 2: Introducir respuesta.

Se debe permitir que el jugador introduzca a través del teclado el nombre de la imagen mostrada.

RF 3: Verificar respuesta.

Una vez introducido el nombre de la imagen el sistema procederá a verificar si el valor es correcto o no.

RF 4: Mostrar pantalla de verificación de respuesta.

Cuando se verifica la respuesta del jugador se mostrará "Correcto" o "Incorrecto" según corresponda.

RF 5: Asignar bono.

Cuando el jugador identifica una imagen que trae en ella un bono incluido el mismo se activará y el jugador podrá hacer uso de él.

RF 6: Calcular efectividad.

RF 7: Mostrar el tiempo consumido por el jugador.

El jugador podrá observar el tiempo transcurrido en el juego.

RF 8: Mostrar los resultados del juego.

El jugador conocerá al finalizar el juego la efectividad y la puntuación que obtuvo en el desarrollo del mismo.

RF 9: Guardar estado del juego.

Cuando el jugador decide salir del juego se guarda el nivel, el nombre, la posición y la puntuación que tenía, para cuando vuelva a acceder siga en el mismo lugar.

RF 10: Cambiar nivel.

Cuando el jugador identifique las imágenes correspondientes al nivel debe cambiar para el próximo.

RF 11: Múltiples jugadores en el mismo juego.

Permite a varios usuarios jugar en el mismo juego.

RF 12: Terminar juego.

Permite al jugador terminar y salir del juego.

Juego Rescatando el tesoro

RF 1: Mostrar acertijo a descubrir.

El sistema deberá mostrar los acertijos.

RF 2: Introducir respuesta.

Se debe permitir que el jugador introduzca a través del teclado la respuesta al acertijo mostrado.

RF 3: Verificar respuesta.

Una vez introducida la respuesta al acertijo el sistema procederá a verificar si el valor es correcto o no.

RF 4: Mostrar pantalla de verificación de respuesta.

Cuando se verifica la respuesta del jugador se mostrará "Correcto" o "Incorrecto" según corresponda.

RF 5: Asignar bono.

Cuando el jugador identifica un acertijo que trae un bono incluido, el mismo se activará y el jugador podrá hacer uso de él.

RF 6: Calcular efectividad.

RF 7: Mostrar el tiempo consumido por el jugador.

El jugador podrá observar el tiempo transcurrido en el juego.

RF 8: Mostrar los resultados del juego.

El jugador conocerá al finalizar el juego la efectividad y la puntuación que obtuvo en el desarrollo del mismo.

RF 9: Guardar estado del juego.

Cuando el jugador decide salir del juego se guarda el nivel, el nombre, la posición y la puntuación que tenía, para cuando vuelva a acceder siga en el mismo lugar.

RF 10: Cambiar nivel.

Pasará al siguiente nivel cuando haya respondido todos los acertijos correspondientes al nivel.

RF 11: Múltiples jugadores en el mismo juego.

Permite a varios usuarios jugar en el mismo juego.

RF 12: Terminar juego.

Permite al jugador terminar y salir del juego.

Juego Siguiendo la pista

RF 1: Mostrar pista.

El sistema deberá mostrar una imagen oculta y el jugador deberá descifrar el nombre de la misma antes de que se le acabe el tiempo y los intentos que tiene.

RF 2: Introducir respuesta.

Se debe permitir que el jugador introduzca la respuesta a través del teclado.

RF 3: Verificar respuesta.

Una vez que el jugador va escribiendo cada letra el sistema deberá revisar si es correcto o no.

RF 4: Mostrar pantalla de verificación de respuesta.

Cuando se verifica la respuesta del jugador y es correcto se descubrirá un fragmento de imagen y en caso contrario se le restará oportunidades al jugador.

RF 5: Asignar bono.

Cuando el jugador descifra una pista que trae un bono incluido, el mismo se activará y el jugador podrá hacer uso de él.

RF 6: Calcular efectividad.

RF 7: Mostrar el tiempo consumido por el jugador.

El jugador podrá observar el tiempo en el transcurso del juego.

RF 8: Mostrar los resultados del juego.

El jugador conocerá al finalizar el juego la efectividad y la puntuación que obtuvo en el desarrollo del mismo.

RF 9: Guardar estado del juego.

Cuando el jugador decide salir del juego se guarda el nivel, el nombre, la posición y la puntuación que tenía, para cuando vuelva a acceder siga en el mismo lugar.

RF 10: Cambiar nivel.

Cuando el jugador descubra varias pistas correspondientes a cada nivel el sistema debe cambiar automáticamente para el próximo y asignarle al jugador una distinción.

RF 11: Múltiples jugadores en el mismo juego.

Permite a varios usuarios jugar en el mismo juego.

RF 12: Terminar juego.

Permite al jugador terminar y salir del juego.

2.3.2. Requisitos no funcionales

RNF 1 Requisito de portabilidad

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

RNF 2 Requisito de hardware

- ✚ Procesador Pentium 233 MHz (recomendado 500 MHz o mayor).
- ✚ Como mínimo 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.

RNF 3 Requisito de interfaz

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

RNF 4 Requisitos de usabilidad

El software será usado por los niños en las edades de 6 a 12 años.

RNF 5 Restricciones de diseño e implementación

- ✚ Lenguaje de desarrollo: JavaScript
- ✚ Lenguaje de marcado: HTML 5
- ✚ Hoja de estilo: CSS 3
- ✚ Framework de desarrollo: JQuery v1.7 y NodeJS v0.8.11
- ✚ Entorno de Desarrollo Integrado: Netbeans v7.2
- ✚ Herramienta CASE: Visual Paradigm v8.0

2.4. Modelo de casos de uso del sistema

El modelo de casos de uso del sistema representa gráficamente la interacción de los casos de uso con los actores del sistema. Este modelo le permite a los desarrolladores y al cliente llegar a un acuerdo sobre los requisitos que debe presentar el producto. Los casos de uso del sistema son un conjunto de secuencias de acciones que un sistema ejecuta y que produce un resultado observable para un actor. (48)

2.4.1. Actores del sistema

Un actor es una persona, sistema o dispositivo que interactúa con el sistema, iniciando, recibiendo los resultados o participando en alguna de las acciones de un caso de uso. Representa un rol. (48)

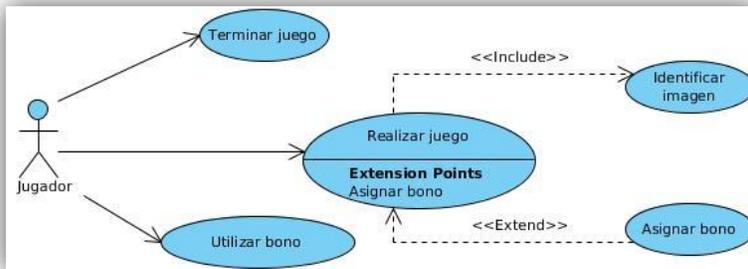
Tabla 1: Actores del sistema.

Actor	Descripción
Jugador	El jugador es un niño el cual interactúa con el software, puede jugar y de esa forma ejercitar las habilidades con el uso del teclado.

2.4.2. Diagrama de casos de uso del sistema

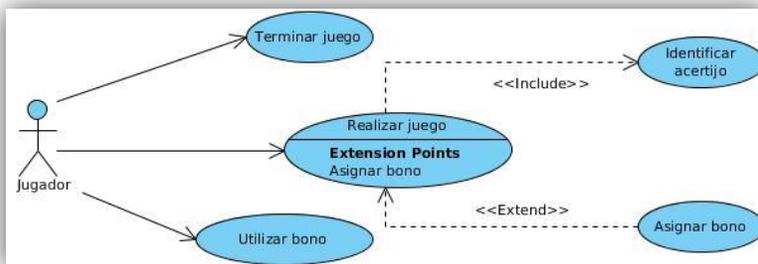
A continuación se presenta el diagrama de casos de uso del sistema, que muestra la relación entre los actores y los casos de uso, quedando representadas gráficamente las funcionalidades del sistema y su interacción con los usuarios.

Juego Habitando mi isla



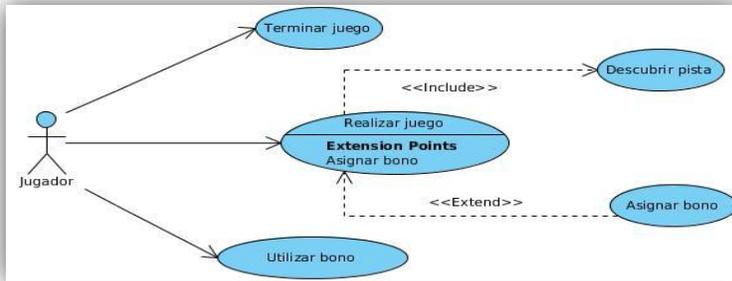
Figura# 4 Diagrama de casos de uso del juego Habitando mi isla.

Juego Rescatando el tesoro



Figura# 5 Diagrama de casos de uso del juego Rescatando el tesoro.

Juego Siguiendo la pista



Figura# 6 Diagrama de casos de uso del juego Siguiendo la pista.

2.4.3. Descripción de los casos de uso del sistema

En el siguiente epígrafe se realiza una descripción de los casos de uso presentes en el diagrama de casos de uso. La descripción de un caso de uso detalla una secuencia de interacciones usuario-sistema, incluyendo cómo comienza y cómo termina. Estas descripciones tienen como objetivo mostrarle al cliente un acercamiento de la propuesta de sistema y orientar a los desarrolladores durante la implementación. A continuación se muestran las descripciones de los caso de usos (CU) arquitectónicamente significativos de la propuesta de los juegos Habitando mi isla, rescatando el tesoro y Siguiendo la pista para la colección de juegos educativos MundoClick.

Debido a que los juegos tienen características similares, se presentará en el capítulo solamente la descripción de los CU del juego Habitando mi isla y Siguiendo la pista, el resto pueden encontrarse en los Anexos [4](#), [5](#), [6](#), [7](#) y [8](#).

Juego Habitando mi isla

Tabla 2: Descripción del caso de uso Realizar juego.

Caso 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 o termina el juego.
Prioridad	Crítico
Precondiciones	-

Postcondiciones	-	
Flujo normal de eventos		
	Actor	Sistema
1.	Selecciona la opción Jugar del menú principal.	
2.		Inicia la secuencia lógica del juego según lo especificado en el guion del mismo. (Ver Anexo 3)
3.		Inicia la secuencia de imágenes que debe nombrar el jugador (Ver CU Identificar imagen).
4.		Muestra al jugador el momento en que pasa o sube de nivel.
5.		Muestra cuando un jugador obtiene un bono: (Ver CU Asignar bono).
6.		<p>Cuando se termina el juego muestra los resultados generales del mismo:</p> <ul style="list-style-type: none"> - Nombre del jugador - Puntos - Efectividad
Relaciones	CU Incluidos	Identificar imagen
	CU Extendidos	Asignar bono

Tabla 3: Descripción del caso de uso Identificar imagen.

