

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 4



TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

Desarrollo de los juegos: “Parchís” y “Localiza y Aprende”
para MundoClick.

Autor: Javier Cruz Hernández

Tutores: Ing. Héctor Luis Reyes Pupo

Ing. Nersa Doraines Acosta Labrada

Fecha: junio, 2013.

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo al Centro de Desarrollo de Tecnologías para la Formación (FORTES) de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Autor: Javier Cruz Hernández

Firma: _____

Tutores:

Ing. Héctor Luis Reyes Pupo.

Firma: _____

Ing. Nersa Doraines Acosta Labrada

Firma: _____

Datos de contacto

Tutor: Ing. Héctor Luis Reyes Pupo

Correo electrónico: hpupo@uci.cu.

Profesión: Ingeniero en Ciencias Informáticas.

Años de graduado: 3.

Años de experiencia en el tema: 6.

Tutor: Ing. Nersa Doraines Acosta Labrada.

Correo electrónico: ndacosta@uci.cu.

Profesión: Ingeniera en Ciencias Informáticas.

Años de graduado: 2.

Años de experiencia en el tema: 1.

Dedicatoria

A mis abuelos Teresa y Domingo donde quiera que estén. A mis padres por darme la vida. A mis hermanos, Annalie, Janier y Kevin que les sirva de ejemplo.

Agradecimientos

A mis tutores por la ayuda que me dieron en la realización de este trabajo. A Mailin que fue casi una tutora más. A mi familia por el apoyo de toda la vida. A mi amigo y hermano en esta escuela Frank por servirme de ejemplo. A mi gente, Ernesto, la Gata, Víctor, la Rubia, la Flaca, Diwel, Jorge, Asley, aunque andemos dispersos sé que puedo contar con ellos. A los frikies del café por tantas noches perdidas.

A todos mis amigos en esta escuela, Yasmany, Draco, Yaray, Jorge Manchita, Carlos, Rayko, Ariel, Ortiz, Yurell. A mi gente de la calle, Merino, Yanet, Nieto, Esteban, Omar. A Yenima por haberme soportado estos años y haber sido mi apoyo. A mi grupo de baile, D Calle G. A mi novia Anna por soportar mis pesadeces, apoyarme y pelearme y estar conmigo en los momentos más duros de la tesis, y por la ayuda tan grande que me dio, si hoy me gradúo es gracias a ella. Mi gente del antiguo 135 202, Gabriel, Roylan, Javiel, Yerandy, Mario, Azpiri. A todas esas personitas especiales que en algún momento formaron parte de mi vida en esta escuela, de cada una aprendí algo nuevo y guardo siempre los recuerdos alegres. A mi amiga especial María, más una amiga es mi hermana grande, y que ahora me trae un sobrinito.

Resumen

La Universidad de las Ciencias Informáticas cuenta con el proyecto Multisaber-Navegante, perteneciente al Centro de Tecnologías para la Formación (FORTES), el cual se dedica a la creación de sistemas de formación educativa. Este proyecto ha comenzado a trabajar en una nueva colección de juegos denominada MundoClick. Por ello surge la necesidad de desarrollar juegos educativos cuyas características se ajusten a las de la plataforma Pointer, la cual contendrá los juegos de la colección.

El presente trabajo aborda el desarrollo de los juegos educativos “Parchís” y “Localiza y Aprende”, los cuales serán integrados al paquete de juegos educativos MundoClick. La estructura del trabajo se organizó a partir de la definición de cuatro capítulos. Para el desarrollo de los mismos fueron utilizados métodos teóricos y empíricos; los que permitieron estudiar el desarrollo de los videojuegos a nivel nacional e internacional, así como, el uso del Lenguaje Unificado de Modelado (UML) como soporte a la metodología RUP y Visual Paradigm como herramienta de modelado. Como lenguajes de programación HTML5, CCS3 y JavaScript, JQuery y Node.js como frameworks de desarrollo y el NetBeans 7.1.2 como IDE (Entorno de Desarrollo Integrado). Se realizaron las actividades que comprenden todo el ciclo de vida del producto.

Los juegos brindarán un sistema dirigido especialmente para niños de 6 a 8 años, lo que contribuirá al proceso de enseñanza-aprendizaje y al desarrollo de la memoria y la motivación hacia el estudio de forma amena e interactiva.

Como principales resultados de la investigación se obtuvo la implementación de los juegos “Parchís” y “Localiza y Aprende” así como su integración a la plataforma de juegos educativos Pointer.

Palabras clave: juegos educativos, MundoClick, plataforma.

Índice de contenido

Introducción	11
Capítulo 1: Fundamentación teórica	15
1.1. Introducción	15
1.2. Los videojuegos	15
1.3. Clasificación de los videojuegos	15
1.4. Aporte pedagógico de los juegos educativos	17
1.5. Sistemas similares	17
1.6. Plataforma Pointer	20
1.7. Metodología a utilizar	20
1.7.1. Metodología RUP	21
1.7.2. Lenguaje Unificado de Modelado (UML)	22
1.8. Herramienta de modelado: Visual Paradigm 8.0	22
1.9. Lenguajes de programación	23
1.9.1 HTML 5	24
1.9.2 CSS 3	24
1.9.3 JavaScript	24
1.10. Frameworks	25
1.10.1. Node.js v0.8.8	25
1.10.2. jQuery v1.9	26
1.11. Entorno de Desarrollo Integrado(IDE)	26
1.11.1. NetBeans 7.1.2	26
1.12. Control de versiones: Subversion	27
1.13. Conclusiones parciales	27
Capítulo 2: Características del sistema	28
2.1. Introducción	28
2.2. Descripción general de la propuesta del sistema	28
2.3. Especificación de los requisitos de software	29
2.3.1. Requisitos funcionales	29
2.3.2. Requisitos no funcionales	31
2.4. Modelo conceptual	33
2.4.1. Descripción de los conceptos	34
2.5. Definición de Casos de Uso	35
2.5.1. Diagramas de Casos de Uso	36

2.5.2. Descripción textual de los casos de uso	37
2.6. Conclusiones parciales	50
Capítulo 3: Diseño del sistema	51
3.1. Introducción	51
3.2. Modelo de diseño	51
3.2.1. Diagrama de clases del diseño	51
3.2.2. Diagramas de secuencia	53
3.3. Definiciones de diseño	54
3.3.1. Patrón arquitectónico	54
3.3.2. Integración a la plataforma Pointer	56
3.3.3. Estándar de codificación	56
3.4. Conclusiones parciales	57
Capítulo 4: Implementación y prueba	58
4.1. Introducción	58
4.2. Modelo de implementación	58
4.2.1. Diagrama de componentes	58
4.2.2. Diagrama de despliegue	59
4.3. Conclusiones parciales	64
Conclusiones generales	65
Recomendaciones	66
Glosario de Términos	71
Anexos	73
Anexo 1: Guión del juego “Parchís”	73
Anexo 2: Guión del juego “Localiza y Aprende”	74

Índice de Figuras

Figura 1. Modelo conceptual.....	34
Figura 2. Diagrama de Casos de Uso del juego “Parchís”.....	36
Figura 3. Diagrama de Casos de Uso del juego “Localiza y Aprende”.....	37
Figura 4. Diagrama de clases del diseño del juego “Parchís”.....	52
Figura 5. Diagrama de clases del diseño del juego “Localiza y Aprende”.....	52
Figura 6. Diagrama de secuencia del caso de uso Realizar juego. Juego de “Parchís”.	53
Figura 7. Diagrama de secuencia del caso de uso Realizar juego. “Localiza y Aprende”.....	54
Figura 8. Patrón de arquitectura Modelo-Vista-Controlador.	55
Figura 9. Diagrama de componentes del juego “Parchís”.....	59
Figura 10. Diagrama de componentes del juego “Localiza y Aprende”.....	59
Figura 11. Diagrama de secuencia para MundoClick online.....	60
Figura 12. Diagrama de secuencia para MundoClick offline.....	60

Índice de Tablas

Tabla 1. Actor de los juegos “Parchís” y “Localiza y Aprende”.	36
Tabla 2. Descripción textual del caso de uso: Realizar juego.....	39
Tabla 3. Descripción textual del caso de uso: Obtener Trofeo.	40
Tabla 4. Descripción textual del caso de uso: Otorgar bono.....	41
Tabla 5. Descripción textual del caso de uso: Mostrar resultado.....	42
Tabla 6. Descripción textual del caso de uso: Guardar estado.....	43
Tabla 7. Descripción textual del caso de uso: Realizar juego.....	46
Tabla 8. Descripción textual del caso de uso: Otorgar trofeo.	47
Tabla 9. Descripción textual del caso de uso: Otorgar bono.	48
Tabla 10. Descripción textual del caso de uso: Mostrar resultado.....	49
Tabla 11. Descripción textual del caso de uso: Guardar estado.....	50
Tabla 12. No conformidades encontradas en los sistemas en tres iteraciones.....	63

Introducción

El desarrollo acelerado de la ciencia y la tecnología en el campo de la Informática y las Comunicaciones (TIC) ha provocado que en la actualidad se implementen aplicaciones informáticas en todos los sectores de la sociedad. (1)

En el campo educativo se ha impulsado el uso de la tecnología en función de potenciar la calidad del proceso de enseñanza aprendizaje como la vía esencial para la formación de las nuevas generaciones, posibilitando la transmisión de conocimientos de una forma más amena, interactiva, y activa que el resto de los medios de enseñanza (1). Buscando nuevos medios de enseñanza que cumplan ambos requisitos(hagan uso de las tecnologías y faciliten el proceso de enseñanza aprendizaje) se ha comenzado a analizar los videojuegos como una poderosa herramienta para transmitir y fomentar el conocimiento, alcanzando hasta el momento resultados satisfactorios.

Los videojuegos ayudan a dinamizar las relaciones de grupo, y potenciar el trabajo participativo y de colaboración en todas las esferas activas de la vida social. Igualmente permiten introducir la reflexión acerca de ciertos valores y conductas a través de su contenido y de las consecuencias de las acciones que efectúan virtualmente. Logran fortalecer la experiencia del aprendizaje y acercarla al mundo adecuado y operativo en el que se mueve el hombre. (2)

Debido a estas potencialidades, Cuba, junto a la Universidad de las Ciencias Informáticas ha creado estrategias para la creación de proyectos que se dediquen a realizar aplicaciones de formación educativa. Un ejemplo de ello es el proyecto Multisaber-Navegante, perteneciente al Centro de Tecnologías para la Formación (FORTES). Este proyecto ha decidido realizar una colección de juegos educativos, denominada MundoClick, la cual contará con 27 juegos destinados a desarrollar diversas habilidades en los niños, en su mayoría en la edad comprendida entre los 6 y 8 años.

La madurez cerebral que se produce entre los 6 y los 8 años permite que los niños respondan a nuevas demandas académicas y personales. Sin embargo, no se debe olvidar que siguen siendo niños, el juego todavía es una necesidad para completar su desarrollo. Los niños de 6, 7 y 8 años muchas veces no entienden esa necesidad de centrarse en los deberes. Para ellos, el juego es una manera de aprender. (3)

Esta etapa es, además, importante a nivel de desarrollo intelectual por todas las capacidades de memoria que alcanzan. Con respecto a esta lo que cambia realmente es el uso que hace de ella,

puesto que a partir de los 6 años utiliza estrategias de almacenamiento como la estrategia de repaso y comienza a desarrollar el interés por adquirir nuevos conocimientos de una manera imperceptible para el niño.

A raíz de lo anteriormente planteado se define como **problema a resolver**: ¿Cómo potenciar la motivación hacia el estudio y el uso de la memoria en los niños de 6 a 8 años?

Se ha definido como **objeto de estudio** el aprendizaje basado en juegos para niños de 6 a 8 años. Enmarcándose en el **campo de acción**, videojuegos para niños de 6 a 8 años en Cuba. Todo ello encaminado a cumplir con el **objetivo general**: desarrollar dos juegos educativos que permitan la motivación hacia el estudio y desarrollen habilidades relacionadas con el uso de la memoria en niños de 6 a 8 años.

Para dar cumplimiento al objetivo general se plantean los siguientes **objetivos específicos**:

- ✓ Elaborar el marco teórico que servirá de sustento a la propuesta de solución.
- ✓ Diseñar los juegos “Parchís” y “Localiza y Aprende” para MundoClick.
- ✓ Implementar los juegos “Parchís” y “Localiza y Aprende” para MundoClick.
- ✓ Realizar las pruebas de software a los juegos “Parchís” y “Localiza y Aprende” para MundoClick.

Para guiar el cumplimiento de los objetivos propuestos se definen como **tareas de la investigación**:

- ✓ Realización de un estudio crítico valorativo de los conceptos, tecnologías, metodologías y herramientas relacionadas con la investigación para elaborar la fundamentación teórica.
- ✓ Identificación de los requisitos funcionales y no funcionales del sistema.
- ✓ Realización de la especificación de requisitos funcionales y no funcionales.
- ✓ Realización de los diagramas de clases del diseño y los diagramas de secuencia del diseño.
- ✓ Realización del diagrama de despliegue y el diagrama de componentes.
- ✓ Implementación de los requisitos funcionales.
- ✓ Integración de los juegos de “Parchís” y “Localiza y Aprende” a la plataforma Pointer.
- ✓ Realización de las pruebas al software.

Para dar solución a las tareas de investigación se emplearon métodos teóricos y empíricos:

Métodos Teóricos

Histórico-lógico: Este método fue de gran importancia en la elaboración de la fundamentación teórica de la investigación, mediante su estudio se logró realizar un análisis profundo de los antecedentes del tema: los videojuegos y los distintos tipos de géneros existentes, así como su aporte pedagógico y las habilidades que desarrollan en los niños de 6 a 8 años. Se indagó sobre los diferentes juegos educativos existentes a nivel nacional e internacional, las diferentes metodologías, herramientas y tecnologías que se utilizan en el desarrollo de la presente investigación.

Analítico-sintético: Este método fue utilizado durante todo el proceso investigativo, permitiendo realizar un análisis detallado de la documentación relacionada con el tema de los videojuegos, su aporte pedagógico en los niños de 6 a 8 años, las diferentes metodologías, herramientas y tecnologías utilizadas, así como, el Proceso Unificado de Desarrollo (RUP), para los flujos de trabajo: Análisis y Diseño e Implementación y Prueba. A partir de esto se sintetizaron las cuestiones más importantes y de mayor utilidad para el desarrollo del trabajo.

Métodos Empíricos:

Entrevista: Este método fue de gran importancia en la obtención de información necesaria para el cumplimiento de los objetivos.