Casos de uso	Identificar imagen
Actores	Jugador
Resumen	El caso de uso inicia cuando el jugador selecciona la opción jugar y termina cuando haya identificado la imagen que muestra el sistema.
Prioridad	Crítico

Precondiciones	Que se haya ejecutado el caso de uso Realizar juego.	
Postcondiciones		
Flujo de eventos		
Flujo básico Identificar imagen		
	Actor	Sistema
1.		Muestra la secuencia de imágenes.
2.	Introduce el nombre de la imagen mostrada.	
3.		Verifica respuesta dada por el jugador.
4.		Muestra resultado de la verificación Correcto o Incorrecto.
5.		Suma 3 puntos.
6.		Muestra la imagen siguiente y se comenzará a ejecutar nuevamente el flujo básico de eventos a partir del paso 1.
7.		Termina el caso de uso.
Flujos alternos		
4a. Los datos son incorrectos.		
	Actor	Sistema
1.		Resta 1 punto.
2.		Se ejecuta el paso 3 del flujo normal de eventos.

Tabla 4: Descripción del caso de uso Asignar bono.

Casos de uso	Asignar bono
Actores	Jugador
Resumen	El caso de uso inicia cuando se haya ejecutado el caso de uso Realizar

	juego.	
Prioridad	Crítico	
Precondiciones	Que se haya ejecutado el caso de uso Realizar juego.	
Postcondiciones		
Flujo de eventos		
Flujo básico Asignar bono		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor identifica la imagen que trae un bono.	
2.		<p>Adiciona el bono correspondiente al nivel si el jugador identificó la imagen que lo traía incluido.</p> <p>Nivel 1: Teclado guía.</p> <p>Nivel 2: Indicador cantidad de letras.</p> <p>Nivel 3: Indicador primera letra.</p> <p>Nivel 4: Señalador ortográfico.</p> <p>Nivel 5: Congelar tiempo.</p> <p>Nivel 6: Súper control remoto.</p>
3.		Se activa la tecla correspondiente al bono.
4.		Permite que el jugador continúe identificando las imágenes que van saliendo.
5.		Termina el caso de uso.
Relaciones	CU Incluidos	

	CU Extendidos	Asignar bono en el CU Realizar juego
--	----------------------	--------------------------------------

Tabla 5: Descripción del caso de uso Utilizar bono.

Casos de uso	Utilizar bono	
Actores	Jugador	
Resumen	El caso de uso inicia cuando se haya ejecutado el caso de uso Asignar bono.	
Prioridad	Crítico	
Precondiciones	Que se haya ejecutado el caso de uso Asignar bono.	
Postcondiciones		
Flujo de eventos		
Flujo básico Utilizar bono		
	Actor	Sistema
1.	Selecciona la tecla correspondiente a uno de los bonos asignados.	
2.		<p>Realiza una acción determinada en dependencia del bono seleccionado:</p> <p>Bono:</p> <ul style="list-style-type: none"> - Teclado guía: En el teclado que estará en pantalla se resaltarán las letras que deben ser presionadas. - Indicador cantidad de letras: Indica la cantidad de letras que tiene el nombre de la imagen. - Indicador primera letra: Indica la primera letra que tiene el nombre de la imagen.

		<ul style="list-style-type: none"> - Señalador ortográfico: Indica en que letra o letras de la palabra está el error ortográfico. - Congelar tiempo: Congela el tiempo de caída de la imagen por 5 segundos. - Súper control remoto: Brinda el nombre de la imagen.
3.		Termina el caso de uso.

Tabla 6: Descripción del caso de uso Terminar juego.

Casos de uso	Terminar juego	
Actores	Jugador	
Resumen	El caso de uso inicia cuando el jugador selecciona la opción Salir y termina cuando se cierra el juego.	
Prioridad	Crítico	
Precondiciones		
Postcondiciones	Se termina el juego.	
Flujo de eventos		
Flujo básico Terminar juego		
	Actor	Sistema
1.	Selecciona la opción Salir.	
2.		Muestra un mensaje de confirmación: - ¿Desea realmente salir del juego?
3.		Guarda los resultados del jugador.
4.		Termina el caso de uso.
Flujos alternos		

2a. Cancelar salir del juego.		
	Actor	Sistema
1.	Selecciona la opción cancelar.	
2.		Sigue la secuencia del juego.

Juego Siguiendo la pista

Tabla 7: 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 o termina el juego.	
Prioridad	Crítico	
Precondiciones		
Postcondiciones		
Flujo de eventos		
Flujo básico Realizar juego		
	Actor	Sistema
1.	Selecciona la opción Jugar del menú principal.	
2.		Inicia el juego según lo especificado en el guion del mismo. (Ver Anexo 2)
3.		Inicia la secuencia de pistas que debe ir descubriendo el jugador (Ver caso de uso Descubrir pista).
4.		Muestra al jugador el momento en que pasa o sube de nivel.
5.		Muestra cuando un jugador obtiene un bono: (Ver caso de uso Asignar bono).

6.		Cuando se termina el juego muestra los resultados del mismo: <ul style="list-style-type: none"> - Puntos - Usuario
Relaciones	CU Incluidos	Descubrir pista
	CU Extendidos	Asignar bono

Tabla 8: Descripción del caso de uso Descubrir pista.

Casos de uso	Descubrir pista	
Actores	Jugador	
Resumen	El caso de uso inicia cuando el jugador selecciona la opción jugar y termina cuando haya descubierto la pista que muestra el sistema.	
Prioridad	Crítico	
Precondiciones		
Postcondiciones	Se muestra cuando la pista ha sido descubierta exitosamente.	
Flujo de eventos		
Flujo básico Descubrir pista		
	Actor	Sistema
1.		Muestra pista a descubrir.
2.	Debe descifrar la pista.	
3.		Verifica respuesta dada por el jugador.
4.		Muestra resultado de la verificación Correcto.
5.		Suma 3 puntos.
6.		Muestra próxima pista y se comenzará a ejecutar nuevamente el flujo básico de eventos a partir del paso 1.

Flujos alternos		
4a. Los datos son incorrectos.		
	Actor	Sistema
1.		Resta 1 intento.
2.		Se ejecuta el paso 3 del flujo normal de eventos.

Tabla 9: Descripción del caso de uso Asignar bono.

Casos de uso	Asignar bono	
Actores	Jugador	
Resumen	El caso de uso inicia cuando se haya ejecutado el caso de uso Realizar juego.	
Prioridad	Crítico	
Precondiciones	Que se haya ejecutado el caso de uso Realizar juego.	
Postcondiciones		
Flujo de eventos		
Flujo básico Asignar bono		
	Actor	Sistema
1.	El caso de uso se inicia cuando el jugador descubre la pista que trae incluida un bono.	
2.		Adiciona el bono si el jugador descubrió la pista que lo traía incluido. <ul style="list-style-type: none"> - Agregar intento - Lupa - Pista
3.		Permite que el jugador continúe identificando las próximas pistas.

4.		Se activarán con la utilización del mouse.
Relaciones	CU Incluidos	
	CU Extendidos	Asignar bono en el caso de uso Realizar juego.
Casos de uso	Terminar juego	
Actores	Jugador	
Resumen	El caso de uso inicia cuando el jugador selecciona la opción Salir y termina cuando se cierra el juego.	
Prioridad	Crítico	
Precondiciones		
Postcondiciones	Se termina el juego.	
Flujo de eventos		
Flujo básico Terminar juego		
	Actor	Sistema
5.	Selecciona la opción Salir.	
6.		Muestra un mensaje de confirmación: - ¿Desea realmente salir del juego?
7.		Guarda los resultados del jugador.
8.		Termina el caso de uso.
Flujos alternos		
2a. Cancelar salir del juego.		
	Actor	Sistema
3.	Selecciona la opción cancelar.	
4.		Sigue la secuencia del juego.

Tabla 10: Descripción del caso de uso Utilizar bono.

Casos de uso	Utilizar bono
---------------------	---------------

Actores	Jugador	
Resumen	El caso de uso inicia cuando se haya ejecutado el caso de uso Asignar bono.	
Prioridad	Crítico	
Precondiciones	Que se haya ejecutado el caso de uso Asignar bono.	
Postcondiciones		
Flujo de eventos		
Flujo básico Utilizar bono		
	Actor	Sistema
1.	Selecciona con el mouse uno de los bonos asignados.	
2.		Realiza una acción determinada en dependencia del bono seleccionado: Bono: - Agregar intento: Agrega otro intento cuando se está descubriendo una pista. - Lupa: Muestra la imagen que se está descubriendo al pasar el mouse por arriba. - Pista: Resalta alguna de la o las letras que contiene la frase.
3.		Termina el caso de uso.

Tabla 11: Descripción del caso de uso Utilizar bono.

Casos de uso	Terminar juego
Actores	Jugador
Resumen	El caso de uso inicia cuando el jugador selecciona la opción Salir y

	termina cuando se cierra el juego.	
Prioridad	Crítico	
Precondiciones		
Postcondiciones	Se termina el juego.	
Flujo de eventos		
Flujo básico Terminar juego		
	Actor	Sistema
9.	Selecciona la opción Salir.	
10.		Muestra un mensaje de confirmación: - ¿Desea realmente salir del juego?
11.		Guarda los resultados del jugador.
12.		Termina el caso de uso.
Flujos alternos		
2a. Cancelar salir del juego.		
	Actor	Sistema
5.	Selecciona la opción cancelar.	
6.		Sigue la secuencia del juego.

2.5. Conclusiones

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 logró un mejor entendimiento de las funcionalidades definidas para la propuesta de solución realizada.

Capítulo III: Diseño del sistema

3. 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, restricciones relacionadas con los lenguajes de programación y 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

Los patrones arquitectónicos son patrones de software encargados de definir la estructura de un sistema, estos a su vez se componen de subsistemas con sus responsabilidades. Cuentan con varias directivas para organizar los componentes del propio sistema, con el objetivo de facilitar el diseño del producto. (49)

Para la implementación de los juegos se emplea el patrón Modelo-Vista-Controlador (MVC), enfoque que separa la lógica de la aplicación de la presentación: (49)

- ✚ El modelo representa las estructuras de datos que brindan información al usuario.
- ✚ La vista cuenta con la información que se presenta al usuario, en el caso de una aplicación web la vista es una página HTML sobre la que el usuario puede realizar varias operaciones.
- ✚ El controlador se encarga de manejar y responder las solicitudes del usuario, de esta forma procesa la información y realiza los cambios necesarios en el modelo.

En el caso de los juegos, este patrón queda reflejado en la organización de cada juego, a la hora de guardar el estado y obtener la información que debe presentarse a los jugadores. La vista a través de una función Ajax le envía la petición a la clase controladora que accede a los datos que se encuentran en el modelo y le da la respuesta a la vista.

3.2. Modelo de diseño

El modelo de diseño es una representación no genérica específica del plano de implementación. Se realiza teniendo en cuenta los requisitos tanto funcionales como no funcionales del sistema con el objetivo de adquirir una comprensión en profundidad de los aspectos relacionados con estos y los lenguajes de

programación, componentes reutilizables y tecnologías de interfaz de usuario (47). Es una disciplina clave dentro de la ingeniería de software a diferencia del análisis, que en muchos proyectos posee carácter opcional, siendo el centro de atención al final de la fase de elaboración e inicio de la fase de construcción. De manera general, tiene como propósito transformar los requisitos tanto funcionales como no funcionales en un diseño de clases que describa las relaciones entre estas, en un lenguaje técnico lo más cercano posible al lenguaje de programación.

3.2.1. 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 (50).

Patrones GRASP⁴

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 (50). 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 la información, la clase que tiene la información necesaria para realizar la responsabilidad” (50).

Se conserva el encapsulamiento de los datos, 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 responsabilidades a las clases Imagen, Acertijos y Pistas, en estas clases es donde se maneja únicamente toda la información referente a una imagen, un acertijo o a una pista, en ellas se conforma la información de cada tipo en correspondencia del nivel. De igual forma es observable en la clase Usuario, la cual contiene las responsabilidades y operaciones que competen al usuario así como sus características.

⁴Patrones de diseño de asignación de responsabilidades

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” (50).

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 Juego crea objetos de la clase Usuario, Nivel y C_Tiempo con el fin de poder acceder a algunos de sus eventos, tal y como se muestra a continuación:

```
juego = new Juego();
nivel = new Nivel(nivel);
reloj = new C_Tiempo('tiempoReloj','desc',function({});
usuario = new Jugador(parametro1,parametro2,parametro3);
```

Controlador: Es el responsable de controlar o manejar los eventos del sistema (50).

En el diseño de la propuesta de solución el uso de este patrón está asociado al manejo de los eventos referentes a las distintas funciones generales del juego, en este caso las clases ET, ADV y DI2 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 responsabilidad de manera que exista bajo acoplamiento” (50).

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 se hace de forma secuencial, es decir la Clase Nivel accede a la información de la clase Imagen, Acertijos y Pistas a través de un objeto de estas clases, esto permite que la clase Nivel 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 anteriormente.

Alta cohesión: “Asignar una responsabilidad de modo que la cohesión siga siendo alta” (50).

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 especificado en el patrón anteriormente explicado.

3.2.2. Diagrama de interacción

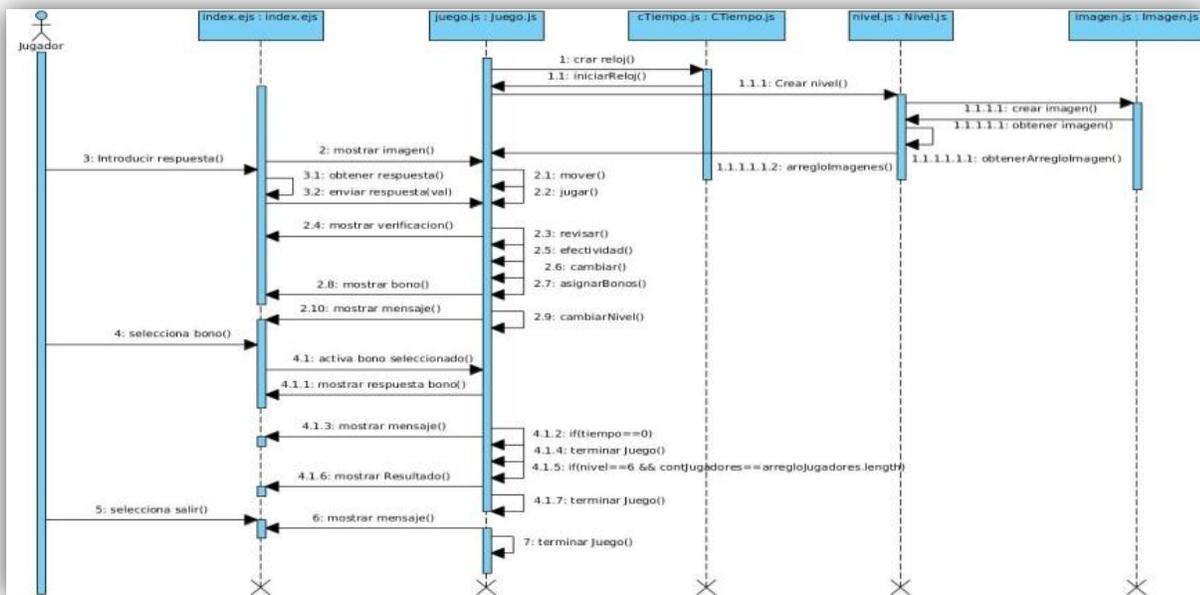
El diagrama de interacción describe secuencias de intercambios de mensajes entre los roles que implementan el comportamiento de un sistema, este diagrama permite al diseñador identificar las clases de software que intervienen en la solución, así como los métodos de las clases (51).

Este se puede visualizar a través de dos diagramas (secuencia y colaboración), ambos sirven para expresar interacciones semejantes o idénticas de mensaje (51). En la investigación se utilizarán los diagramas de secuencia para explicar el paso de mensajes entre las distintas clases que coexisten en el diagrama de diseño.

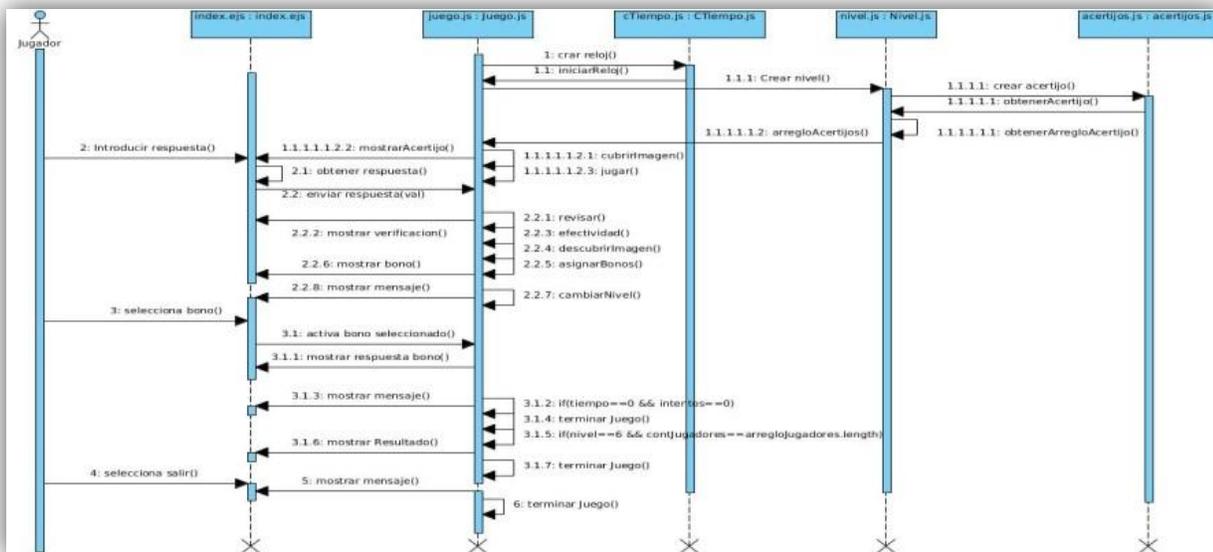
Diagramas de secuencia

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. (51)

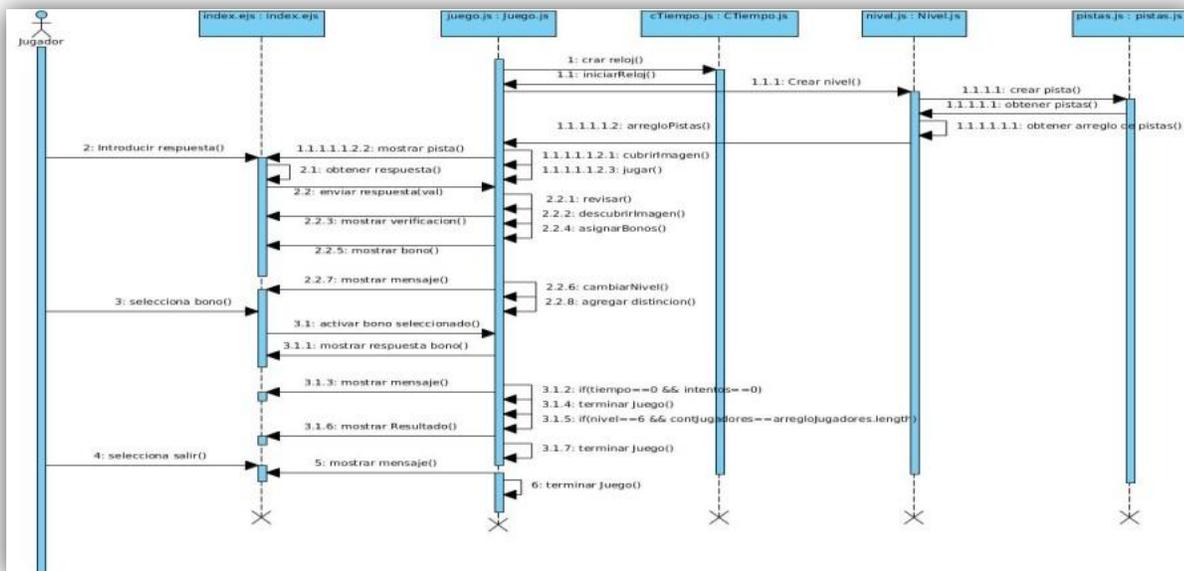
A continuación se muestra el diagrama de secuencia de los juegos ***Habitando mi isla, Rescatando el tesoro*** y ***Siguiendo la pista***.



Figura# 7 Diagrama de secuencia del juego Habitando mi isla.