Estructura capitular

Capítulo I: Fundamentación teórica. En esta sección se fundamentan los elementos teóricos necesarios que sustentan la investigación. Se realiza un análisis crítico del estado del arte en el tema de los videojuegos, los distintos tipos de géneros existentes, su aporte pedagógico y las habilidades que desarrollan en los niños de 6 a 8 años. Se hace un estudio de juegos similares existentes a nivel nacional e internacional y se fundamenta la utilización de las diferentes metodologías, herramientas y tecnologías para dar cumplimiento al objetivo general de la investigación.

Capítulo II: Características del sistema. Este capítulo engloba las principales características de los juegos y se identifican y describen los requisitos funcionales y no funcionales de los sistemas. Se muestra además, los actores y los diagramas de casos de usos, seguido de una descripción detallada de los mismos. Se describen los conceptos fundamentales, las relaciones y los principales atributos que se manejan mediante un Modelo Conceptual.

Capítulo III: Diseño del sistema. En este apartado se describen los aspectos fundamentales del modelo del diseño que sirven como base para el desarrollo de la solución propuesta y la obtención

práctica de la misma. Incluye la descripción y las ventajas del patrón arquitectónico utilizado. Contiene los diagramas de clases del diseño y los de secuencia. Finalmente se detalla el estándar de codificación a tener en cuenta en la implementación de los juegos.

Capítulo IV: Implementación y prueba: Comprende todo el proceso de implementación de clases y componentes encontrados durante el diseño. Incluye el diagrama de despliegue y de componentes, así como un estudio de los niveles de pruebas, los tipos de pruebas y los métodos empleados en la verificación de todos los requerimientos implementados.

Capítulo 1: Fundamentación teórica

1.1. Introducción

Actualmente los videojuegos constituyen una de las actividades de ocio o entretenimiento más populares. Su campo de acción, desde la segunda mitad de la década de los ochenta, se ha ampliado y ha sobrepasado la frontera del entretenimiento abriendo posibilidades de uso en el ámbito educativo (4). En este capítulo se propone un breve análisis del concepto de los videojuegos, los diferentes géneros existentes así como el aporte pedagógico en los niños, en aras de fortalecer el proceso de enseñanza-aprendizaje y en el desarrollo de la memoria y la motivación hacia el estudio. Se caracterizan, a partir de una búsqueda exhaustiva a nivel nacional e internacional, de los juegos “Parchís” y “Localiza y Aprende” existentes, además aquellos que utilizan preguntas de contenido educativo y que fomenten el desarrollo de habilidades en la ubicación de puntos en un eje de coordenadas. A su vez se realiza un estudio de las metodologías, herramientas y tecnologías utilizadas para dar cumplimiento al objetivo general de la investigación.

1.2. Los videojuegos

El concepto de videojuegos 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 pro del desarrollo del mismo. Estas aplicaciones desarrollan habilidades en los jugadores, como el razonamiento lógico y la planificación, aún cuando no tengan un contenido educativo explícito. (4)

La investigación y el interés por el aprendizaje basado en los videojuegos viene desarrollándose con fuerza desde el 2003, cuando James Gee comenzó a describir el impacto del juego en el desarrollo cognitivo. Desde entonces los juegos y sus plataformas se han diversificado y los desarrolladores e investigadores están trabajando en cada área del juego basado en el aprendizaje. (5)

Actualmente representan una herramienta pedagógica de muy alto valor, y esto se debe a que aparte de ser divertidos tienen la capacidad de inspirar y motivar. Pueden generar ideas, estimular la conversación, retar a la mente y animar a la resolución de problemas. (6)

1.3. Clasificación de los videojuegos

Desde el surgimiento de los juegos hasta la actualidad, se han ramificado varios géneros, cada uno con sus características propias, entre estos se encuentran:

Juegos de estrategia en tiempo real: Se basan fundamentalmente en la estrategia. Los jugadores controlan los aspectos económicos y militares de un ejército o de una población. Deben tomar decisiones estratégicas rápidas.

Juegos de rol: Los jugadores encarnan uno de los personajes de ficción. El personaje tiene varias características que pueden evolucionar a lo largo del juego, como salud, fuerza u otras habilidades, siendo un punto importante que es el jugador quien define cual parámetro evolucionar, así como diferenciándolo por el uso de ítems (armas, escudos, accesorios) que hacen que cada personaje del juego sea distinto a otro. En su mayoría están ambientados en entornos de fantasía y/o magia.

Aventuras gráficas: La jugabilidad (7) se basa en la narración de una historia. Los jugadores recorren mundos complejos, van recuperando objetos o ítems, como llaves y cofres, y afrontando desafíos que pueden incluir desde resolver una combinación de números hasta resolver un puzle con imágenes, hasta llegar al objetivo final.

Juegos de rol multijugador masivos en línea: Se trata de una variante de los juegos de rol con un gran número de participantes interactuando en un mundo virtual en línea. Este tipo de juegos constituye un elemento fundamental para las actividades colaborativas y de exploración, pues para ir explorando el mapa, o resolver determinadas misiones, el jugador debe buscar compañeros que lo ayuden, armando lo que se conoce en el juego como partys (grupo de jugadores que avanzan juntos por el mapa apoyándose entre sí).

Rompecabezas: Los jugadores tienen que resolver un rompecabezas para avanzar en el juego. Suelen ser juegos de pantalla estática. El Tetris es uno de los juegos de rompecabezas más conocido. Se basan fundamentalmente en la estrategia.

Juegos de deportes: Estos juegos reproducen deportes conocidos como el fútbol, el golf o el baloncesto. Están disponibles en 2D o 3D. Requieren coordinación y estrategia, en particular si el jugador tiene que administrar un equipo.

Laberintos: Los jugadores tienen que desplazarse en un laberinto perseguidos por enemigos que deben evitar. Necesitan estrategia, habilidades de planificación prospectiva y reflejos. (8)

Ludo-educativos: Son una combinación de actividades lúdicas con contenidos educativos. Muchas empresas consideran este género casi exclusivo de los ordenadores.

Juegos de sociedad: Son las versiones de los clásicos juegos de mesa o salón. Son casi inexistentes en las consolas pero tienen una regular presencia en los ordenadores. (4)

Los juegos “Parchís” y “Localiza y Aprende” son una combinación de los juegos de sociedad con los de tipo ludo-educativo pues presentan características de ambas clasificaciones (ambos juegos son adaptaciones de juegos de mesa o salón y además, tienen un contenido educativo implícito).

1.4. Aporte pedagógico de los juegos educativos

Son muchas las investigaciones que han ilustrado en las últimas dos décadas los beneficios pedagógicos de los juegos educativos. Éstos permiten el desarrollo de habilidades y destrezas, como son, el control psicomotriz, la coordinación ojo-mano, el desarrollo de la especialidad y de la capacidad deductiva, la resolución de problemas, la imaginación, el pensamiento (comprensión, reflexión, memorización, facultad de análisis y síntesis). (2) Ayudan, al mismo tiempo, a dinamizar la experiencia del aprendizaje y acercarla al mundo adecuado y operativo en el que se mueve el niño y/o adolescente.

La sensación de trabajar hacia una meta, la posibilidad de lograr éxitos espectaculares, la colaboración con los demás, la interacción social son características del juego que pueden ser transferidas al mundo de la educación. En este ámbito, el juego podría permitir que los niños adquieran las habilidades necesarias propias de una cultura basada en la información. La investigación y la experiencia han demostrado que los juegos se pueden aplicar de manera muy eficaz en muchos contextos de aprendizaje. (8)

1.5. Sistemas similares

Antes de iniciar el desarrollo de un proyecto de software es importante realizar un estudio detallado de los sistemas ya existentes tanto nacionales como internacionales. En la actualidad se pueden encontrar numerosos sistemas que brindan soluciones informáticas en el ámbito educativo con el objetivo de mejorar la calidad del proceso de enseñanza-aprendizaje. A continuación se muestran algunos juegos educativos existentes de “Parchís” y “Localiza y Aprende”, así como aquellos que utilizan preguntas de contenido educativo y que fomenten el desarrollo de habilidades en la ubicación de puntos en un eje de coordenadas.

POPI (Parchís Ortográfico para Primaria Interactivo): El sistema brinda, un divertido juego de parchís virtual, en el que pueden participar cuatro jugadores representados por un avatar. El juego se desarrolla sobre un tablero y para avanzar hacia la meta, además de lanzar los dados, los jugadores deberán resolver una actividad poniendo en acción sus conocimientos ortográficos.

Dentro de los aspectos negativos del juego se encuentra la repetitividad de los contenidos, lo que hace que los alumnos estén motivados a la hora de realizar la actividad por primera vez, pero al hacerlo a la quinta o sexta vez ya la motivación no es la misma, por lo que el aprendizaje deja de ser significativo. Además, el diseño gráfico no está muy trabajado, los recursos expresivos son utilizados solo cuando el jugador ha acertado a la respuesta del problema, cuando este se equivoca, solo sabe que falló porque el ordenador no le dice que ha acertado. (9)

También existe un juego publicado en el sitio <http://www.ctim.es>, llamado simplemente “**Parchís**”, esta versión es descargable pero está bajo licencia Attribution-NonCommercial-ShareAlike 3.0 Unported Licence, de Creative Commons, licencia que permite compartir y modificar el juego, pero siempre que sea con fines no comerciales y no admite la venta del mismo. Presenta variedad de temas modificables y se juega con un solo dado. Otras versiones de este juego publicadas en la red no presentan contenido educativo y se basan solamente en el clásico juego de tablero.

FyQ “Hundir la flota”: Es un juego inspirado en el clásico y famoso juego conocido como Batalla Naval. Consiste en conseguir “hundir” diez barcos que se encuentran ocultos en un panel. Para poder realizar los “disparos” se deberá contestar correctamente una serie de preguntas. Cuando se acierte una pregunta se podrá realizar un disparo introduciendo las coordenadas correspondientes. El juego está diseñado para dos jugadores (naranja y azul). Presenta dos niveles de dificultad, un nivel inicial básico y un nivel avanzado. Permite guardar y clasificar las preguntas y contenidos generados. (10)

El juego es gratuito, y se puede descargar o usar online. Está realizado con software libre y se usa a través de un navegador web (Internet Explorer, Firefox, Chrome) en Windows, Linux y Mac. Para utilizar el juego es necesario tener instalada la máquina virtual de java y el plugin Descartes web 2.0. También se recomienda tener instalados los plugins o visores para documentos flash y pdf.

Localiza y Aprende: Es un juego educativo cuyo objetivo es localizar desde continentes, hasta las regiones de los países en un mapa. Dispone de ayuda para facilitar el aprendizaje y hacerlo más ameno. Presenta cuestionarios geográficos. Solo puede jugarse en red y forma parte de una colección de juegos educativos.

El viaje de Ko: Diseñado por el profesor de matemáticas Scott Laidlaw. “El viaje de Ko” motiva a los estudiantes a aprender planos de coordenadas colocando problemas en el contexto de la narrativa del juego. Los estudiantes deben resolver problemas de matemáticas para ayudar a Ko a lo largo de su

viaje, trazando las coordenadas y navegando con las estrellas. Las tecnologías usadas para su desarrollo no son las requeridas por la plataforma.

Ubicación de los extraterrestres: Es un juego que pide a los jugadores localizar personajes extraterrestres en un plano de coordenadas. La historia del juego gira en torno a una conferencia intergaláctica de álgebra. Como miembro del comité de planeación, el jugador debe localizar a los extraterrestres que faltan, a tiempo para la conferencia. Los estudiantes deben trazar rápidamente varios pares ordenados en el transcurso de 90 segundos. Es soportado únicamente en plataformas de Windows (Windows XP/Vista/7/8). No se tiene acceso al código fuente ni se permite la modificación del mismo.

Atrapa a la mosca: Es un juego que pide a los jugadores cazar moscas a medida que aterrizan en un plano de coordenadas. Los jugadores deben introducir las coordenadas de la mosca antes de que se acabe el tiempo. Este juego proporciona a los jugadores ejercicios prácticos y esenciales para el dominio de las habilidades de los planos de coordenadas. El juego sigue la puntuación del jugador como un método para motivarlo a mejorar sus habilidades. Cuanto más el jugador contesta cada problema correctamente, mayor será la puntuación. Tiene como característica que no permite la descarga del juego, está realizado con software propietario y no permite su uso con fines comerciales.
(11)

Juego de batalla: El juego propone un conflicto bélico que tiene lugar sobre un planisferio dividido en cincuenta países. Para empezar se reparten los cincuenta países entre los jugadores, quienes ocupan sus dominios con fichas (ejércitos). Cada jugador tiene un objetivo secreto a cumplir, que se le asigna por azar y que los jugadores desconocen. Para cumplir con su objetivo (ocupar cierta cantidad de países o destruir a otros bandos) el jugador deberá ampliar sus dominios y reordenar sus fuerzas, lo cual exigirá emprender y defenderse de ataques adversarios; agrupar y desplazar ejércitos; y concertar pactos con otros contendientes. La conquista de nuevos territorios le permitirá aumentar el número de ejércitos a su disposición, otorgándole mayores oportunidades para triunfar en los combates. La dinámica de los combates se basa en el tiro de los dados, pero toma en cuenta la cantidad de ejércitos que lo defienden. El jugador que primero cumpla su objetivo secreto, gana el juego.

Luego de analizar algunos productos existentes se arribó a las siguientes conclusiones.

- ✓ Algunos de los sistemas antes mencionados están desarrollados sobre software propietario.

- ✓ Otros están bajo licencia, no se tiene acceso al código fuente y son soportados únicamente en plataformas Windows.
- ✓ No usan tecnologías utilizadas por el proyecto.

Debido a lo anteriormente planteado se puede concluir que los juegos analizados presentan funcionalidades que pueden usarse como una línea base para el desarrollo de los presentes sistemas, pero no constituyen por sí solos una posible solución ya que por sus características no permiten la integración con la plataforma Pointer, ni permiten el uso de los mismos con fines comerciales.

1.6. Plataforma Pointer

Plataforma de desarrollo orientada a la implementación y despliegue de juegos basados en HTML5, CCS3 y JavaScript, tiene como objetivo principal apoyar el desarrollo de los juegos educativos creando una abstracción de varias funcionalidades comunes entre ellos, lo cual permitirá acortar el tiempo de desarrollo, estandarizando gran parte del código y ofreciendo un ambiente controlado para el funcionamiento de los mismos.

Las tecnologías que utiliza esta plataforma para su correcto funcionamiento son:

- ✓ AppJS.
- ✓ NodeJS v0.8.8.
- ✓ JQuery v1.9.
- ✓ HTML5.
- ✓ CSS 3.
- ✓ JavaScript.