Figura# 8 Diagrama de secuencia del juego Rescatando el tesoro.



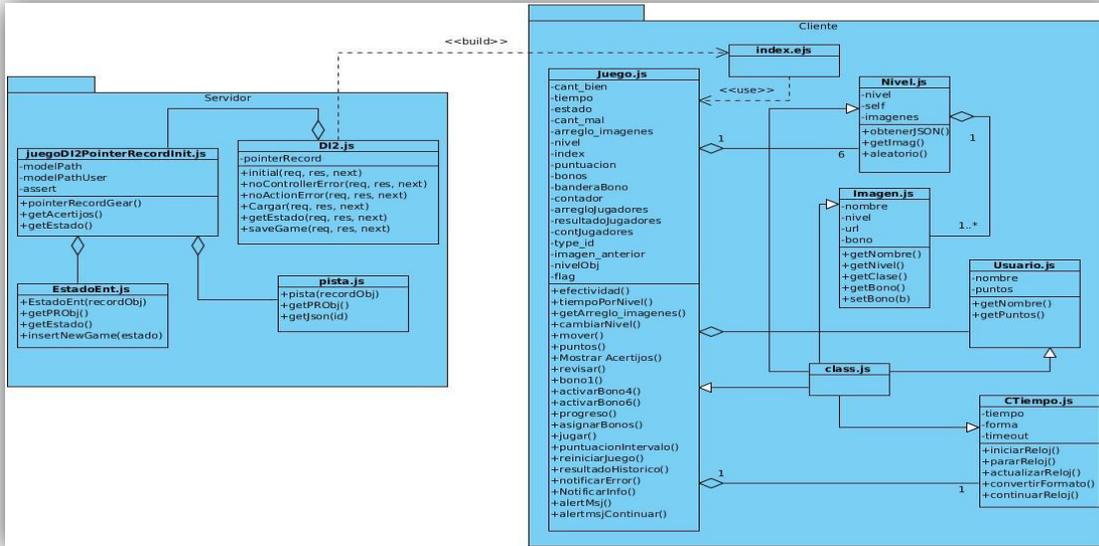
Figura# 9 Diagrama de secuencia del juego Siguiendo la pista.

3.2.3. Diagrama de clases del diseño

El diagrama de clases del diseño de un sistema 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 (52).

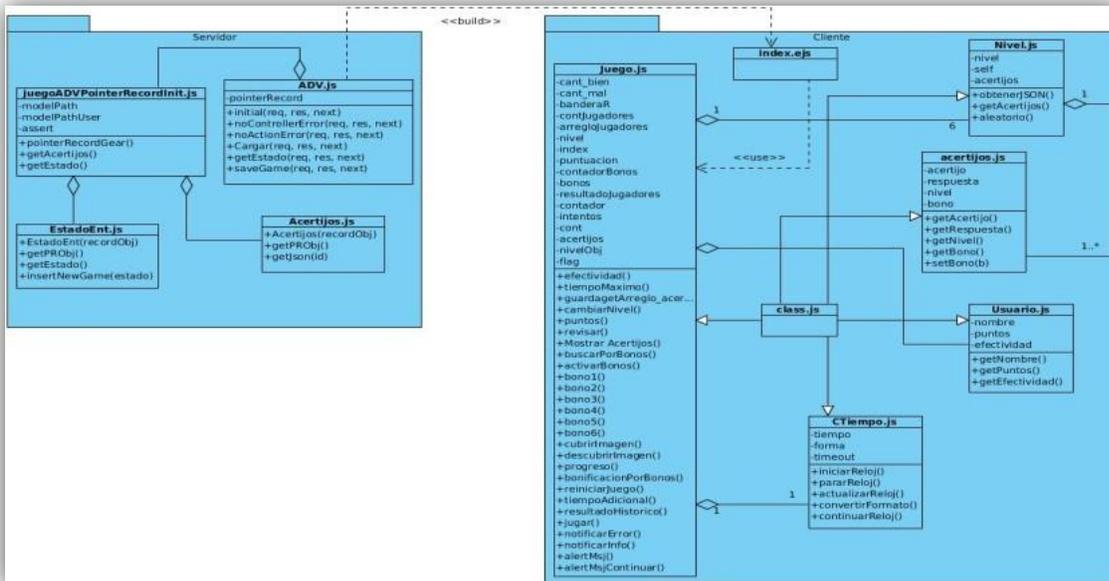
A continuación se muestran los diagramas de clase del diseño para los juegos Habitando mi isla, Rescatando el tesoro y Siguiendo la pista.

Diagrama de clase del diseño del juego *Habitando mi isla*.



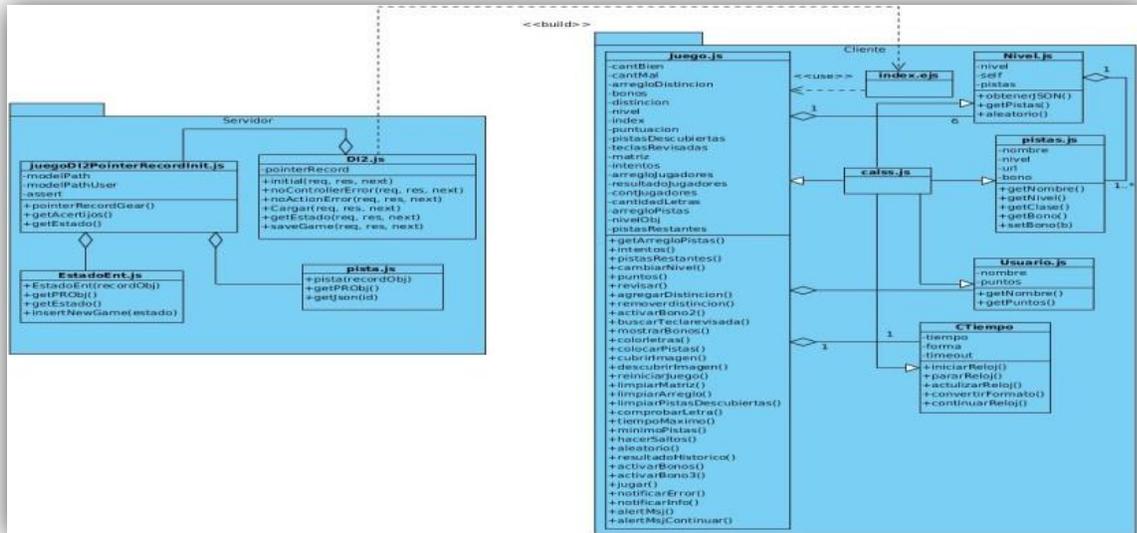
Figura# 10 Diagrama de clase del diseño del juego *Habitando mi isla*.

Diagrama de clase del diseño del juego *Rescatando el tesoro*.



Figura# 11 Diagrama de clase del diseño del juego *Rescatando el tesoro*.

Diagrama de clase del diseño del juego Siguiendo la pista.



Figura# 12 Diagrama de clase del diseño del juego Siguiendo la pista.

3.3. Estándares 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. (53)

A continuación se establecen las normas de codificación que serán utilizadas en el desarrollo de los juegos: **Habitando mi isla, Rescatando el tesoro y Siguiendo la pista**, las cuales 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 (54).

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 Nivel = 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.

```
mostrarAcertijos: function ()
```

```
asignarBonos: function ()
```

```
cambiarNivel: function ()
```

```
tiempoMaximo: function ()
```

3.4. Conclusiones

En este capítulo se realizó el diseño de los casos de uso del sistema donde se relacionan los requisitos funcionales y los no funcionales, llevándolos a una expresión de la programación a partir de la transformación de los requerimientos en un diagrama de clases del diseño. Se estructuró cómo debe quedar el sistema haciendo uso de los patrones de diseño definidos para el mismo, contribuyendo a que dicho diseño se corresponda con el entorno de la implementación.

Capítulo IV: Implementación y pruebas

4. Introducción

Los artefactos generados en la etapa de diseño son la principal entrada en el flujo de trabajo implementación. En esta etapa se describe el software en términos de componentes (ficheros de código fuente y binario, scripts, ejecutables y similares). En el presente capítulo se describen cada uno de dichos componentes y la estrategia de prueba utilizada para verificar el resultado de la implementación.

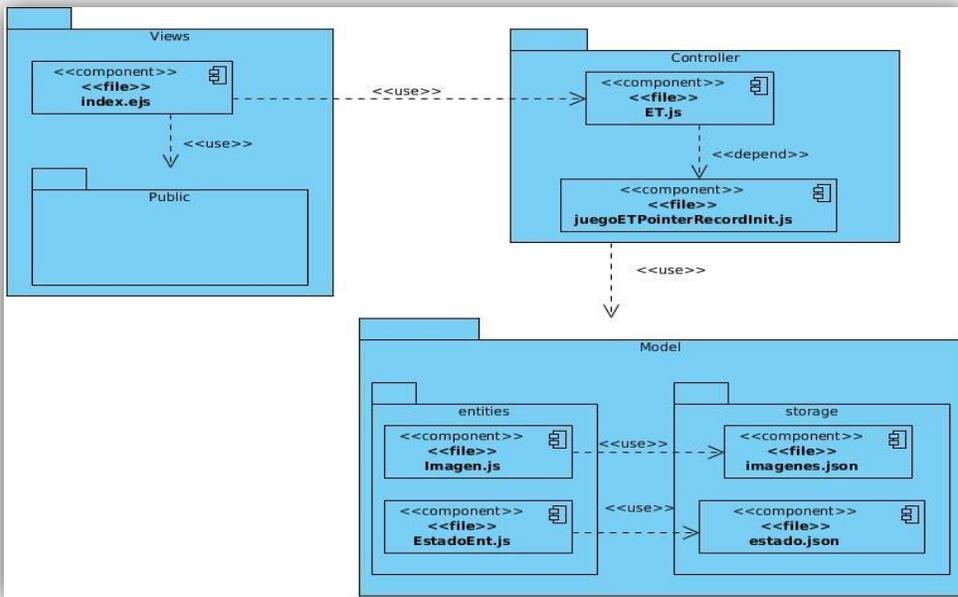
4.1. Modelo de implementación

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 los 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 (22).

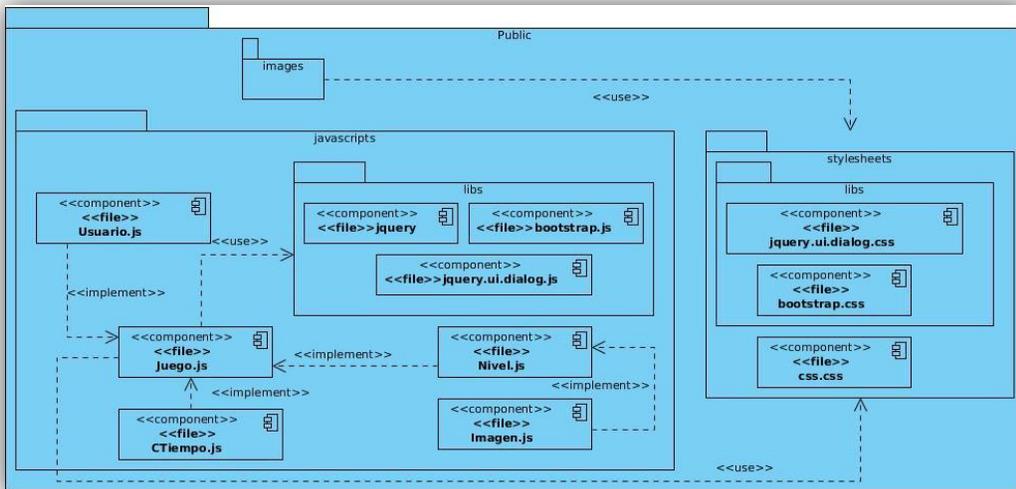
4.1.1. Diagrama de componentes

Los diagramas de componentes muestran tanto los componentes software (código fuente, binario y ejecutable) como las relaciones lógicas entre ellos en un sistema. Representan todos los tipos de elementos del software implicados en la fabricación de aplicaciones informáticas.

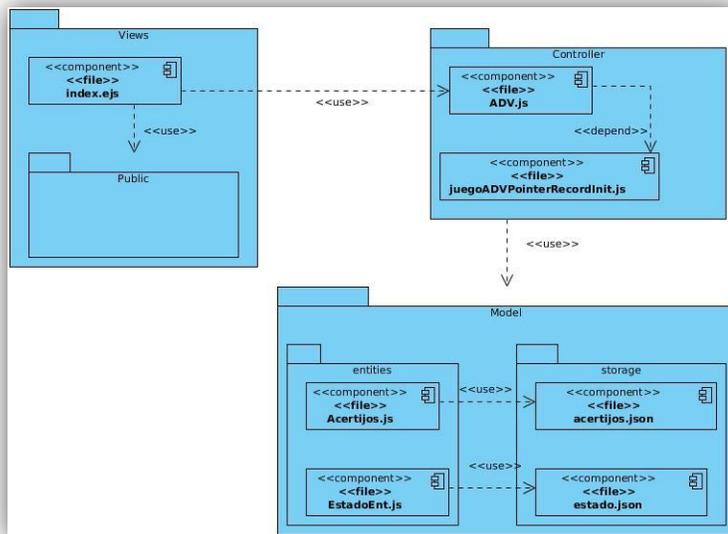
Los siguientes diagramas son la representación de los juegos Habitando mi isla, Rescatando el tesoro y Siguiendo la pista respectivamente dividido según la arquitectura basada en el patrón MVC.



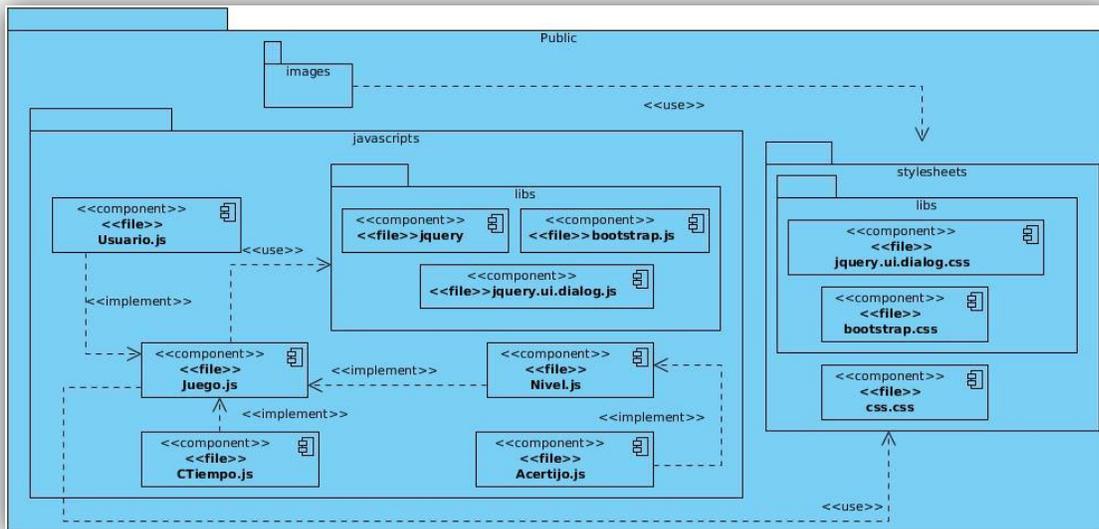
Figura# 13 Diagrama de componentes general del juego Habitando mi isla.



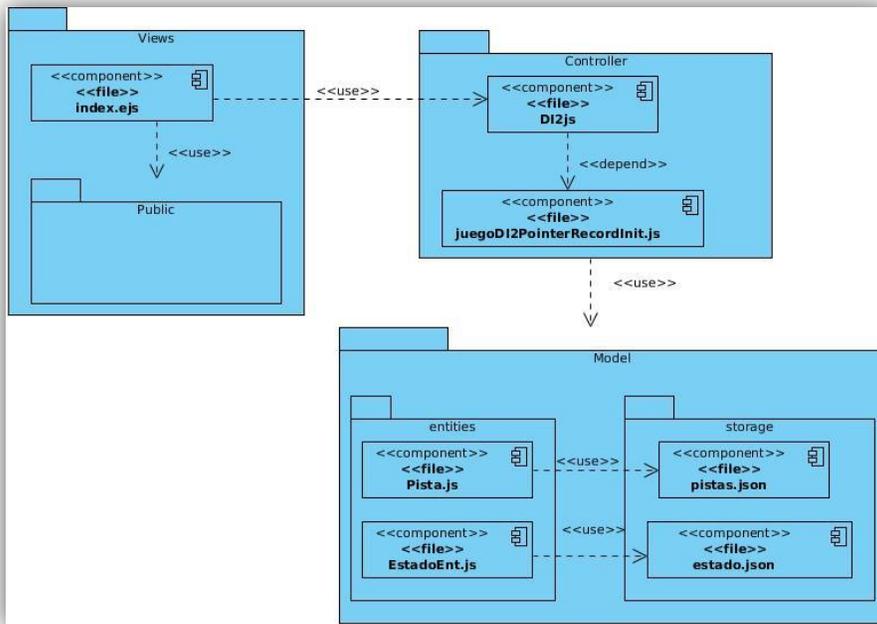
Figura# 14: Diagrama de componentes del paquete *Public* del juego Habitando mi isla.



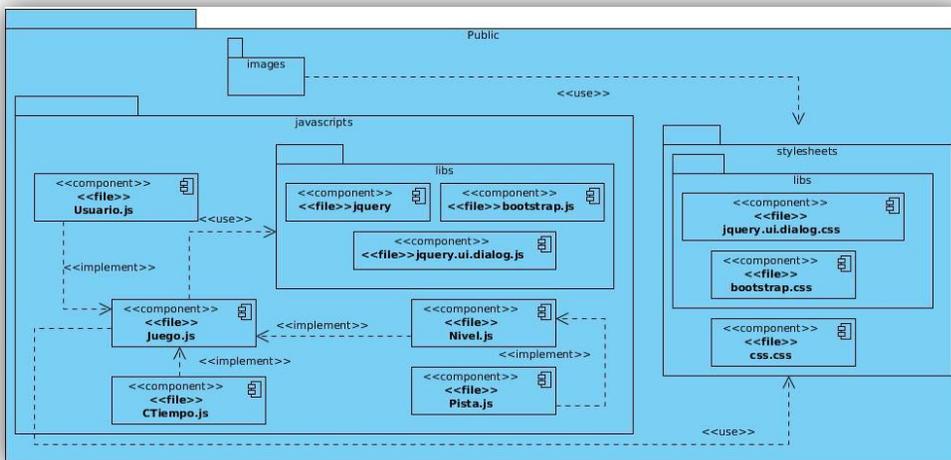
Figura#15 Diagrama de componentes general del juego Rescatando el tesoro.



Figura# 16: Diagrama de componentes del paquete *Public* del juego Rescatando el tesoro.



Figura# 17 Diagrama de componentes general del juego Siguiendo la pista.

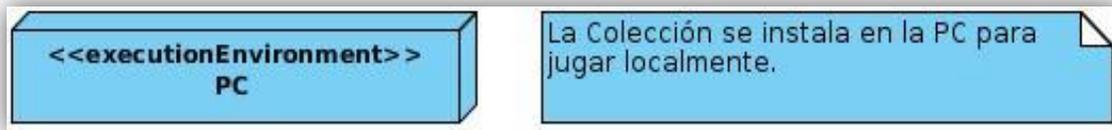


Figura# 18: Diagrama de componentes del paquete Public del juego Rescatando el tesoro.

4.1.2. Diagrama de despliegue

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es una colección de nodos;

donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar. (55)



Figura# 19 Diagrama de despliegue MundoClick offline.

4.2. Pruebas de software

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 (56). 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 (57).

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.

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 (58).

Para la realización de las pruebas existen varias técnicas (58):

- ✚ 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.

4.2.1. Análisis de los resultados

Las pruebas se realizaron en 2 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 web, sin tener en cuenta las características propias de la plataforma. Al ejecutar las tareas de integración con la plataforma Pointer fue necesario realizar un conjunto de transformaciones mínimas para permitir que los juegos se ejecutaran a través de dicha Plataforma.

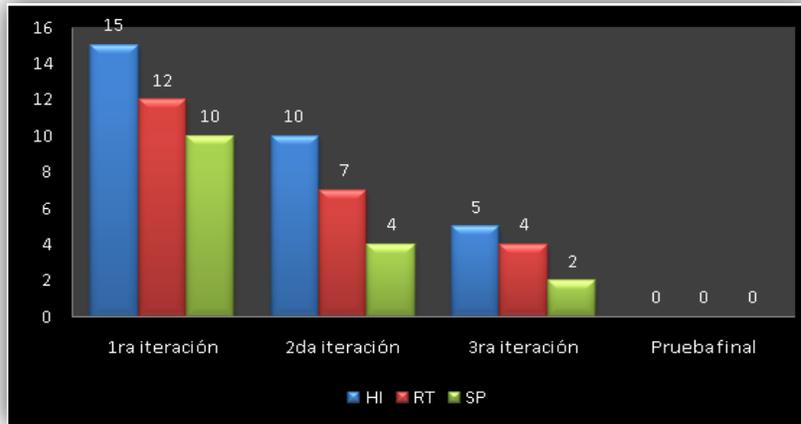
Las NC encontradas responden a dos tipos:

- ✚ Error de aplicación.
- ✚ No correspondencia entre los diseños de caso de prueba y el producto de software obtenido.