Entre las principales funcionalidades propuestas para la versión 0.5 están:

- ✓ Gestión de los jugadores (registro, sistema de trazas).
- ✓ Control de audio y video.
- ✓ Gestión de configuración.
- ✓ Gestión plug and play de los juegos.

1.7. Metodología a utilizar

Hacer un software resulta algo complejo, para ello se requiere de una guía o un plan de trabajo que el equipo de desarrollo se debe trazar o planificar, de tal forma que el producto o resultado del trabajo tenga éxito, o sea que se cumplan los objetivos y los requerimientos, se entregue en tiempo, forma y con la calidad que requiere o que el cliente exija. Para lograr cumplir todos los objetivos con una mayor facilidad de trabajo, existen las metodologías de desarrollo.

Una metodología, en ingeniería de software, es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. Puede seguir uno o varios modelos de ciclo de vida del sistema, y es este ciclo de vida el que indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto, pero no cómo hacerlo. La metodología indica cómo hay que obtener los distintos productos parciales y finales. Debe ajustarse a los objetivos y cubrir el ciclo entero de desarrollo de software. (12)

1.7.1. Metodología RUP

Esta metodología fue definida por el proyecto en que se enmarca la solución, por tanto es la que se usará en el desarrollo de los juegos.

El **Proceso Unificado de Desarrollo (RUP)** constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas de software. RUP define claramente quién está haciendo qué, cuándo lo hace y cómo alcanzar un determinado objetivo. Divide el proceso de desarrollo en fases, obteniendo una representación del producto al final de cada una:

- ✓ **Inicio:** Se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.
- ✓ **Elaboración:** Se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- ✓ **Construcción:** Se enfoca en la elaboración de un producto completamente operativo y eficiente.
- ✓ **Transición:** Se instala el producto y se capacita al usuario. Como consecuencia de esto suelen surgir nuevos requisitos al ser analizados.

El proceso de desarrollo de software propuesto por RUP tiene tres características esenciales:

- ✓ **Guiado por casos de uso:** Un caso de uso es una descripción de una secuencia de interacciones de un sistema con una persona u otro sistema que usa alguno de sus servicios.

Los casos de uso responden a un proceso o parte de un proceso dentro del negocio, guiando de esta manera el ciclo de desarrollo del software.

- ✓ **Iterativo e incremental:** Se dice que un proceso es iterativo incremental cuando en cada iteración se alcanza una mayor cercanía a los objetivos finales, añadiendo algo nuevo en cada una de ellas.
- ✓ **Centrado en la arquitectura:** La arquitectura es fundamentalmente una actividad centrada en el diseño, que muestra una visión común del sistema completo, por lo que repercute en todo el ciclo de vida de desarrollo del software. Las decisiones de la arquitectura se concentran en lo que es esencial en un sistema: la influencia de los requerimientos y las actividades importantes como el diseño colaborativo, el desarrollo e implementación del sistema, la planeación de su evolución y su adaptación.

Brinda una guía para encontrar, organizar, documentar, y seguir los cambios de los requisitos funcionales y restricciones. Utiliza una notación de casos de uso y escenarios para representar los requisitos.

1.7.2. Lenguaje Unificado de Modelado (UML)

RUP propone como lenguaje de modelado a **UML** (Lenguaje Unificado de Modelado) para visualizar, especificar, construir y documentar los artefactos de un software. El modelado visual ayuda a mantener la consistencia entre los artefactos del sistema: requisitos, diseños e implementaciones, por tanto, ayuda a mejorar la capacidad del equipo para gestionar la complejidad del software.

UML utiliza diagramas y una semántica bien definida con el propósito de elaborar los artefactos de un sistema a través de las distintas etapas de su ciclo de vida, principalmente durante el análisis y el diseño del mismo. Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso utilizar.

Este lenguaje de modelado formal permite tener un mayor rigor en la especificación, realizar una verificación y validación del modelo desarrollado, automatizar determinados procesos y generar código a partir de los modelos y a la inversa. Esto último permite que el modelo y el código estén actualizados.

(13)

1.8. Herramienta de modelado: Visual Paradigm 8.0

Esta herramienta será la utilizada debido a que fue definida por el proyecto en que se enmarca la investigación.

Visual Paradigm es una herramienta CASE (Ingeniería de Software Asistida por Computación). La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta privada disponible en varias ediciones. (14)

Se caracteriza por:

- ✓ Disponibilidad en múltiples plataformas (Windows, Linux).
- ✓ Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- ✓ Disponibilidad de múltiples versiones, para cada necesidad.
- ✓ Varios idiomas, entre los que incluye el español y el inglés.
- ✓ Su instalación y actualización no requiere de grandes conocimientos.
- ✓ Compatibilidad entre ediciones.
- ✓ Soporte de UML versión 2.1.
- ✓ Diagramas de flujo de datos.
- ✓ Generador de informes.

1.9. Lenguajes de programación

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; además se pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes. (15)

El uso de la plataforma Pointer en el desarrollo de los juegos condiciona el uso de determinados lenguajes los cuales se describen a continuación.

1.9.1 HTML 5

El HTML 5, HyperText Markup Language, (Lenguaje de Marcado Hipertextual versión 5) es la quinta revisión del lenguaje de programación “básico” de la World Wide Web, el HTML. Esta nueva versión pretende reemplazar al actual HTML, corrigiendo problemas con los que los desarrolladores web se encuentran, así como rediseñar el código actualizándolo a nuevas necesidades que demanda la web de hoy en día.

A diferencia de otras versiones de HTML, los cambios en HTML5 comienzan añadiendo semántica y accesibilidad implícitas, especificando cada detalle y borrando cualquier ambigüedad. (16)

Entre sus nuevas facilidades se encuentra el manejo de medias, la estructura de formulario y el bloqueo de contenido en las páginas.

1.9.2 CSS 3

CSS 3 es usado para definir la presentación de un documento escrito en HTML o XML. Los primeros borradores de esta revisión del popular lenguaje de hojas de estilo en cascada se publicaron en 1999 pero se comenzó a desarrollar en el 2004. La principal diferencia con otras versiones es que CSS 3 está integrado por documentos separados llamados módulos, cada uno aportando nuevas funcionalidades y preservando las anteriores para mejor compatibilidad.

Permite además webs más elaboradas y más dinámicas, con mayor separación entre estilos y contenidos. Brinda soporte a muchas necesidades de las webs actuales, sin tener que recurrir a trucos de diseñadores o lenguajes de programación. (17)

Algunas de las nuevas características de CSS 3 son:

- ✓ Incluye bordes redondeados, que define la curvatura que debe tener las esquinas de los elementos.
- ✓ Permite que un elemento de la página tenga varias imágenes de fondo a la vez.
- ✓ Reacomoda las palabras que son demasiado largas y no caben enteras dentro de un elemento.
- ✓ Animaciones de transición y transformación.

1.9.3 JavaScript

Lenguaje de programación interpretado por un explorador web, simple, liviano, basado en Java, dinámico pero principalmente muy útil y funcional. Fue desarrollado originalmente por BrendanEich con

el nombre de Mocha y renombrado a LiveScript, finalmente terminó bautizado como JavaScript, coincidentemente con el momento en que Netscape agregó soporte para la tecnología Java en su navegador Netscape Navigator 2.0B3 en diciembre de 1995, y causando cierta confusión al dar la impresión que este lenguaje es una prolongación de Java.

Este lenguaje es utilizado en la creación de páginas web dinámicas, como complemento a HTML y CSS. El código no es compilado sino que es interpretado por el navegador en tiempo real y de forma dinámica. Como tal no es un lenguaje de programación propiamente dicho, sino un lenguaje script u orientado a documentos.

Su sintaxis es muy similar a la de Java, pero no es un lenguaje orientado a objetos, sino basado en estos. Al ser de tipo script no puede escribir directamente en el disco duro. Puede ser incluido en las páginas web escribiendo el código antes del HTML o en algunos tags como las imágenes y los hipervínculos, o escribiéndose en un archivo aparte. JavaScript es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Mozilla Firefox, Chrome y Opera. (18)

1.10. Frameworks

Los frameworks se utilizan en el ámbito de la programación de aplicaciones desde hace décadas. Genéricamente, es un conjunto de herramientas, librerías, convenciones y buenas prácticas que pretenden encapsular las tareas repetitivas en módulos genéricos fácilmente reutilizables.

El uso de la plataforma Pointer para desarrollar los juegos, condiciona el uso de determinados frameworks los cuales se describen a continuación.

1.10.1. Node.js v0.8.8

Node.js es un entorno de programación en la capa del servidor basado en el lenguaje de programación JavaScript, con entrada/salida de datos en una arquitectura orientada a eventos, y basado en el motor JavaScript. Fue creado por Ryan Dahl en 2009, y su evolución está apadrinada por la empresa Joyent.

Node.js es actualmente una buena herramienta para generar sistemas de alto rendimiento, y gracias a la versatilidad y poder de JavaScript, se ha convertido en una herramienta popular para el desarrollo de servicios de tiempo real. (19)

Es OpenSource(código abierto), y facilita el trabajo con los códigos de cliente y servidor al trabajar ambos sobre un mismo lenguaje, y creando un solo hilo de procesos lo cual permite al servidor

soportar más conexiones.

1.10.2. jQuery v1.9

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. (20)

Este potente framework permite el acceso a cualquier elemento del DOM a través de selectores CSS, lo cual, en el caso de los juegos a desarrollar en esta investigación es especialmente útil pues permite trabajar las animaciones y transiciones de imágenes de una forma bastante sencilla.

1.11. Entorno de Desarrollo Integrado(IDE)

Un IDE es una aplicación informática compuesto por un conjunto de herramientas para programar en un lenguaje o varios, donde se puede encontrar como mínimo un editor, compilador, intérprete y depurador de uno o varios lenguajes de programación. (21)

Las herramientas que normalmente componen un entorno de desarrollo integrado son: herramientas para la automatización, un sistema de ayuda para la construcción de interfaces gráficas de usuario, opcionalmente, un sistema de control de versiones, y las mencionadas anteriormente.

Hoy en día los entornos de desarrollo proporcionan un marco de trabajo para la mayoría de los lenguajes de programación existentes en el mercado (por ejemplo C, C++, C#, Java, Python y Visual Basic). Además es posible que un mismo entorno de desarrollo tenga la posibilidad de utilizar varios lenguajes de programación, como es el caso de NetBeans. (22)

1.11.1. NetBeans 7.1.2

NetBeans IDE es una aplicación de código abierto (open source) diseñada para el desarrollo de aplicaciones fácilmente portables entre las distintas plataformas, haciendo uso de la tecnología Java. Dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, así como colaboración entre varias personas. (23)

Entre los cambios implementados en NetBeans IDE 7.1.2 se destaca soporte para JavaFX 2.0, con lo que es posible llevar a cabo los procesos completos de compilación y depuración de aplicaciones desarrolladas en esta plataforma, además soporte para CSS 3, mejoras en el debug de PHP o integración con Subversion.

Fue seleccionado por el proyecto Multisaber-El Navegante debido a que NetBeans 7.1.2 posee un avanzado editor de código, compatible con multitud de tecnologías (AJAX, JSP, EJB) y, dentro de un mismo entorno, ofrece análisis del código, compilación, ejecución, debugging, control de versiones, detección de duplicaciones, análisis de dependencias y soporte para plugins.

1.12. Control de versiones: Subversion

Subversion es un controlador de versiones de software libre que mantiene los registros de todos los cambios que se han realizado a los archivos de un software, lo que permite el trabajo de distintos desarrolladores en un mismo proyecto, esta herramienta es usada por los programadores de software.

Es uno de los sistemas de control de versiones más modernos, utiliza un sistema con repositorio centralizado y fue diseñado como reemplazo del sistema más utilizado hasta la fecha, CVS (del inglés: Concurrent Versions System). Sus principales características están relacionadas con gestionar las modificaciones durante el desarrollo, mediante el seguimiento del historial de los archivos y directorios a través de copias y renombrados, además sus modificaciones son automáticas por lo que permite que varias personas trabajen sobre los mismos ficheros enviando solo las diferencias en ambas direcciones. (24)

1.13. Conclusiones parciales

Como resultado de la investigación realizada en este capítulo se puede concluir que:

Se definieron los videojuegos educativos como una poderosa herramienta para el proceso de enseñanza aprendizaje de las nuevas generaciones. Además la investigación exhaustiva de sistemas similares a los juegos “Parchís” y “Localiza y Aprende” demostró que los mismos no cumplen con los requisitos necesarios para darle solución al problema planteado, sin embargo, poseen una serie de funcionalidades que sí permiten tener una línea base de desarrollo para la realización del presente sistema. El análisis de las herramientas, metodologías y tecnologías indicó que las seleccionadas por el equipo de desarrollo del proyecto Multisaber-Navegante son las más idóneas para el desarrollo de la propuesta de solución.

Capítulo 2: Características del sistema

2.1. Introducción

Un sistema, por pequeño que sea, generalmente es complicado. Por eso se necesita dividirlo en piezas si se pretende comprender y gestionar su complejidad. Esas piezas se pueden representar a través de modelos que permitan abstraer sus características esenciales. (25) Es por ello que en el presente capítulo se describen los conceptos fundamentales, las relaciones y los principales atributos que se manejan mediante un Modelo Conceptual. Se definen los requisitos funcionales y no funcionales y se representan los casos de uso, los actores, y sus relaciones mediante los diagramas de casos de uso.

2.2. Descripción general de la propuesta del sistema

Se propone el desarrollo de dos sistemas que permitan el aprendizaje de los niños de 6 a 8 años logrando que adquieran y comprueben conocimientos de una forma atractiva y divertida. A continuación se realiza una descripción del funcionamiento de cada una de las aplicaciones a desarrollar para la solución.

Juego “Parchís”: El juego educativo “**Parchís**” está estructurado por cuatro niveles los cuales para completarlos hay que llegar a la meta, al vencer todos los niveles se gana el juego. En cada nivel cada jugador dispone de 4 fichas de un mismo color (amarillas, rojas, verdes o azules), una zona de inicio llamada Casa o Base y una zona final llamada Meta.

El objetivo del juego es conseguir llevar antes que los contrarios las 4 fichas desde la Casa hasta la casilla de Meta recorriendo todo el tablero. Todos inician el juego con las fichas en sus respectivas Casas, antes de iniciar la partida cada jugador lanza los dados, y solo podrán sacar una ficha cuando obtengan un 5. Antes de caminar una ficha a cada jugador le aparecerá una pregunta, la cual es de carácter obligatorio responderla y a medida que acierte ganará puntos e irá avanzando hacia la Meta. Si no acierta, la ficha no avanzará a la casilla de la jugada. Al responder correctamente y de forma consecutiva cierta cantidad de preguntas se obtienen trofeos.