Estas no conformidades fueron resueltas con diferentes niveles de prioridad:

- ✚ Prioridad Alta: NC asociadas a errores de aplicación.
- ✚ Prioridad Media: NC asociadas a la no correspondencia.
- ✚ 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.



Figura# 20: Resultado de las pruebas.

4.3. Conclusiones

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 generales

El estudio del arte permitió obtener elementos teóricos que permiten inferir 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, tales como: incorporar el elemento tiempo mientras se juega, posibilidad de obtener bonificaciones y que varios usuarios puedan jugar en una misma estación de trabajo en modo multijugador.

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 y 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.

Recomendaciones

Con el cumplimiento del objetivo de la aplicación y teniendo como base el estudio efectuado durante la realización de esta investigación, se recomienda:

- ✚ Validar el diseño pedagógico por el Ministerio de Educación, en correspondencia a las peculiaridades de los niños de la enseñanza Primaria para los cuales fueron diseñados los juegos.
- ✚ Desplegar la colección en la cual están integrados los juegos para que los niños puedan de esta manera desarrollar nuevas habilidades.

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. **Ricardo, Carlos Expósito.** *Conceptos generales de Software.* 2009.
3. **Pivec, Maja, Koskinen, Tapio y Tarín, Lluís.** eLearning Papers. [En línea] [Citado el: 16 de Enero de 2013.] <http://www.elearningpapers.eu/es/paper/aprendizaje-basado-en-juegos-nuevas-pr-cticas-nuevas-aulas>.
4. **CIENTÍFICA, EL PROCESO DE INVESTIGACIÓN.** *Rolando Alfredo Hernández León y Sayda Coello González.* s.l. : Universitaria del Ministerio de Educación Superior, 2011.
5. **L Ramos Pérez, J Domínguez Lovaina, X Gavilondo Mariño, C Fresno Chávez.** Centro Nacional de Información de Ciencias Médicas-Infomed. [En línea] 2008. [Citado el: 12 de Noviembre de 2012.] http://bvs.sld.cu/revistas/aci/vol18_4_08/aci61008.htm.
6. **Rojas, Nemecio Núñez.** Concepción del Software educativo. La Webquest, el aula virtual y el desarrollo de competencias para la investigación en los estudiantes del I Ciclo de educación - USAT. [En línea] 2010. [Citado el: 12 de Noviembre de 2012.] <http://www.eumed.net/libros/2010a/669/>.
7. **Coordinación General de Investigación y Desarrollo de Modelos Educativos.** Coordinación General de Investigación y Desarrollo de Modelos Educativos. [En línea] [Citado el: 24 de Noviembre de 2012.] http://investigacion.ilce.edu.mx/panel_control/doc/c36,evaluacsoft.pdf.
8. **Marqués, Perez.** El software educativo. [En línea] Universidad Autónoma de Barcelona. [Citado el: 24 de Noviembre de 2012.] http://www.lmi.ub.es/te/any96/marques_software.
9. **Panqueva, Álvaro Galvis.** SliderShare Inc. [En línea] [Citado el: 24 de Noviembre de 2012.] <http://slidershare.org>.
10. **Carbó, César Infante.** [En línea] [Citado el: 3 de Diciembre de 2012.] http://www.roca.rimed.cu/index.php?option=com_content&view=article&id=115:el-juego-educativo-informatico-sus-caracteristicas-y-componentes-didacticos-esenciales&catid=114:juegos-educativos&Itemid=117.

11. **Rodríguez., Esmeralda Jiménez.** *La importancia del juego.* España : s.n. ISSN 1696-7208, 2006, Vol. III..
12. **MARQUÉS, P.** VENTAJAS E INCONVENIENTES DEL MULTIMEDIA EDUCATIVO. [En línea] 2000. [Citado el: 20 de enero de 2013.] <http://www.peremarques.net/ventajas.htm>.
13. **MÁRQUEZ RODRÍGUEZ, ALEIDA.** *Habilidades: reflexiones y proposiciones para su evaluación.* Instituto Superior Pedagógico. Santiago de Cuba : s.n., 1993.
14. **CASTILLO, ROSA MARÍA BERNAL.** *SISTEMA DE TAREAS PARA EL DESARROLLO DE HABILIDADES MATEMÁTICAS GENERALES EN EL NIVEL MEDIO SUPERIOR.* Nuevo León : s.n., 1999.
15. **Grupo de Trabajo de SCONUL.** *Aptitudes para el acceso y uso de la información en la enseñanza superior: la postura de Sconul.* Granada : Universidad de Granada, 1999.
16. **Cabello, Roxana.** *APROXIMACIÓN AL ESTUDIO DE COMPETENCIAS TECNOLÓGICAS.*
17. **Rodríguez Rodríguez, Jorge Remberto.** *Arquitectura de software. Guía base. Proyecto MundoClick.* 2013.
18. **Multimedia, Cibal.** PipoClub.com. [En línea] 1996. [Citado el: 7 de Diciembre de 2012.] <http://www.pipoclub.com/>.
19. **Lic. Liana Isabel Araujo Pérez¹, Lic. Héctor Matías González¹, Ing. Osdalme Fuentes Colina¹, Ing. Mailyñ Cabrera Torres¹, MSc. Ismael Armando Nodarse Mora¹, Ing. Ana María Álvarez Valdés¹, Ing. Osmany Montes De Oca Rodríguez¹ y otros.** *COLECCIÓN DE SOFTWARE EDUCATIVO MULTIPLATAFORMA "LA CAJA MÁGICA".* 2011.
20. *Informática Aplicada a la Gestión Pública.* [En línea] UMU, Facultad de Derecho. [Citado el: 25 de Noviembre de 2012.] <http://www.um.es/docencia/barzana/IAGP/IAGP2-Metodologias-de-desarrollo.html>.
21. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo, La guía completa del Proceso Unificado, escrita por sus creadores.* [En línea] 2000.
22. **Pressman, Roger A.** *Ingeniería de software: Un enfoque práctico.* 5. 2005. pág. 404. 970105473.
23. **Larman, Craig.** *UML y Patrones: Introducción al análisis y diseño orientado a objetos.* México : PRENTICE HALL, 1999. 970-17-0261-1.
24. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** *El Lenguaje Unificado de Modelado.* California : Addison Wesley, 1998. págs. 3-4.

25. **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/>.
26. *Herramientas libres para modelar software.* **Callejas Cuervo, Mauro y Baquero Moreno, Oscar Yovany.** Colombia : Revista Facultad de Ingeniería, UPTC, 2005. 0121–1129.
27. RENA- Red Escolar Nacional. [En línea] Ministerio del poder Popular para Ciencia, Tecnología e Innovación. [Citado el: 12 de Enero de 2013.] <http://www.rena.edu.ve/cuartaEtapa/Informatica/Tema13.html>.
28. **Diego Gauchat, Juan.** *El gran libro de HTML5, CSS3 y Javascript*. Barcelona : MARCOMBO, S.A., 2012. págs. 1-2. 978-84-267-1782-5.
29. **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/>.
30. **Sánchez-Heredero Pérez, Alberto.** *Accesibilidad a los contenidos audiovisuales en la Web a través de HTML5*. Leganés : Universidad Carlos III de Madrid, 2011.
31. w3schools.com. *w3schools.com*. [En línea] [Citado el: 6 de Diciembre de 2012.] http://www.w3schools.com/html/html5_intro.asp.
32. **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.
33. **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/>.
34. —. 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/>.
35. 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>.
36. **Diego Gauchat, Juan.** *El gran libro de HTML5, CSS3 y JavaScript*. Barcelona : Marcombo, S.A., 2012. 978-84-267-1782-5.

37. w3schools. *w3schools*. [En línea] [Citado el: 7 de Enero de 2013.] <http://www.w3schools.com/json/default.asp>.
38. Introducción a JSON. *Introducción a JSON*. [En línea] [Citado el: 7 de Enero de 2013.] <http://www.json.org/json-es.html>.
39. **Gutiérrez, Javier J.** ¿Qué es un framework web? . [En línea] [Citado el: 24 de Noviembre de 2012.] http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
40. DesarrolloWeb. *DesarrolloWeb*. [En línea] 19 de Septiembre de 2012. [Citado el: 5 de Diciembre de 2012.] <http://www.desarrolloweb.com/manuales/manual-jquery.html>.
41. **Herron, David.** *Node Web Development*. 2011.
42. **Rodríguez, Txema.** Bootstrap . [En línea] 16 de junio de 2012. [Citado el: 22 de Diciembre de 2012.] <http://www.genbetadev.com/frameworks/bootstrap>.
43. **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>.
44. **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>.
45. Netbeans IDE. *Netbeans IDE*. [En línea] 2012. [Citado el: 6 de Diciembre de 2012.] <http://netbeans.org/features/index.html>.
46. **Larman, Craig.** *UML y patrones: Introducción al análisis y diseño orientado a objetos*. México : PRENTICE HALL, 1999. pág. 11. 970-17-0261-1.
47. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software*. Madrid : Pearson Educación, S.A., 2000. págs. 5-8. 84-7829-036-2.
48. **García, Joaquín.** IngenieroSoftware. [En línea] [Citado el: 22 de Enero de 2013.] <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>.

49. IBM WebSphere Commerce. Guías de programación y aprendizaje. [En línea] [Citado el: 24 de Noviembre de 2012.] <ftp://ftp.software.ibm.com/software/websphere/commerce/55/es/WC55ProgrammingGuide.pdf>..
50. **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.
51. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady**. *El Lenguaje Unificado de Modelado*. California : Addison Wesley, 1998. págs. 25-27.
52. **Schmuller, Joseph**. *Aprendiendo UML en 24 horas*. México : PEARSON EDUCACION,S.A, 2000. 968-444-463-X.
53. Carmen. Estándares de codificación . *Carmen. Estándares de codificación* . [En línea] 1 de abril de 2013. <http://www.cisiad.uned.es/carmen/estilo-codificacion.pdf>.
54. pedrormarcos.com. *pedrormarcos.com*. [En línea] 2012. [Citado el: 29 de Marzo de 2013.] <http://www.pedrormarcos.com/escritura-estilo-camelcase/>.
55. **Wesley, Addison**. El proceso unificado de desarrollo de software. s.l. [En línea] 1999. [Citado el: 15 de marzo de 2013.]
56. **Pressman, Roger A**. *Ingeniería del Software:Un enfoque práctico*. 2005. págs. 281-284. 970105473.
57. —. *Ingeniería del Software:Un enfoque práctico*. 2005. págs. 284-293. 970105473.
58. —. *Ingeniería del Software:Un enfoque práctico*. 2005. págs. 294-302. 970105473.

Bibliografía

GALLEGO, JUAN PABLO GOMEZ. Fundamentos de la Metodología RUP. [En línea] Universidad Tecnológica de Pereira, 2007. [Citado el: 24 de noviembre de 2012.]

MSc. Judith Cárdenas Roble, MSc. Amalia Enríquez Rodríguez, MSc. Lenia Robledo Ortega. *NUEVAS TECNOLOGÍAS PARA EL APOYO DEL APRENDIZAJE DE NIÑOS CON RETRASO MENTAL LEVE ENTRE 7 Y 9 AÑOS DE EDAD.* s.l. : Universidad de Matanzas, 2011.

Kwan, Fancisco. *Impacto de las nuevas tecnologías de la información y comunicación en la gestión del conocimiento.*

Plasencia, Nadiela Milan Cristo y Ernesto Yordi. *Herramienta de selección didáctica para guiar el aprendizaje interactivo en el módulo Ejercicios de la colección El Navegante.* La Habana : s.n., 2012.

Marcos, Pedro R. PEDRORMARCOS.COM. [En línea] 2012. [Citado el: 25 de marzo de 2013.]

HEREDIA, Y. P. *METODOLOGÍAS PARA EL DESARROLLO DE SOFTWARE EDUCATIVO: UN ESTUDIO COMPARATIVO.* Ciudad de la Habana, Universidad de las Ciencias Informaticas. : s.n., 2007.

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>.

Larman, Craig. *UML y Patrones: Introducción al análisis y diseño orientado a objetos.* México : PRENTICE HALL, 1999. pág. 55. 970-17-0261-1.

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.

Pressman, Roger A. *Ingeniería del Software: Un Enfoque Práctico.* 2005. 9701054733.

Larman, Craig. *UML y Patrones: Introducción al análisis y diseño orientado a objetos.* México : PRENTICE HALL, 1999. págs. 185-192. 970-17-0261-1.

Glosario de términos

API: Una interfaz de programación de aplicaciones o API (del inglés *application programming interface*) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro *software* como una capa de abstracción. Usados generalmente en las bibliotecas.

HTML: *Hypertext Markup Language*. Lenguaje usado para escribir documentos para servidores *World Wide Web*. Es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.

Herramientas CASE: Herramientas utilizadas para el desarrollo de proyectos de Ingeniería de Software.

JavaScript: Es un lenguaje de script multiplataforma basado en objetos.

RUP: *Rational Unified Process* (Proceso Unificado de desarrollo). Metodología para el desarrollo de Software.

TIC: Tecnologías de la Información y las Comunicaciones.

UML: *Unified Modeling Language*. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de *software*.

Anexos

Anexo 1:

Guion del juego “Rescatando el tesoro” para la colección de juegos educativos MundoClick.

Historia

La historia se basa en rescatar el tesoro del reino que lo tiene escondido un brujo. El brujo encantó el castillo de forma que las puertas están ocultas y solo se descubren respondiendo acertijos. En la medida que acierte se irá descubriendo el camino al tesoro. Al final se habrá recorrido todo el camino y se encontrará el tesoro.

Objetivo

-  El objetivo del juego es crear habilidades en el uso del teclado.
-  En el desarrollo de la historia, el objetivo es encontrar el tesoro.

Descripción

Interfaces:

-  En el cómic inicial se debe mostrar al brujo robando el tesoro y escondiéndolo en un castillo (que se entienda que está difícil llegar hasta él).
-  En el cómic final debe mostrarse el tesoro encontrado.
-  El cómic, en caso de perder el juego, debe mostrar al brujo.
-  En el momento en que se accede a jugar se mostrará una pantalla con un camino oculto que se descubrirá si se responden bien las adivinanzas, la cantidad de adivinanzas estarán en dependencia del nivel.
-  El objeto que da paso al próximo nivel puede ser representado por una puerta, un espejo, una ventana, una amenaza (ejemplo un bicho, un objeto cortante) o un obstáculo cualquiera.
-  Para descubrir el obstáculo se hará en forma de rompecabezas, cuando se responda una adivinanza correctamente se descubrirá una parte de la puerta.
-  La adivinanza o acertijo será mostrado dentro de la misma pantalla del juego, (pudiera ser en una pared o algún componente de la pantalla) donde se le permita escribir la respuesta.
-  El fondo de cada uno de los niveles pudiera ser representado por habitaciones del propio castillo, espacios abiertos o cerrados.
-  El juego se pierde cuando los puntos llegan a 0.

- ✚ El transcurso del tiempo debe mostrarse en un reloj donde se vea claramente el tiempo restante.

Secuencia del juego por niveles

El juego estará compuesto por 6 niveles:

- ✚ Para vencer el primer nivel se deberán responder 7 acertijos en 8 minutos.
- ✚ Para vencer el segundo nivel se deberán responder 7 acertijos en 7 minutos.
- ✚ Para vencer el tercer nivel se deberán responder 7 acertijos en 6 minutos.
- ✚ Para vencer el cuarto nivel se deberán responder 7 acertijos en 5 minutos.
- ✚ Para vencer el quinto nivel se deberán responder 7 acertijos en 4 minutos.
- ✚ Para vencer el sexto nivel se deberán responder 7 acertijos en 3 minutos.

Puntos

- ✚ Cada acertijo que se responda correctamente acumulará 3 puntos.
- ✚ Cada acertijo que se responda mal resta 2 puntos.
- ✚ Al finalizar el nivel se otorga 1 punto por cada bono no utilizado.
- ✚ Los puntos son acumulativos y pasan al siguiente nivel.
- ✚ En los niveles, cuando este se acabe el tiempo, se darán 5 segundos adicionales por cada 1 punto. Dicho punto se restará de los acumulados.

Bonos

- ✚ **Lupa:** resalta la primera letra de la respuesta.
- ✚ **Súper lupa:** resalta todas las letras de la respuesta.
- ✚ **Contador:** indica la cantidad de letras que tiene la respuesta.
- ✚ **Lápiz mágico:** indica específicamente en que letra o letras de la palabra se encuentra el error ortográfico.
- ✚ **Tiempo extra:** se le asignarán 20 segundos extras y se congela el tiempo del juego.
- ✚ **Rompe acertijo:** da la respuesta del acertijo.

Forma de obtener bonos: en ocasiones un acertijo tendrá el icono de un bono. Se obtendrá sólo si responde correctamente. Al pasar el nivel se resetean.

Forma de activar el bono: Se activan presionando la tecla del número correspondiente al bono. Se debe mostrar, con qué número se activarán.

Anexo 2:

Guion del juego Siguiendo la pista para la colección de juegos educativos MundoClick

Historia

En la ciudad están ocurriendo extraños robos en diferentes lugares. El habilidoso ladrón, para burlarse de la policía va dejando pistas ocultas. Ya anunció que con suficientes pistas su escondite sería descubierto, ya que dice que la policía no es tan inteligente como él para descifrar sus pistas. Se te ha asignado el caso pues eres el detective más inteligente de la policía. Tu misión es recopilar las pistas y descifrarlas para dar con el escondite del ladrón y recuperar las cosas que se ha robado.

Objetivo

-  Descifrar las pistas para recuperar los objetos robados y capturar al ladrón.
-  Desarrollar habilidades en el uso del teclado.

Descripción

Interfaces

-  Comics: **Inicial**; un ladrón con una bolsa robada burlando a la policía.
-  Comics: **Intermedio**; detective buscando y descifrando pistas.
-  Comics: **Final**; objetos robados recuperados y ladrón tras las rejas.
-  Cuando se está descifrando una pista se muestran: las letras del alfabeto, un recuadro donde se irá mostrando la imagen a descubrir, un recuadro con la frase a descubrir y en algún lugar los intentos que tiene el jugador para equivocarse.
-  La imagen a descubrir se irá mostrando paulatinamente a medida que se acierten letras.
-  Para descifrar la pista necesitas completar la frase sin agotar los intentos de fallo.
-  Cuando se señala una letra esta se desactiva, si la letra pertenece a la frase aparece entonces en los lugares de la frase donde va la letra, si no pertenece a la frase la letra se marca de forma distinta y se representa de alguna forma que se ha agotado un intento.

- ✚ En una parte de la pantalla debe contener las pistas descifradas hasta el momento y la cantidad que restan para poder descubrir donde están guardados los objetos robados por el ladrón.
- ✚ Cuando se han fallado tantas pistas que es imposible completar las suficientes para pasar de nivel entonces se muestra una pantalla donde el detective es removido de su cargo y el juego termina, tendría que comenzar de nuevo el juego porque el ladrón escapó.
- ✚ Al terminar cada nivel se le otorga al detective una condecoración y aumenta de rango.

Secuencia del juego por niveles

- ✚ Cada uno de los 6 niveles representan un reto para el jugador sin perder de vista los objetivos del juego.
- ✚ Para completar un nivel hay que descifrar X pistas de Y posibles. A medida que avanzan los niveles son más las pistas que hay que descubrir

Nivel 1, descubrir X pistas en un tiempo Y minutos

Nivel 2, descubrir X+10 pistas en un tiempo Y minutos

Nivel 3, descubrir X+15 pistas en un tiempo Y-2 minutos

Nivel 4, descubrir X+20 pistas en un tiempo Y-3 minutos

Nivel 5, descubrir X+25 pistas en un tiempo Y-3 minutos

Nivel 6, descubrir X+30 pistas en un tiempo Y-5 minutos

Puntos

- ✚ Cada imagen descubierta aporta 3 puntos.
- ✚ Cada imagen que no se descubra resta -1 punto.
- ✚ Si la se descubre con ayuda de un bono aporta 2 puntos.

Bonos (se activan con el mouse)

- ✚ agregar intento (sirve para agregar otro intento cuando se está descubriendo una pista).
- ✚ Lupa (muestra la imagen que se está descubriendo cuando se pasa mouse por encima).
- ✚ Pista (resalta alguna de las letras que hay que contiene la frase).

Forma de obtener bonos: en ocasiones cuando se descifre una pista esta indicara que se obtuvo alguno de los bonos.