De forma aleatoria aparecen bonos en algunas casillas, de forma tal que la ficha que llegue a esa casilla pueda obtenerlo. Al terminar la partida se muestra la información de los resultados del juego, el nombre del jugador ganador con la cantidad de preguntas respondidas correcta e incorrectamente, los puntos, bonos y trofeos obtenidos. Se cuenta con una serie de instrucciones a lo largo del juego para que los usuarios se informen sobre las instrucciones y las reglas del mismo. (Ver [Anexo 1](#))

Juego “Localiza y Aprende”: El juego consiste en una batalla naval donde cada jugador representa al capitán de su flota, éste tiene la misión de llevarla a puerto seguro, pero para ello, tiene que destruir las flotas enemigas que intenten destruirlos. Está estructurado por cinco niveles, diferenciados por la cantidad y los tipos de barcos. Para completar cada nivel hay que vencer al contrario. El juego cuenta con cinco tipos de barcos: lancha, fragata, buque, submarino y porta avión.

EL juego se inicia cuando el usuario, de acuerdo al nivel en que se encuentra, ubica los diferentes barcos en el eje de coordenadas. Posteriormente lanza los dados, donde cada uno de ellos representa una coordenada (x, y). Si la posición correspondiente al resultado de los dados tiene un barco, se levanta una pregunta. Si el usuario la contesta correctamente el disparo será certero de lo contrario será fallido. Luego el jugador ubica el puntero en el área correspondiente al valor de los dados. Algunos barcos tienen asociados bonos que se activan cuando se destruye la embarcación.

Al terminar la partida aparece la información de los resultados del juego, mostrando el nombre de cada jugador con la cantidad de preguntas respondidas correcta e incorrectamente, los puntos, bonos y trofeos obtenidos. Se cuenta con unas instrucciones que se muestran al inicio del juego para que el usuario conozca el funcionamiento y las reglas del juego. (Ver [Anexo 2](#))

2.3. Especificación de los requisitos de software

Según la IEEE, Standard Glossary of Software Engineering Terminology, (Glosario Estándar de la IEEE de Terminologías de Ingeniería del Software) un requerimiento es una condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. (26)

Son definidos como condiciones o capacidades que deben estar presentes en un sistema o componentes de éste, para satisfacer un contrato, estándar, especificación u otro documento formal. Se pueden clasificar en: funcionales y no funcionales.

2.3.1. Requisitos funcionales

Los requerimientos funcionales (RF) son capacidades o condiciones que el sistema debe cumplir. Estos se deben mantener invariables sin importar con qué propiedades o cualidades se relacionen. (25)

Requisitos funcionales del juego “Parchís”:

RF1: Mostrar pantalla principal del juego.

- ✓ RF1.1- Seleccionar tablero.

RF2: Mostrar la información del juego.

- ✓ RF2.1.- Mostrar instrucciones generales del juego.
- ✓ RF2.2.- Mostrar mensajes de orientación.
- ✓ RF2.3 - Mostrar área de jugador en turno.
- ✓ RF2.4.- Mostrar casillas con posibles jugadas.
- ✓ RF2.5 -Terminar juego.

RF3: Permitir movimiento.

- ✓ RF3.1 - Mover los dados.
- ✓ RF3.2 - Mover la ficha.

RF4: Realizar ejercicio.

- ✓ RF4.1 - Mostrar pregunta.
- ✓ RF4.2 - Seleccionar respuesta.
- ✓ RF4.3 - Evaluar respuesta.
- ✓ RF4.4 - Registrar respuestas correctas.
- ✓ RF4.5 - Registrar respuestas incorrectas.
- ✓ RF4.6 - Incrementar puntuación.
- ✓ RF4.6 - Disminuir puntuación.
- ✓ RF4.7- Mostrar puntuación obtenida en el ejercicio.

RF5: Otorgar premio.

- ✓ RF5.1 - Mostrar nombre del trofeo.
- ✓ RF5.2 - Otorgar trofeo.
- ✓ RF5.3 - Otorgar bono.

RF6: Mostrar resultados.

- ✓ RF6.1 - Mostrar nombre del jugador ganador.
- ✓ RF6.2 - Mostrar cantidad de preguntas respondidas correctamente.
- ✓ RF6.3 - Mostrar cantidad de preguntas respondidas incorrectamente.
- ✓ RF6.4 - Mostrar puntos obtenidos por el jugador en el juego.

RF7: Guardar el estado del juego.

Requisitos funcionales del juego “Localiza y Aprende”:

RF1. Mostrar pantalla principal del juego.

RF2. Mostrar la información del juego.

- ✓ RF2.1 - Mostrar instrucciones generales del juego.
- ✓ RF2.2 - Terminar juego.

RF3. Permitir movimiento.

- ✓ RF3.1 - Mover dados.
- ✓ RF3.2 - Activar mirilla.
- ✓ RF3.3 - Mover mirilla.
- ✓ RF3.4 - Rotar barco.

RF4. Gestionar información del barco.

- ✓ RF4.1 - Cargar barcos en el eje de coordenadas.
- ✓ RF4.2 - Ubicar barcos.
- ✓ RF4.3 - Restar vida a los barcos.
- ✓ RF4.4 - Eliminar barcos.

RF5. Realizar ejercicio.

- ✓ RF5.1 - Mostrar ejercicio.
- ✓ RF5.2 - Seleccionar respuesta.
- ✓ RF5.3 - Evaluar respuesta.
- ✓ RF5.4 - Registrar respuestas correctas.
- ✓ RF5.5 - Registrar respuestas incorrectas.

RF6. Otorgar premio.

- ✓ RF6.1 - Otorgar trofeo.
- ✓ RF6.2 - Otorgar bono.

RF7. Mostrar resultados del juego.

- ✓ RF7.1 - Mostrar nombre del jugador ganador.
- ✓ RF7.2 - Mostrar puntos obtenidos por el jugador.
- ✓ RF7.3 - Calcular efectividad.
- ✓ RF7.4 - Mostrar efectividad.

RF8. Guardar el estado del juego.

2.3.2. Requisitos no funcionales

Los requerimientos no funcionales son cualidades o propiedades que el producto debe poseer, o restricciones en el producto que está siendo desarrollado. Estos no describen lo que el software hará, sino cómo lo hará. Estos requisitos son de gran significación en la aceptación del software, debido a que representan las ventajas más visibles al usuario y repercuten en el óptimo funcionamiento y mantenimiento del sistema. (26)

RNF1. Usabilidad

- ✓ Diseño de interfaces intuitivas.
- ✓ Mantendrán siempre visible la opción de Ayuda, lo que posibilitará a los usuarios una mejor comprensión de las funcionalidades.
- ✓ Todo el texto de los mensajes y preguntas que aparezcan en las aplicaciones deberá estar en idioma castellano (español) y expresados de forma clara y concisa.

RNF2. Confiabilidad.

- ✓ Debe informarse al usuario cómo guardar el estado de los juegos.
- ✓ Los sistemas deben estar libres de fallas, en caso de que existieran, el tiempo medio de reparación debe ser menor de 24 horas.
- ✓ Deben montarse sistemas de respaldo eléctrico en los locales de los servidores para mantener activos los servicios.

RNF3. Rendimiento.

- ✓ Los sistemas deberán atender y responder a una determinada petición por parte de los usuarios en un período de tiempo no mayor de 5 segundos.

RNF4. Soporte.

- ✓ El administrador se encargará de la instalación, configuración y puesta a prueba de las aplicaciones; además de ofrecer mantenimiento y evaluar en la práctica las funcionalidades.

RNF5. Portabilidad.

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

RNF6. Seguridad.

- ✓ Integridad: Sólo los administradores podrán modificar los contenidos educativos que serán presentados en los juegos.

RNF7. Interfaces de usuario.

- ✓ Los sistemas deberán poseer una interfaz externa sencilla, amigable, lo más clara e intuitiva posible para los usuarios, incluso para aquellos que no han tenido mucha experiencia en la interacción con computadoras o con sistemas informáticos.
- ✓ La información que se mostrará en las interfaces debe estar bien detallada y organizada.
- ✓ Se debe contar con un formato y una combinación de colores parejos para cada una de las interfaces.
- ✓ Las imágenes presentadas en cada interfaz deben estar acorde al contenido de cada nivel.
- ✓ Todos los textos y mensajes en pantalla aparecerán en idioma español.

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

RNF8. Políticos-culturales.

- ✓ Sólo deben contener palabras en idioma español, exceptuando términos propios de la especialidad o contenido de un nivel determinado.

RNF9. Legales.

- ✓ Los sistemas se deben ajustar y regir por las leyes, decretos, decretos leyes, resoluciones y manuales establecidos, que norman los procesos que serán automatizados.
- ✓ Las herramientas escogidas para el desarrollo de las aplicaciones, están basadas en licencias de software libre.

RNF10. Requisitos de hardware.

- ✓ Procesador Pentium 233 MHz (recomendado 500 MHz o superior).
- ✓ 64 MB de RAM (recomendado 512 MB de RAM o superior).
- ✓ 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.

RNF11. Restricciones en el diseño y la implementación.

- ✓ JavaScript.
- ✓ JQuery v1.9.
- ✓ Node.Js v0.8.8.
- ✓ Socket.IO v 0.9.10.
- ✓ NetBeans v7.1.2.

2.4. Modelo conceptual

Un modelo conceptual explica los conceptos más significativos en un dominio del problema, identificando los atributos y las asociaciones, y es la herramienta más importante del análisis orientado a objetos. Un modelo conceptual representa elementos del mundo real, no componentes del software. (27)

En estos diagramas se muestran conceptos (objetos), asociaciones entre conceptos (relaciones) y atributos de conceptos (atributos). A continuación se muestra el modelo conceptual generado. (Figura 1).

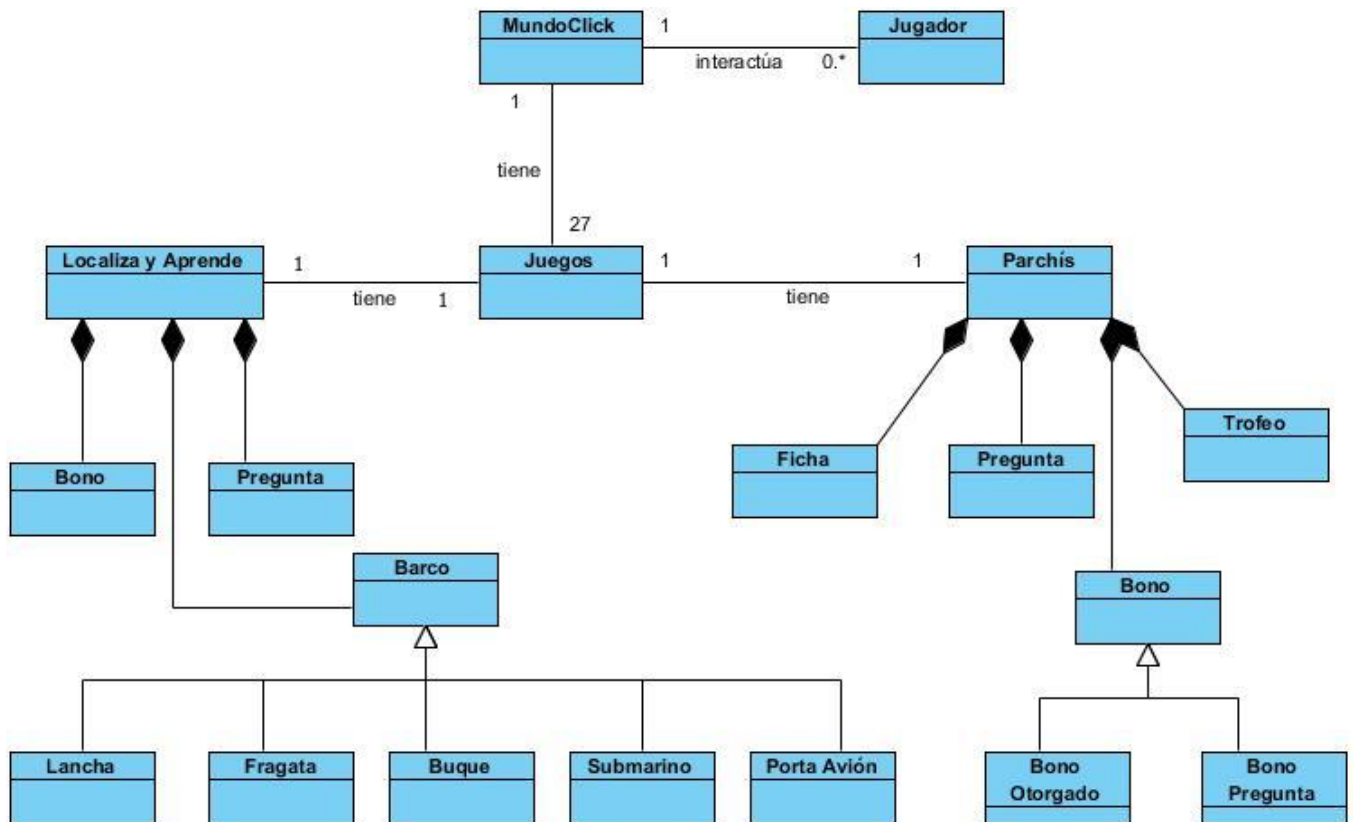


Figura 1. Modelo conceptual.

2.4.1. Descripción de los conceptos

Para una mejor comprensión del modelo conceptual se hace necesario describir los principales conceptos mostrados anteriormente.

MundoClick: Colección de juegos educativos dirigido a niños y jóvenes. Se incluyen un total de 27 juegos, que fomentan y ejercitan una serie de habilidades como el uso del mouse y el teclado, la memoria y la agilidad mental.

Pointer: Plataforma de desarrollo orientada a la implementación y despliegue de juegos basados en HTML 5, CSS 3 y JavaScript. Tiene como objetivo principal apoyar el desarrollo de los juegos creando una abstracción de varias funcionalidades comunes entre ellos.

Juego “Localiza y Aprende”: El juego consiste en una batalla naval donde cada jugador representa al capitán de su flota, este tiene la misión de llevarla a puerto seguro, pero para ello, tiene que hundir las flotas enemigas que intenten destruirlos.

Juego “Parchís”: Juego donde cada jugador dispone de 4 fichas de un mismo color (amarillas, rojas,

verdes o azules) y de una zona de inicio llamada casa en el tablero. El objetivo del juego es conseguir llevar antes que los contrarios, las 4 fichas desde la casa hasta la casilla de meta recorriendo todo el tablero.