Forma de activar el bono: dando clic sobre el bono.

Distinciones

Se otorgan cuando se completa un nivel y es una distinción o grado X que engrandece al detective. Dichos grados pudieran ser Inspector, Comisario, entre otros.

Anexo 3:

Guion del juego **Habitando mi isla para la colección de juegos educativos MundoClick**

Historia

Eres dueño de una isla y has pedido por encargo todo tipo de animales, plantas, frutas y objetos para acondicionarla. El único problema es que el avión que se está usando para traer los recursos no puede aterrizar, por lo que estos deben tirarse en paracaídas. Para poder abrir el paracaídas hay que usar un control remoto. El mismo funciona si se escribe correctamente el nombre del objeto, animal o fruta que lo porta. Tu misión es lograr poner todo a salvo en tierra, para ello se te ha dado el control remoto.

Objetivo

-  Poner a salvo la mayor cantidad de animales, plantas y objetos posibles para tu isla.
-  Desarrollar habilidades en el uso del teclado

Descripción

Interfaces

-  Comics: **Inicial**; se muestra la isla solitaria.
-  Comics: **Intermedio**; se muestra la isla sobrevolada por una avioneta u avión que suelta cosas a las que se le abren paracaídas.
-  Comics: **Final**; se muestra la isla poblada.
-  En el momento en que se accede a jugar se mostrarán imágenes que van cayendo.
-  El fondo puede ser representado alguna locación natural cualquiera.
-  Las imágenes deben ser lo suficientemente claras como para que el jugador pueda identificar qué son.
-  Para poner a salvo lo que identifica a una imagen debe escribirse el nombre de la misma correctamente.

- ✚ Cuando una palabra correcta es escrita con errores ortográficos y se ha activado el bono “Señalador ortográfico” el jugador podrá hacer uso de él o de otra forma utilizar los bonos que se han obtenido hasta el momento.
- ✚ Los puntos deben mostrarse en forma de barra o algo que sea gráfico, que representen la batería del control remoto y que el niño pueda ir viendo cuánto le falta para completar el nivel o cuánto pierde cuando juega mal.
- ✚ Al terminar cada nivel se mostrará una pantalla donde se le notifique al jugador que ha pasado al siguiente nivel.

Descripción del control remoto: debe tener: teclado con las letras; teclas para activar los bonos e imágenes de los bonos; pantalla donde sale lo que se escribe; e indicador de estado de batería. El control remoto cambia por niveles, en su diseño debe tenerse en cuenta que a medida que el nivel cambia la batería es más potente y se agrega una tecla correspondiente al bono que aparece en ese nivel.

Secuencia del juego por niveles

- ✚ Cada uno de los 6 niveles representa un reto para el jugador sin perder de vista los objetivos del juego.
- ✚ Para completar un nivel hay que acumular como mínimo los definidos para ese nivel. (nivel 1->30, nivel 2->50, nivel 3->70, nivel 4->70, nivel 5->80, nivel 6->90)
- ✚ En el primer nivel se deben nombrar las imágenes correspondientes al nivel y el jugador tendrá un tiempo de 8 minutos.
- ✚ En el segundo nivel se deben nombrar las imágenes correspondientes al nivel y el jugador tendrá un tiempo de 7 minutos.
- ✚ En el tercer nivel se deben nombrar las imágenes correspondientes al nivel y el jugador tendrá un tiempo de 6 minutos.
- ✚ En el cuarto nivel se deben nombrar las imágenes correspondientes al nivel y el jugador tendrá un tiempo de 5 minutos.
- ✚ En el quinto nivel se deben nombrar las imágenes correspondientes al nivel y el jugador tendrá un tiempo de 4 minutos.
- ✚ En el sexto nivel se deben nombrar las imágenes correspondientes al nivel y el jugador tendrá un tiempo de 3 minutos.

Puntos

- ✚ Cada imagen identificada correctamente aporta 3 puntos.
- ✚ Cada imagen que no se identifique correctamente resta 1 punto.
- ✚ Si se identifica con ayuda de un bono suma 2 puntos.

Bonos (se activan con la tecla correspondiente al nivel)

- ✚ Teclado guía (en el teclado del control remoto se resaltan las letras que tiene el nombre de la imagen) (nivel 1, tecla 1).
- ✚ Indicador de cantidad de letras (indica en la pantalla del control remoto la cantidad de letras de la imagen) (nivel 2, tecla 2).
- ✚ Indicador de primera letra (indica solo la primera letra de la imagen a identificar) (nivel 3, tecla 3).
- ✚ Señalador ortográfico (indica específicamente en que letra o letras de la palabra se encuentra el error) (nivel 4, tecla 4).
- ✚ Congelar tiempo (congela el tiempo de caída de la imagen por 5 segundos) (nivel 5, tecla 5).
- ✚ Súper control remoto (da el nombre de la imagen) (nivel 6, tecla 6).

Forma de obtener bonos: en ocasiones una imagen tendrá el icono de un bono incrustado, de obtener la imagen se obtiene el bono que se pondrá de alguna forma visible para el jugador y podrá ser activado en el momento que se desee.

Forma de activar el bono: presionando la tecla correspondiente al bono que deberá estar indicada en el lugar donde se encuentre este. Dicha tecla deberá ser de las que no se usan para escribir las palabras del juego, ejemplo, números o signos de puntuación o combinaciones de estos.

Anexo 4:

Tabla 12: Descripción del caso de uso Realizar juego (Rescatando el tesoro).

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 o termina el juego.
Prioridad	Crítico

Precondiciones		
Postcondiciones		
Flujo de eventos		
Flujo básico Realizar juego		
	Actor	Sistema
1.	Selecciona la opción Jugar del menú principal.	
2.		Inicia el juego según lo especificado en el guion del mismo. (Ver Anexo 1)
3.		Inicia la secuencia de acertijos que debe ser identificado por el jugador (Ver caso de uso Identificar acertijo).
4.		Muestra al jugador el momento en que pasa o sube de nivel.
5.		Muestra cuando un jugador obtiene un bono: (Ver caso de uso Asignar bono).
6.		En la trayectoria del juego se muestran los siguientes datos: <ul style="list-style-type: none"> - Tiempo - Bonos obtenidos - Puntos - Intentos
7.		Cuando se termina el juego muestra el nombre, los puntos y la efectividad alcanzada en el juego.
Relaciones	CU Incluidos	Identificar acertijo

	CU Extendidos	Asignar bono
--	----------------------	--------------

Anexo 5:**Tabla 13: Descripción del caso de uso Identificar acertijo (Rescatando el tesoro).**

Casos de uso	Identificar acertijo	
Actores	Jugador	
Resumen	El caso de uso inicia cuando el jugador selecciona la opción jugar y termina cuando haya identificado el acertijo que muestra el sistema.	
Prioridad	Crítico	
Precondiciones		
Postcondiciones	Se muestra si el acertijo ha sido identificado correcta o incorrectamente.	
Flujo de eventos		
Flujo básico Identificar acertijo		
	Actor	Sistema
1.		Muestra los acertijos.
2.	Introduce la respuesta del acertijo mostrado.	
3.		Verifica respuesta dada por el jugador.
4.		Muestra resultado de la verificación Correcto o Incorrecto.
5.		Suma 3 puntos.
6.		Muestra próximo acertijo y se comenzará a ejecutar nuevamente el flujo básico de eventos a partir del paso 1.
7.		Termina el caso de uso.
Flujos alternos		

4a. Los datos son incorrectos.		
	Actor	Sistema
1.		Resta 2 puntos.
2.		Se ejecuta el paso 3 del flujo normal de eventos.

Anexo 6:**Tabla 14: Descripción del caso de uso Asignar bono (Rescatando el tesoro).**

Casos de uso	Asignar bono	
Actores	Jugador	
Resumen	El caso de uso inicia cuando se haya ejecutado el caso de uso Realizar juego.	
Prioridad	Crítico	
Precondiciones	Que se haya ejecutado el caso de uso Realizar juego.	
Postcondiciones		
Flujo de eventos		
Flujo básico Asignar bono		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor identifica el acertijo que trae incluido un bono.	
2.		<p>Adiciona los bonos de forma aleatoria en todos los niveles si el jugador identificó el acertijo que lo traía incluido.</p> <ul style="list-style-type: none"> - Lupa. - Súper lupa. - Contador. - Lápiz mágico.

		<ul style="list-style-type: none"> - Tiempo extra. - Rompe acertijo.
3.		Permite que el jugador continúe identificando los acertijos que se van mostrando.
4.		Se activa la tecla correspondiente al bono.
5.		Termina el caso de uso.
Relaciones	CU Incluidos	
	CU Extendidos	Asignar bono en el CU Realizar juego

Anexo 7:

Tabla 15: Descripción del caso de uso Utilizar bono (Rescatando el tesoro).

Casos de uso	Utilizar bono	
Actores	Jugador	
Resumen	El caso de uso inicia cuando se haya ejecutado el caso de uso Asignar bono.	
Prioridad	Crítico	
Precondiciones	Que se haya ejecutado el caso de uso Asignar bono.	
Postcondiciones		
Flujo de eventos		
Flujo básico Utilizar bono		
	Actor	Sistema
1.	Selecciona la tecla correspondiente a uno de los bonos asignados.	
2.		Realiza una acción determinada en dependencia del bono seleccionado: Bono:

		<ul style="list-style-type: none"> - Lupa: Resalta la primera letra de la respuesta. - Súper lupa: Resalta todas las letras de la respuesta. - Contador: Indica la cantidad de letras que tiene la respuesta. - Lápiz mágico: Indica en que letra o letras de la respuesta está el error ortográfico. - Tiempo extra: Se congela el tiempo del juego por 20 segundos. - Rompe acertijo: Da la respuesta del acertijo.
3.		Termina el caso de uso.

Anexo 8:

Tabla 16: Descripción del caso de uso Terminar juego (Rescatando el tesoro).

Casos de uso	Terminar juego
Actores	Jugador
Resumen	El caso de uso inicia cuando el jugador selecciona la opción Salir y termina cuando se cierra el juego.
Prioridad	Crítico
Precondiciones	
Postcondiciones	Se termina el juego.
Flujo de eventos	
Flujo básico Terminar juego	

	Actor	Sistema
1.	Selecciona la opción Salir.	
2.		Muestra un mensaje de confirmación: - ¿Desea realmente salir del juego?
3.		Guarda los resultados del jugador.
4.		Termina el caso de uso.
Flujos alternos		
2a. Cancelar salir del juego.		
	Actor	Sistema
1.	Selecciona la opción cancelar.	
2.		Sigue la secuencia del juego.