Ficha: Fichas que serán usadas por los jugadores para avanzar por el tablero, Se le asignan 4 a cada jugador.

Bono: Se ubicarán aleatoriamente en el mapa contrario como un elemento oculto, o estarán asociados a los barcos destruidos.

Bono Otorgado: Bono que se le otorga al jugador cuando la ficha cae en la casilla en la que se encuentra el bono.

Bono Pregunta: Bono que se le otorga al jugador cuando responde una determinada cantidad de preguntas.

Pregunta: Pregunta que el jugador está obligado a responder.

Jugador: Usuario que interactúa con el juego.

Trofeo: Premio que será otorgado al jugador al responder una determinada cantidad de preguntas correctamente y de forma consecutiva.

Barco: Barcos que serán usados por el jugador.

Lancha, Fragata, Buque, Submarino, Porta avión: Tipos de barcos que serán utilizados en el juego.

2.5. Definición de Casos de Uso

Los requisitos o capacidades internas del sistema a desarrollar han sido expresados como casos de uso teniendo en cuenta que estos son actividades atómicas que ofrecen un resultado definido para cada acción que realiza el usuario.

El modelo de casos de uso del sistema documenta el comportamiento del sistema desde el punto de vista del usuario, permitiendo representar las funciones que se desean en el sistema (casos de uso), el entorno del sistema (actores), y las relaciones entre ellos. Aunque la parte más visible de dicho modelo son los diagramas de casos de uso, suele ir acompañado de una especificación textual de cada uno de ellos.

Los actores del sistema no forman parte del sistema, sino que representan elementos que interactúan con él. Estos elementos son nombrados roles que puede estar formado por una o varias personas, un equipo o un sistema automatizado. Un actor puede introducir o recibir información del sistema. (28)

A continuación se muestra la tabla con el actor de los juegos y la descripción del mismo (**Tabla1**).

Actor	Descripción
Usuario	Representa al usuario que interactúa con los juegos “Parchís” y “Localiza y Aprende”.

Tabla 1. Actor de los juegos “Parchís” y “Localiza y Aprende”.

2.5.1. Diagramas de Casos de Uso

A continuación se presenta el diagrama de casos de uso de los juegos “Parchís” y “Localiza y Aprende”. (Figura 2 y Figura 3 respectivamente).

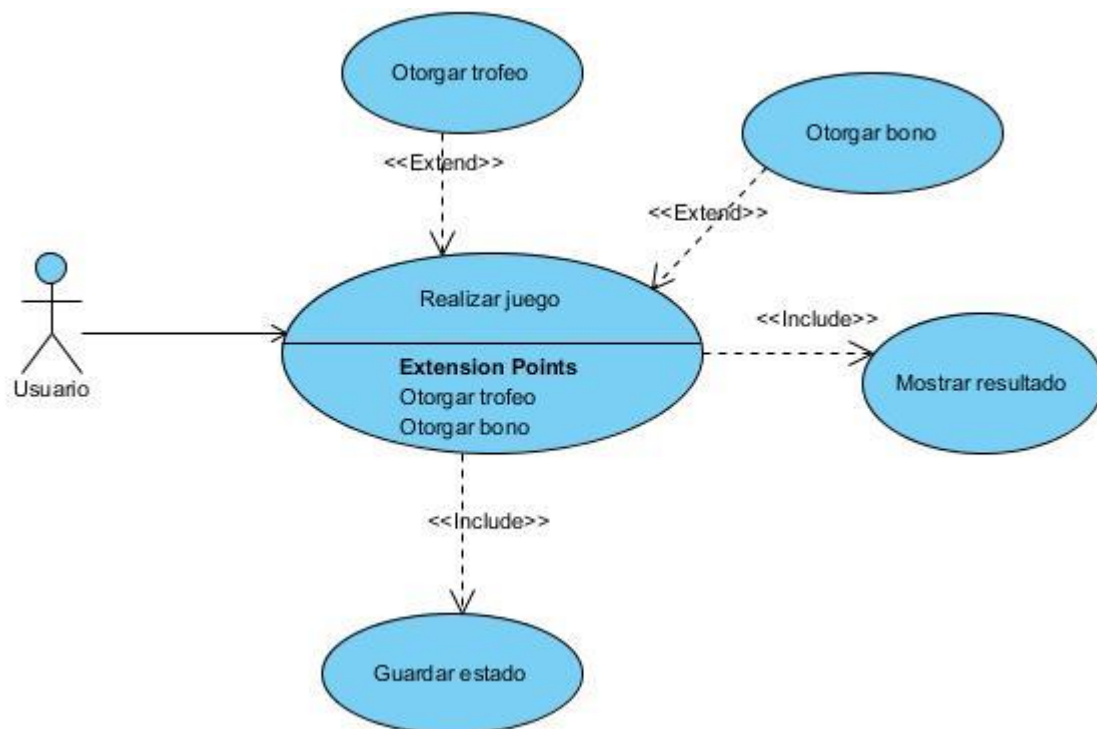


Figura 2. Diagrama de Casos de Uso del juego “Parchís”.

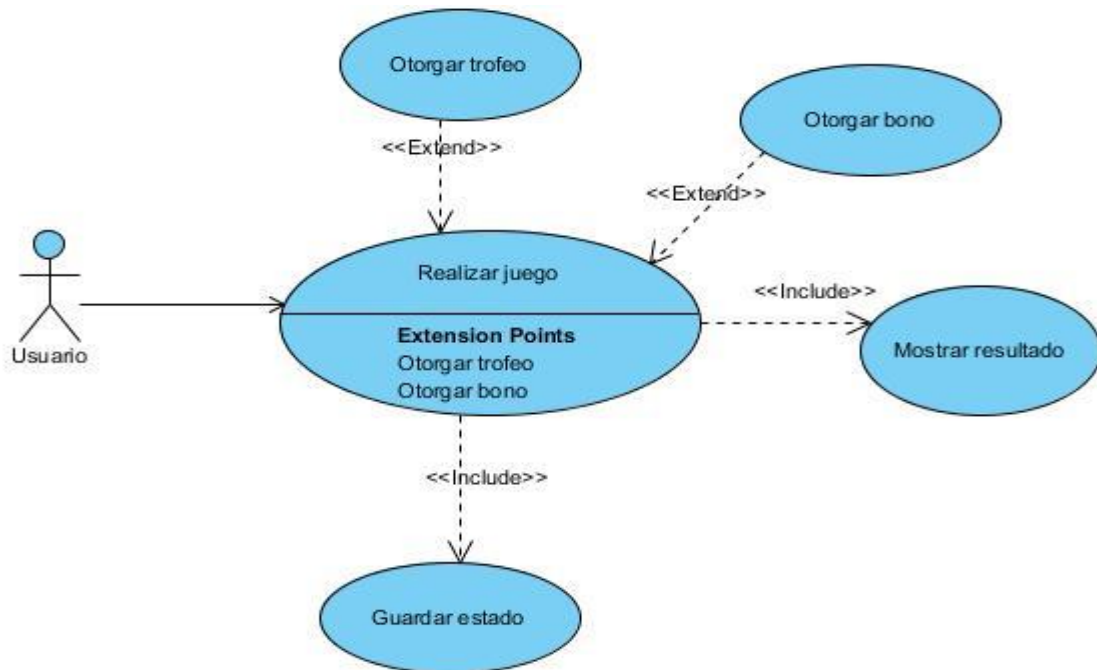


Figura 3. Diagrama de Casos de Uso del juego “Localiza y Aprende”.

2.5.2. Descripción textual de los casos de uso

La descripción textual de un caso de uso es una especificación simple y consistente de cómo interactúan los actores y los casos de uso. A continuación se muestran las descripciones de los casos de uso para los juegos “Parchís” y “Localiza y Aprende”.

Descripción de casos de uso del juego “Parchís”.

Caso de uso: Realizar juego

Objetivo	Permitir al usuario interactuar con el juego.
Actores	Usuario
Resumen	El caso de uso inicia cuando el usuario accede a la interfaz principal. El sistema muestra el tablero seleccionado y permite la interacción con el juego. El usuario realiza las actividades del juego, termina el caso de uso.
Complejidad	Alta

Prioridad	Crítico	
Precondiciones	Usuario autenticado en la plataforma para acceder al módulo "Parchís".	
Postcondiciones	Juego terminado.	
Referencias	RF1, RF1.1, RF2, RF2.2, RF2.3, RF2.4, RF2.5, RF3, RF3.1, RF3.2, RF4.1, RF4.2, RF4.3, RF4.4, RF4.5, RF4.6, RF4.7, RF4.8, RF6, RF7.	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1- Selecciona en la plataforma el juego "Parchís".	2- Ejecuta el juego, mostrando la interfaz principal. 3- Muestra un mensaje de orientación indicando que seleccione el tablero donde desea jugar.	
	4- Muestra la interfaz del tablero seleccionado. 5- Muestra las instrucciones generales del juego.	
6- Selecciona la opción aceptar para comenzar a jugar.	7- Muestra el área del jugador en turno. 8- Permite el lanzamiento de los dados. 9- Muestra ejercicio.	
10- Realiza el ejercicio.	11- Evalúa la respuesta. 12- Muestra una ventana de confirmación informando que la respuesta es correcta y la registra. Incrementa el contador de la puntuación y muestra el resultado obtenido. 13- Muestra las casillas con posibles jugadas. 14- Mueve la ficha seleccionada hacia la casilla seleccionada.	

	15- Comienza el flujo normal de eventos a partir del paso 8 hasta que un jugador logre colocar en la casilla de la meta las cuatro fichas correspondientes.
	16-Muestra los resultados. Ver caso de uso: "Mostrar resultado" . 17-Finaliza el caso de uso.
Flujos alternos	
11. a: "La respuesta es incorrecta".	
Actor	Sistema
	11. a.1-La respuesta es incorrecta el sistema muestra una ventana de confirmación y la registra. 11. a.2- Disminuye el contador de la puntuación y muestra el resultado obtenido. 11. a.3-Regresa al paso 7 del Flujo normal de eventos .

Tabla 2.Descripción textual del caso de uso: Realizar juego.

Caso de uso: Obtener trofeo

Objetivo	Asignar un trofeo al usuario.
Actores	Usuario
Resumen	El caso de uso inicia cuando se le otorga un trofeo al usuario de acuerdo a la efectividad con que haya respondido las preguntas. Termina el caso de uso.
Complejidad	Alta

Prioridad	Crítico	
Precondiciones	Juego terminado.	
Postcondiciones	Se asignó un trofeo al usuario.	
Referencias	RF5,RF5.1, RF5.2	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
	<p>1-Otorga el trofeo de acuerdo a la efectividad con que haya respondido las preguntas.</p> <ul style="list-style-type: none"> ✓ Oro, plata o bronce (se obtiene al responder correctamente de forma consecutiva 20, 15 y 10 preguntas respectivamente). ✓ Experto (se obtiene al responder correctamente todas las preguntas). <p>2- Incrementa el contador de la puntuación.</p> <p>3-Termina el caso de uso.</p>	

Tabla 3. Descripción textual del caso de uso: Obtener Trofeo.

Caso de uso: Otorgar bono.

Objetivo	Obtener bonos en el juego.
Actores	Usuario
Resumen	El caso de uso inicia cuando el sistema verifica que la ficha ha caído en una casilla con bono y lo activa. El usuario selecciona el bono obtenido. El sistema realiza la acción dentro del juego según el tipo de bono, termina el caso de uso.
Complejidad	Alta

Prioridad	Crítico	
Precondiciones		
Postcondiciones	Se asignó el bono al usuario.	
Referencias	RF5.3	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
	1-Verifica que la ficha ha caído en una casilla con bono. 2-Si cayó en una casilla con bono activo en el área de bonos, muestra el símbolo y el nombre del bono obtenido, y permite seleccionarlo.	
3-Selecciona el bono obtenido.	4-Realiza la acción en el juego según el tipo de bono. 5- Termina el caso de uso.	

Tabla 4.Descripción textual del caso de uso: Otorgar bono.

Caso de uso: Mostrar resultado

Objetivo	Mostrar los resultados obtenidos por el usuario ganador del juego.
Actores	Usuario
Resumen	El caso de uso se inicia cuando el sistema muestra los resultados obtenidos, termina el caso de uso.
Complejidad	Alta
Prioridad	Crítico

Precondiciones	Juego terminado.
Postcondiciones	Se mostraron los resultados.
Referencias	RF6.1, RF6.2, RF6.3, RF6.4
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
	<p>1-Muestra los resultados que tuvo el usuario ganador durante la partida:</p> <ul style="list-style-type: none"> ✓ Nombre del usuario. ✓ Preguntas correctas. ✓ Preguntas incorrectas. ✓ Puntuación. ✓ Trofeos obtenidos. <p>2-Termina el caso de uso.</p>

Tabla 5.Descripción textual del caso de uso: Mostrar resultado.

Caso de Uso: Guardar estado

Objetivo	Guardar el estado del juego.
Actores	Usuario
Resumen	El caso de uso inicia cuando el usuario desea salir del juego. El sistema guarda el juego, termina el caso de uso.
Complejidad	Alta
Prioridad	Crítico
Precondiciones	Usuario autenticado en la plataforma para acceder al módulo “Parchís”.

Postcondiciones	Se guardó el estado del juego.	
Referencias	RF7	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1-Selecciona la opción salir del juego.	2- Muestra un mensaje de confirmación de la acción ¿Está seguro que desea salir del juego?	
3-Confirma salida del juego.	4- Guarda el estado del juego. 5- Muestra la pantalla principal del juego. 6- Finaliza el caso de uso.	

Tabla 6.Descripción textual del caso de uso: Guardar estado.

Descripción de casos de uso del juego “Localiza y Aprende”**Caso de uso: Realizar juego**

Objetivo	Permitir al usuario interactuar con el juego.
Actores	Usuario
Resumen	El caso de uso inicia cuando el usuario accede a la interfaz principal. El usuario realiza las actividades del juego, termina el caso de uso.
Complejidad	Alta
Prioridad	Crítico
Precondiciones	Usuario autenticado en la plataforma para acceder al módulo “Localiza y Aprende”.
Post condiciones	Juego terminado.
Referencias	RF1, RF1.1, RF2, RF2.2, RF3, RF3.1, RF3.2, RF3.3, RF3.4, RF4, RF4.1, RF4.2, RF4.3, RF4.4, RF5, RF5.1, RF5.2, RF5.3, RF5.4, RF5.3, RF7.

Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1- Selecciona en la plataforma el juego "Localiza y Aprende".	2-Ejecuta el juego, mostrando la interfaz principal. 3-Muestra las instrucciones del juego.
4- Presiona la opción aceptar para comenzar a jugar.	6-Carga los barcos en uno de los planos. 7-Permite rotar (vertical u horizontal) los barcos que se encuentran en la parte inferior del otro plano y ubicarlos en dicho plano. 8-Permite lanzar los dados. 9-Verifica que en el plano del contrario existe un barco en la posición (x, y) indicada por los dados. 9.1-Si no existe en el plano del contrario un barco en la posición (x, y) indicada. Ver Alternativa 1: "Espere su turno". 10-Muestra ejercicio.
11-Realiza ejercicio.	12-Evalúa la respuesta y la registra. 12.1-Si la respuesta es incorrecta Ver Alternativa 2: "La respuesta es incorrecta". 13-Activa la mirilla. 14-Permite movimiento de la mirilla en el plano del contrario.
15- Ubica la mirilla en la posición (x, y) indicada por los dados y da clic.	16-Verifica que ha ubicado correctamente la mirilla en la posición (x, y) indicada por los dados.

	<p>16.1-Si no ubica correctamente la mirilla en la posición (x, y) indicada por los datos. Ver Alternativa 3: "Espere turno".</p> <p>17-Disminuye la vida del barco y lo muestra.</p> <p>17.1-Si no tiene vida el barco. Ver Alternativa 4: "Elimina barco".</p> <p>18- Comienza el flujo normal de eventos a partir del paso 9 hasta que un jugador logre destruir la flota del contrario.</p>
	<p>19-Muestra los resultados. Ver caso de uso: "Mostrar resultado".</p> <p>20-Termina el caso de uso.</p>
Flujos alternos	
Alternativa 1: "Espere su turno"	
Acción del actor	Acción del sistema
	1-Verifica que no existe en el plano del contrario un barco en la posición (x, y) indicada por los datos.
2-Espera su turno.	3-Regresa al paso 9 del Flujo normal de eventos .
Alternativa 2: "La respuesta es incorrecta"	
Acción del actor	Acción del sistema
	1- Registra la respuesta.
	2-Desactiva la mirilla.

3-Espera su turno.	4-Regresa al paso 8 del Flujo normal de eventos .
Alternativa 3:"Espere turno".	
Acción del actor	Acción del sistema
	1-.Verifica que ha ubicado incorrectamente la mirilla en la posición (x, y) indicada por los dados.
2- Espera su turno.	3-Regresa al paso 8 del Flujo normal de eventos .
Alternativa 4:"Elimina barco".	
Acción del actor	Acción del sistema
	1-Muestra el barco sin vida. 2-Elimina el barco del plano. 3-Incrementa el contador de la puntuación del jugador y lo muestra.
	4-Regresa al paso 8 del Flujo normal de eventos .

Tabla 7.Descripción textual del caso de uso: Realizar juego.

Caso de uso: Otorgar trofeo

Objetivo	Asignar un trofeo al usuario.
Actores	Usuario
Resumen	El caso de uso cuando se le asigna un trofeo al usuario, termina el caso de uso.
Complejidad	Alta

Prioridad	Crítico	
Precondiciones	Juego terminado.	
Post-condiciones	Se asignó un trofeo al usuario.	
Referencias	RF6.1	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
	1-Asigna trofeo. 2-Termina el Caso de Uso.	

Tabla 8.Descripción textual del caso de uso: Otorgar trofeo.

Caso de uso: Otorgar bono.

Objetivo	Obtener bonos en el juego.
Actores	Usuario
Resumen	El caso de uso se inicia cuando el sistema elimina un barco (tiene asociado un bono) del plano de coordenadas del contrario. El sistema le asigna una acción dentro del juego según el tipo de bono, termina el caso de uso.
Complejidad	Alta
Prioridad	Crítico
Precondiciones	Usuario autenticado en la plataforma para acceder al módulo “Localiza y Aprende”.
Post-condiciones	Se asignó un bono al usuario.
Referencias	RF6.2

Flujo normal de eventos	
Acción del actor	Respuesta del sistema
	1-Elimina el barco (tiene asociado un bono). 2-Se muestra en el área de bono el nombre y el símbolo del tipo de bono asignado.
3-Da un clic en el bono asignado.	4-Se activa el bono realizando la acción que corresponda en el juego según el tipo de bono. 5-Termina el caso de uso.

Tabla 9.Descripción textual del caso de uso: Otorgar bono.

Caso de Uso: Mostrar resultado

Objetivo	Mostrar los resultados obtenidos por el usuario ganador del juego.
Actores	Usuario
Resumen	El caso de uso inicia cuando el sistema muestra los resultados obtenidos, termina el caso de uso.
Complejidad	Alta
Prioridad	Crítico
Precondiciones	Juego terminado.
Postcondiciones	Se mostró los resultados.
Referencias	RF7, RF7.1, RF7.2, RF7.3, RF7.4
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
	1-Calcula la efectividad (Preguntas correctas/

	<p>(Preguntas incorrectas+ Preguntas correctas *100)).</p> <p>2-Muestra los resultados que tuvo el usuario ganador durante la partida:</p> <ul style="list-style-type: none"> ✓ Nombre del usuario. ✓ Mostrar efectividad. ✓ Puntuación. <p>3-Termina el caso de uso.</p>
--	--

Tabla 10.Descripción textual del caso de uso: Mostrar resultado.

Caso de Uso: Guardar estado

Objetivo	Guardar el estado del juego.	
Actores	Usuario	
Resumen	El caso de uso inicia cuando el usuario desea salir del juego. El sistema guarda el juego, termina el caso de uso.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Usuario autenticado en la plataforma para acceder al módulo "Localiza y Aprende".	
Post-condiciones	Se guardó el estado del juego.	
Referencias	RF8	
Flujo normal de eventos		
Acción del actor	Respuesta del sistema	
1-Selecciona la opción salir del juego.	2- Muestra un mensaje de confirmación de la acción ¿Está seguro que desea salir del juego?	

3-Confirma salida del juego.	4- Guarda el estado del juego 5- Muestra la pantalla principal del juego. 6- Finaliza el Caso de Uso.
------------------------------	---

Tabla 11.Descripción textual del caso de uso: Guardar estado.

2.6. Conclusiones parciales

Como resultado de la investigación realizada en este capítulo se puede concluir que:

Se definieron los principales conceptos asociados al dominio del problema, mediante un Modelo Conceptual. Se determinaron los requisitos funcionales y no funcionales con el objetivo de establecer las capacidades y cualidades que debe cumplir el sistema. Se obtuvieron los casos de uso a partir del levantamiento de los requisitos, así como las descripciones detalladas de los mismos.

Capítulo 3: Diseño del sistema

3.1. Introducción

El siguiente capítulo está centrado en el flujo de trabajo de Diseño. Según la IEEE el diseño de un software es el proceso de definir la arquitectura, componentes, interfaces y otras características de un sistema o componente. (29) Tiene entre sus principales artefactos el modelo del diseño, este a su vez está compuesto por el diagrama de clases del diseño y el diagrama de secuencia, los cuales serán mostrados en el capítulo. Se describen además, las características y ventajas del patrón arquitectónico utilizado y se detalla, el estándar de codificación a tener en cuenta en la implementación de los juegos con el fin de obtener un estilo de programación homogéneo, comprensible y claro.

3.2. Modelo de diseño

El modelo de diseño es planteado como un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar, constituyendo una entrada principal en la actividad de implementación. (30)

3.2.1. Diagrama de clases del diseño

Los diagramas de clases del diseño permiten describir gráficamente las especificaciones de las clases del software. Muestran las clases (descripciones de objetos que comparten características comunes) que componen el sistema y cómo se relacionan entre sí.

A continuación se muestran los diagramas de clases del diseño correspondientes para cada juego educativo.

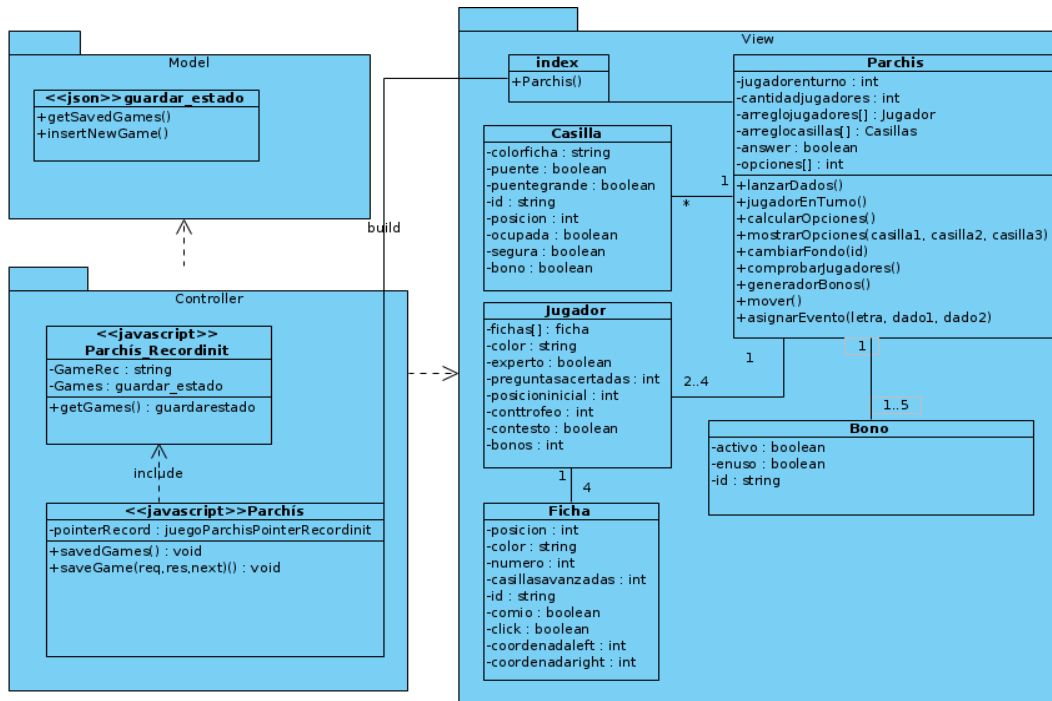


Figura 4. Diagrama de clases del diseño del juego “Parchís”

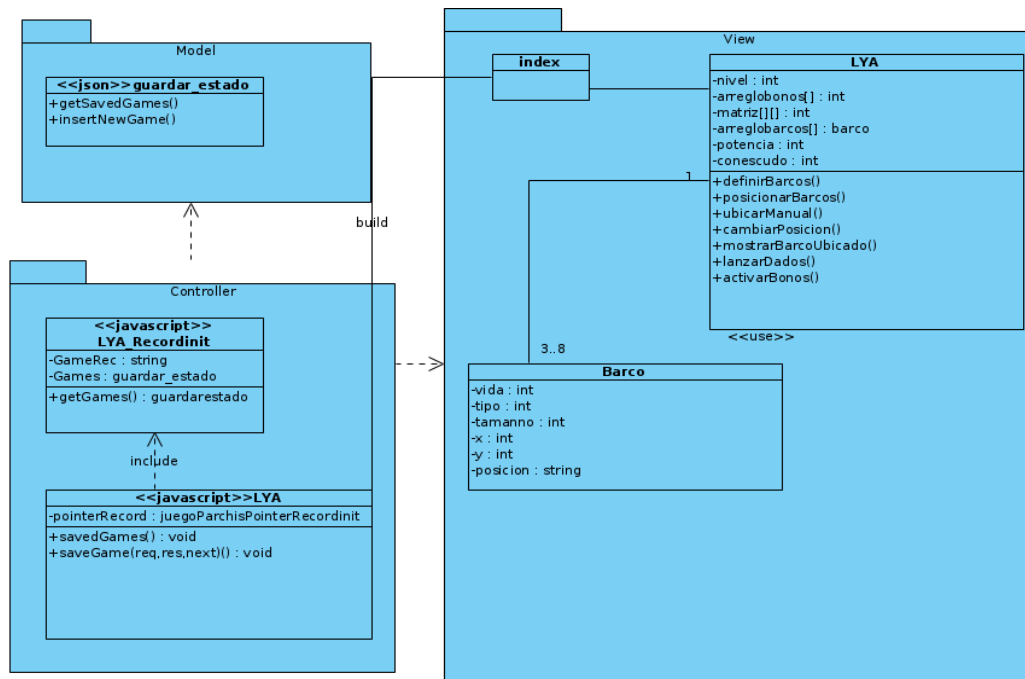


Figura 5. Diagrama de clases del diseño del juego “Localiza y Aprende”.

3.2.2. Diagramas de secuencia

La creación de los diagramas de secuencia es una de las actividades más importantes en el desarrollo de un sistema, pues al construirlos se toman decisiones claves acerca de su funcionamiento futuro. Un diagrama de secuencia consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. (31)

Debido a que la mayoría de los casos de uso son pequeños , exceptuando realizar juego para ambos sistemas, se realizó un solo diagrama de secuencia para cada aplicación que incluye todos los casos de uso.

A continuación se muestran los diagramas de secuencia de los sistemas (figuras 4 y 5):

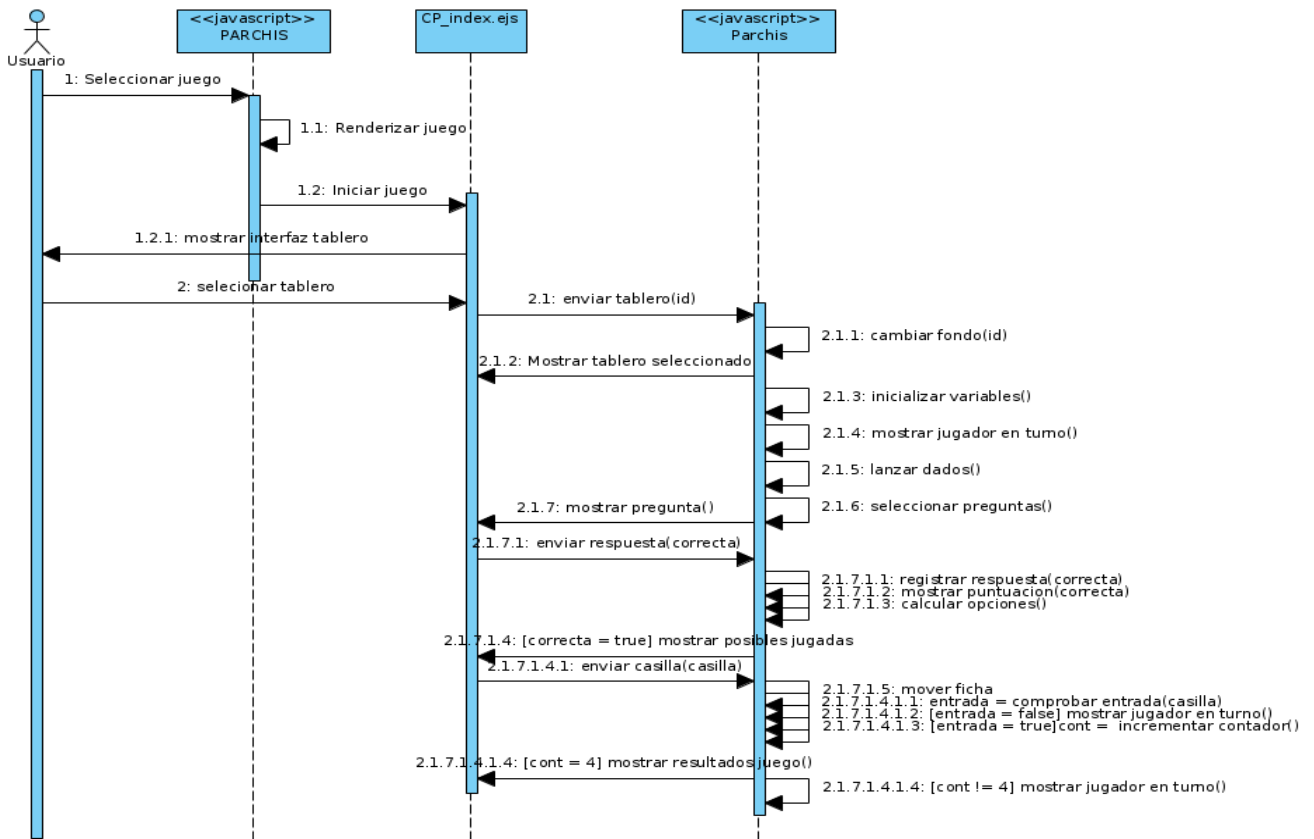


Figura 6. Diagrama de secuencia del caso de uso Realizar juego. Juego de “Parchís”.

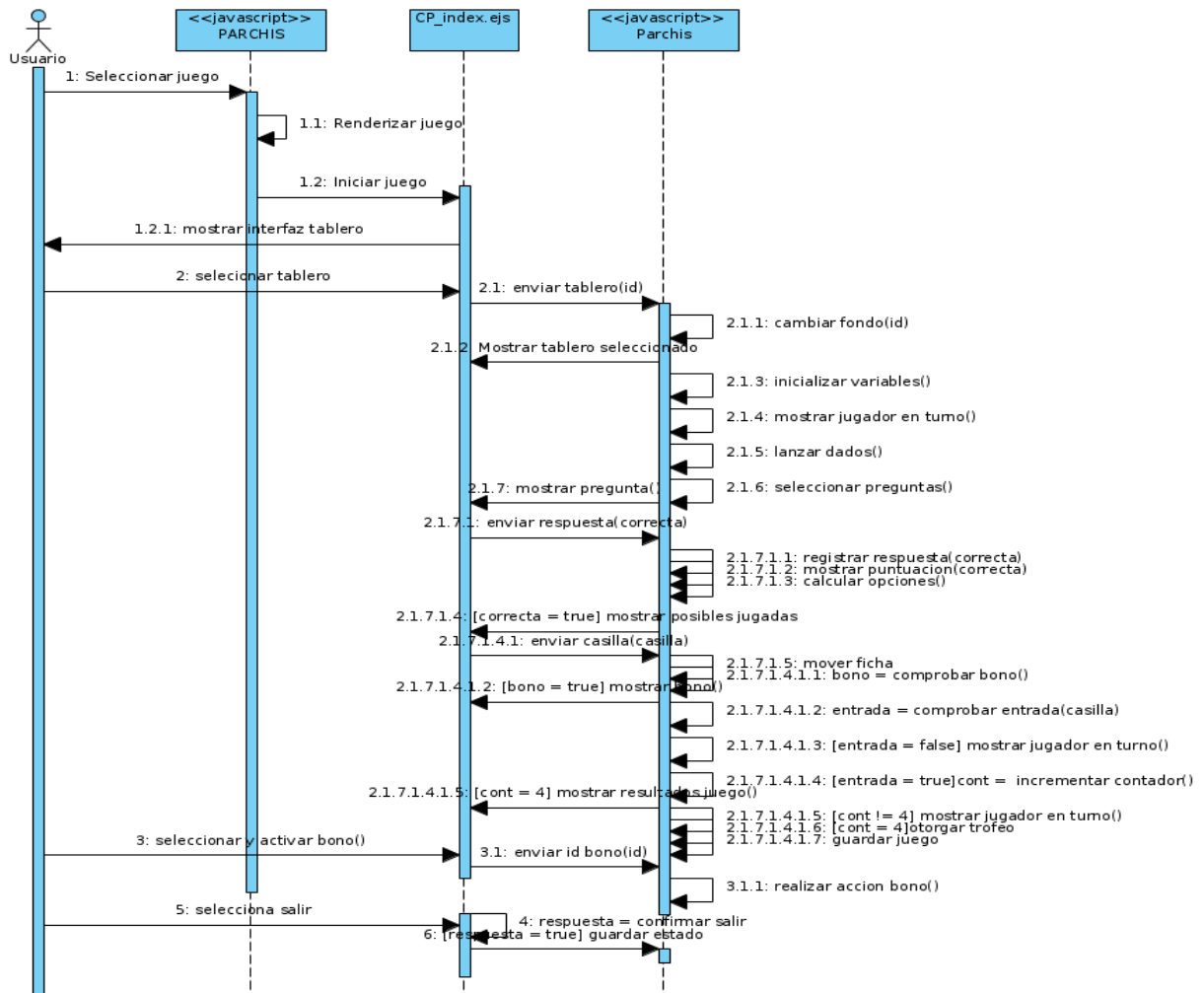


Figura 7. Diagrama de secuencia del caso de uso Realizar juego. “Localiza y Aprende”.

3.3. Definiciones de diseño

Debido a que los sistemas a desarrollar son módulos integrados a la plataforma Pointer, deben utilizar la misma arquitectura y estilos de código aplicados en dicha plataforma. Estos estándares y patrones fueron definidos por el arquitecto del proyecto y se detallan a continuación.

3.3.1. Patrón arquitectónico

El patrón de arquitectura utilizado es el MVC (Modelo – Vista – Controlador). A continuación se realiza una descripción general de sus principales funciones:

El patrón MVC es una arquitectura de diseño software para separar los componentes de aplicación en tres niveles, interfaz de usuario, lógica de control y lógica de negocio. Es una especialización de un modelo de capas, con la diferencia que se utiliza en entornos web como patrón por excelencia. (32)

Modelo: Es la capa encargada de encapsular toda la lógica de negocio de la aplicación.

Controlador: Es el eje central de la arquitectura, encargada de gestionar todas las peticiones, validar los inputs recibidos y dirigir cualquier petición de cualquier tipo. Solo se comunica con el modelo y responde a través de vistas.

Vista: Es la respuesta de cada controlador y lo que se le presenta al usuario final, se puede comunicar con el controlador, los “helpers” (“ayudantes”) apoyan tanto al controlador como a la vista para hacer más livianas algunas.

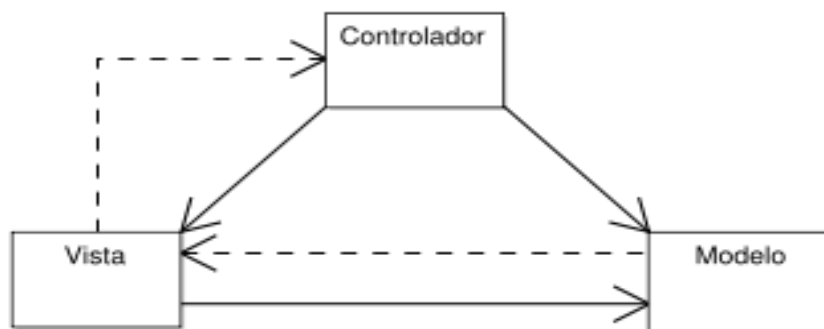


Figura 8. Patrón de arquitectura Modelo-Vista-Controlador.

Las principales ventajas de hacer uso del patrón MVC son (33):

- ✓ Separa el Modelo de la Vista, mostrando los datos de la representación visual separados de los mismos.
- ✓ Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.
- ✓ Crea independencia de funcionamiento.
- ✓ Facilita el mantenimiento en caso de errores.
- ✓ Suelen evolucionar en el tiempo, adaptándose a lo nuevo de los sistemas operativos, a los teléfonos móviles o sencillamente adaptándose a nuevas necesidades.

En resumen, el patrón arquitectónico MVC favorece el diseño de sistemas software en base a componentes que tienen una alta cohesión: el modelo se encarga de gestionar el estado del sistema, y la vista de generar representaciones visuales de dicho estado; mantienen un elevado grado de desacoplamiento, haciendo posible la modificación e incluso sustitución de cualquiera de ellos sin afectar al resto. Todo ello contribuye a simplificar el diseño de aplicaciones complejas que, de otra forma, resultarían mucho más difíciles de abordar y mantener, además disminuye el esfuerzo de los desarrolladores a la hora de programar, permitiéndole la reutilización de los componentes y el

desarrollo de prototipos rápidos.

3.3.2. Integración a la plataforma Pointer

Para integrar los juegos a la plataforma estos tienen que cumplir ciertas características arquitectónicas y se tiene que seguir una serie de pasos, los cuales se muestran a continuación.

- ✓ Ajustar los juegos a la estructura de la plataforma.
- ✓ Crear una nueva carpeta con un nombre distintivo en: /data/plataforma/public/juegos.
- ✓ Agregar el identificador a la lista de juegos en: /data/config/settings.js.
- ✓ Cada juego debe tener:
 - Un controlador (controller) con el mismo nombre que el identificador del juego.
 - Un actioninitial (método que será llamado para comenzar el juego).

3.3.3. Estándar de codificación

Uno de los aspectos fundamentales que se debe tener en cuenta a la hora de implementar un software, es el uso de un estándar de codificación. Estos se definen con el objetivo de establecer un estilo de programación homogéneo que les permita a los programadores de un proyecto trabajar de forma coordinada. Las principales ventajas de utilizar un estándar para escribir el código de las aplicaciones son:

- ✓ Reducción de los errores.
- ✓ Obtención de un código claro y comprensible.
- ✓ Garantizar una buena comunicación entre los programadores del proyecto.
- ✓ Facilitar el mantenimiento del software.
- ✓ Rapidez a la hora de realizar cambios o nuevas versiones del software. (34)

El estándar de codificación definido por la arquitectura de la plataforma Pointer fue idiomatic.js. (Manual completo escrito por varias personas donde se detallan una serie de consejos e ideas de buenas prácticas para programar JavaScript). A continuación se muestran algunas de estas prácticas:

- ✓ Eliminar el espacio en blanco de fin de línea.
- ✓ Evitar códigos con nombres de variables pobres, poco descriptivos.
- ✓ Hacer return tempranos para mejorar la legibilidad del código.
- ✓ Evitar comentarios en el final de línea. (35)

Además se realizaron una serie de convenios entre el arquitecto y el equipo de desarrollo quedando definido:

- ✓ En los juegos no deben existir ni variables ni funciones globales.

- ✓ Para los nombres de las variables y los métodos se utilizará lowCamelCase (primera letra en minúscula y las demás en mayúscula).
- ✓ Para los nombres de las clases se utilizará UpperCamelCase (la primera letra de cada una de las palabras es mayúscula).

3.4. Conclusiones parciales

Como resultado de la investigación realizada en este capítulo se puede concluir que:

Con la elaboración de los diagramas de secuencia se logró describir los aspectos dinámicos del sistema y mostrar las interacciones entre los objetos del sistema y los mensajes enviados entre ellos, ordenados según su secuencia en el tiempo. Se describió la estructura de los sistemas mediante el diagrama de clases. Ambos diagramas conformaron el modelo de diseño los cuales constituyen la entrada fundamental en las actividades que se realizan en el flujo de trabajo de Implementación. Con la descripción detallada de la arquitectura definida por el proyecto Multisaber-Navegante se logró una mejor comprensión de la misma.

Capítulo 4: Implementación y prueba

4.1. Introducción

Teniendo en cuenta los resultados del diseño, en el presente capítulo se realiza la implementación. El resultado fundamental de este flujo de trabajo es el modelo de implementación, el mismo está conformado por el diagrama de componentes y el de despliegue. Por último se le realizan pruebas a los juegos con el objetivo de detectar los errores del software y verificar que los requisitos han sido implementados correctamente.

4.2. Modelo de implementación

Este modelo representa la composición física de la implementación en términos de subsistemas de implementación. Describe cómo los elementos de diseño se implementan en componentes. (30)

Se considera el artefacto más significativo del flujo de trabajo de Implementación, debido a la importancia que tiene para los desarrolladores comprender el funcionamiento del sistema desde el punto de vista de componentes y sus relaciones. Este modelo está conformado por el diagrama de componentes y el de despliegue.

4.2.1. Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes o bibliotecas cargadas dinámicamente. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

Se utilizan para:

- ✓ Modelar la vista estática de un sistema.
- ✓ Mostrar la organización y las dependencias entre un conjunto de componentes.
- ✓ Mostrar qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema. (36)

A continuación se muestran los diagramas de componentes de los sistemas:

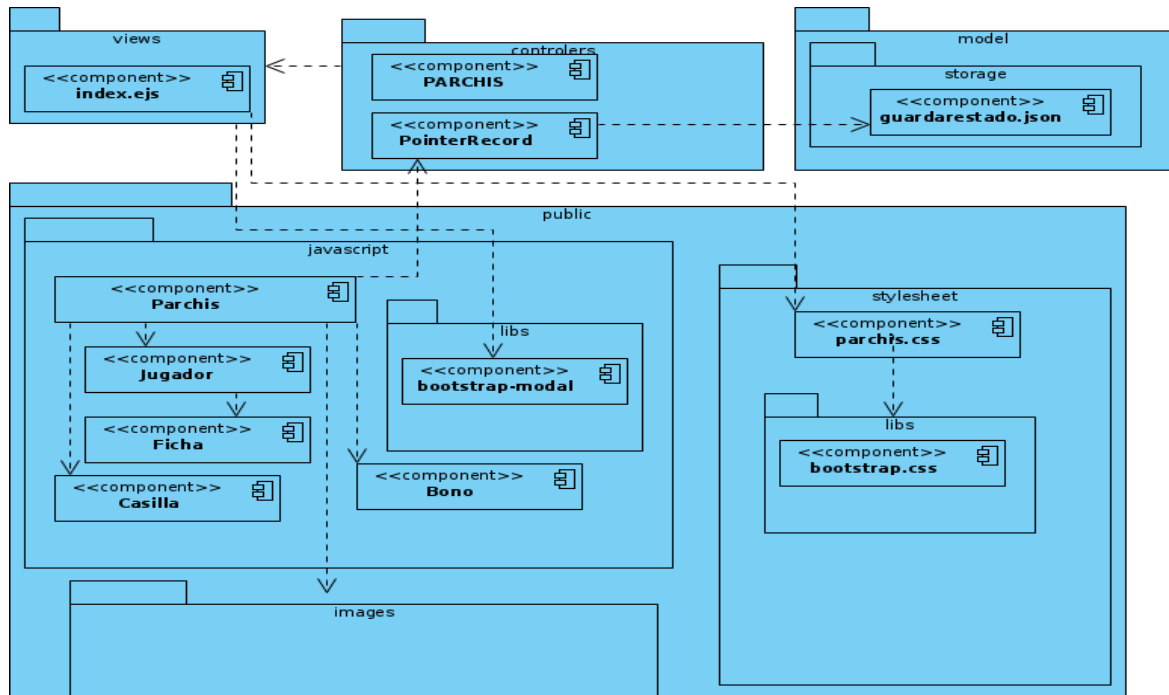


Figura 9. Diagrama de componentes del juego “Parchís”.

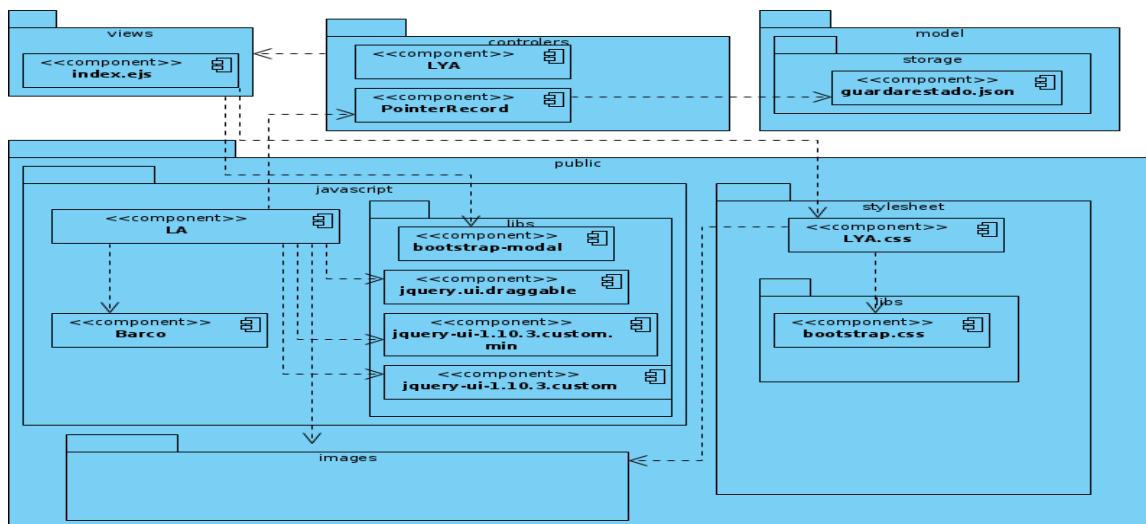


Figura 10. Diagrama de componentes del juego “Localiza y Aprende”.

4.2.2. Diagrama de despliegue

El modelo 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. Un nodo es un elemento

físico que existe en tiempo de ejecución y representa un recurso computacional, que generalmente tiene memoria y a menudo, capacidad de procesamiento. Los nodos se utilizan para modelar la tipología del hardware sobre el que se ejecuta el sistema. Representan un procesador o un dispositivo sobre el que se pueden desplegar los componentes. La relación entre un nodo y el componente que despliega puede mostrarse con una relación de dependencia.

Los estereotipos permiten precisar la naturaleza del equipo:

- ✓ **Procesadores:** Nodo con capacidad de procesamiento. Puede ejecutar un componente.
- ✓ **Dispositivos:** Nodo sin capacidad de procesamiento. Representa cualquier otro dispositivo hardware. (37)

A continuación se muestran las dos formas en que los juegos pueden estar distribuidos:

Forma 1:

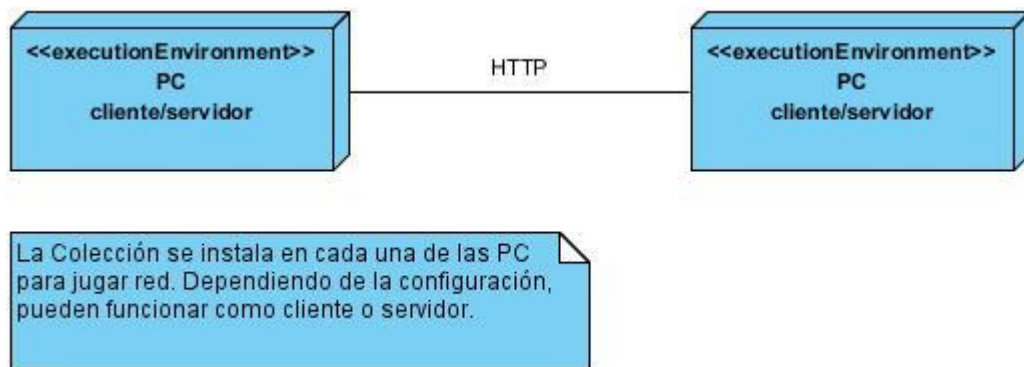


Figura 11. Diagrama de despliegue para MundoClick online.

Forma 2:



Figura 12. Diagrama de despliegue para MundoClick offline.

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. Con las pruebas se verifican que todos los requerimientos hayan sido implementados y se determinan los defectos en el software. (38)

Entre sus objetivos están:

- ✓ Detectar defectos en el software.
- ✓ Verificar la integración adecuada de los componentes.
- ✓ Verificar que todos los requisitos se han implementado correctamente.
- ✓ Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el software al cliente. (39)

Niveles de prueba

RUP define cuatro niveles de prueba (30):

Pruebas de unidad: Se realizan durante la fase de construcción, específicamente en el flujo de trabajo de implementación; las cuales se basan en probar los componentes implementados como unidades individuales. Las pruebas de unidad están divididas en dos grupos, pruebas de caja blanca y pruebas de caja negra.

Pruebas de integración: Se llevan a cabo durante la fase de construcción, las mismas involucran a un número creciente de módulos y terminan probando el sistema como conjunto. Estas pruebas se pueden plantear desde un punto de vista estructural o funcional. Verifican que los componentes interactúan entre sí de un modo apropiado después de haber sido integrados en el sistema. Se toman como casos de prueba los casos de uso del diseño. Para ello se utiliza el diagrama de secuencia correspondiente y se diseñan combinaciones de entrada y salida del sistema que lleven a distintas utilidades de las clases y en consecuencia de los componentes, que participan en el diagrama.

Pruebas de sistema: Prueban que el sistema funciona globalmente de forma correcta. Cada prueba del sistema prueba combinaciones de casos de uso bajo condiciones diferentes. Se prueba el sistema como un todo probando casos de uso unos detrás de otros y si es posible, en paralelo. En este nivel existen una gran variedad de pruebas que se utilizan según el interés que se tenga respecto al funcionamiento del software.

Pruebas de aceptación: Se realizan para permitir que el cliente valide y verifique todos los requisitos pactados. Estas pruebas las realiza el usuario final en lugar del responsable del desarrollo del sistema. El cliente es quien impone los requisitos pues quien mejor que él para dar fe de su satisfacción.

Tipos de prueba

Pruebas de Funcionalidad: Este tipo de pruebas se realiza con el propósito de verificar el cumplimiento de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.

Pruebas Unitarias: El objetivo con el que se realizan estas pruebas es aislar cada parte del programa y mostrar que las partes individuales son correctas.

Pruebas de Regresión: Se realizan para asegurar que cuando una NC encontrada en el sistema ha sido corregida ninguna de las funcionalidades liberadas previamente falla como resultado de las correcciones o que las características nuevamente agregadas no han creado conflicto con las versiones anteriores del software.

Pruebas de Carga: Se realizan generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperadas. (40)

Métodos de prueba

Existen dos métodos de pruebas: de caja blanca y de caja negra.

Método de caja blanca: Se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa. Se examina así la lógica interna del programa sin considerar los aspectos de rendimiento.

Método de caja negra: Se centran fundamentalmente en los requisitos funcionales del software. Estas se realizan sobre la interfaz de la aplicación, con el objetivo de demostrar que las funciones son correctas, que la entrada se acepta de forma adecuada y que se produce una salida correcta. A través de las mismas es muy usual encontrar funciones incorrectas o ausentes, errores de interfaz, errores de rendimiento, así como errores de inicialización o terminación. (41)

Pruebas de software aplicadas

Las pruebas aplicadas a los sistemas, según los niveles de pruebas que propone la metodología RUP, fueron las pruebas integración y sistema, las mismas están dirigidos a la integración de cada componente y el funcionamiento del sistema como un todo.

Según el tipo de prueba se seleccionó las pruebas funcionales con el propósito de verificar el cumplimiento de los requisitos funcionales y como método de prueba para probar los requerimientos, las funciones y las respuestas de los sistemas se decidió utilizar las pruebas de caja negra. Se determinaron además los casos de pruebas por cada caso de uso.

Resultado de las pruebas

A continuación se muestra una tabla resumen con las no conformidades (NC) resultantes:

	Primera iteración	Segunda iteración	Tercera iteración
Cantidad NC/”Parchís”	10	3	1
Cantidad NC/”Localiza y Aprende”	8	2	1

Tabla 12.No conformidades encontradas en los sistemas en tres iteraciones.

En la primera iteración los errores encontrados fueron en su mayoría de validación, de los cuales algunos se muestran a continuación:

En el caso del juego “Parchís”:

- ✓ Cuando una ficha “comía¹” a otra con los puntos alcanzados en un dado y otra en el mismo turno “comía” con lo restante del otro, ninguna de las dos podía caminar los puntos extras que se le asignaban a cada jugador.
- ✓ Cuando en una casilla había dos fichas de distintos colores, el juego lo interpretaba como un puente.
- ✓ Si había dos fichas en una casilla de salida de un jugador, y estas eran de color distinto a la casilla donde se encontraban, cuando el dueño de la casilla base salía, el sistema no se lo permitía.

En el caso del juego “Localiza y Aprende”:

- ✓ Si un barco ocupaba dos o más casillas, el juego marcaba como ocupada una sola, de esta forma se podía ubicar un barco debajo de otro.

¹ Se dice en el parchís cuando una ficha llega a la casilla de una otra contraria, desplazándola hacia la casilla base correspondiente. El jugador que “come” obtiene una cantidad extra de casillas a avanzar.

- ✓ La activación por más de una vez del bono “Lluvia de misiles” no funcionaba correctamente.
- ✓ La activación de un bono durante la animación del disparo, hacía que el juego otorgara el turno al jugador que activaba el bono.

En la segunda iteración las pruebas fueron dirigidas al funcionamiento de los juegos integrados a la plataforma Pointer. Los errores encontrados fueron relacionados con el diseño del juego, como imágenes corridas, y problemas con las ventanas modales que no funcionaban correctamente dentro de la plataforma, en ambos casos, si se mostraba dos ventanas modales seguidas, el juego solo mostraba una y ponía la pantalla en negro.

En la tercera iteración las pruebas fueron dirigidas al manejo de los datos relacionados con cargar y mostrar juegos guardados dentro de la plataforma, en ambos, en el momento de cargar un juego este tomaba los objetos que se encontraran dentro de otro, como un string(cadena de texto), ocasionando el mal funcionamiento de los juegos.

Luego de concluida cada iteración se corrigieron todas las NC encontradas de la iteración anterior, después de la iteración número 3 no se encontraron no conformidades por lo que se decidió concluir las pruebas.

4.3. Conclusiones parciales

Como resultado de la investigación realizada en este capítulo se puede concluir que:

Con los artefactos obtenidos en el flujo de implementación se realizó el desarrollo de los juegos “Parchís” y “Localiza y Aprende”. Una vez terminado el ciclo de pruebas se puede concluir que los sistemas no presentan anomalías en su funcionamiento.

Conclusiones generales

Una vez concluido el ciclo de desarrollo de la investigación se pudo arribar a las siguientes conclusiones:

- ✓ El estudio del estado del arte sobre las soluciones similares existentes a nivel nacional e internacional permitió concluir que estas no constituyen posibles soluciones para la problemática del presente trabajo, pero sí presentan una serie de características a tener en cuenta en el momento que se desarrolló la aplicación.
- ✓ Se realizó el desarrollo de los juegos “Parchís” y “Localiza y Aprende”, integrados a la plataforma Pointer.
- ✓ Con la realización de las pruebas de caja negra y el análisis de resultados se logró validar los juegos desarrollados, obteniéndose dos sistemas completamente funcionales.

Por lo expresado anteriormente se puede decir que se logró darle cumplimiento a los objetivos y tareas propuestas en la investigación.

Recomendaciones

El autor considera oportuno sugerir la valoración de las siguientes recomendaciones:

- ✓ Extender el uso de la plataforma de juegos educativos MundoClick a las escuelas primarias.
- ✓ Integrar a los juegos “Parchís” y “Localiza y Aprende” el módulo de preguntas diseñado por los especialistas del ministerio de educación.

Referencias Bibliográficas

1. **Labrada, Sonia Morejón.** eumed.net. [En línea] julio de 2011. [Citado el: 5 de enero de 2013.] <http://www.eumed.net/rev/ced/29/sml.htm>. Vol 3, Nº 29.
2. **Alarcón, Ricardo Peri.** eumed.net. [En línea] <http://www.eumed.net/libros-gratis/2010a/664/videojuego%20como%20herramienta%20para%20la%20ensenanza%20y%20la%20apacitacion%20de%20personal%20en%20medianas%20y%20grandes%20empresas.htm> Ricardo Peri Alarcón .
3. [En línea] [Citado el: 2013 de junio de 2013.] <http://www.elbebe.com/ninos-6-a-8-anos/caracteristicas-ninos-6-7-8-anos-etapa-escolar>.
4. **Marqués, Santiago Ferrer.** Los videojuegos. [En línea] 15 de enero de 2013. <http://ardilladigital.com/DOCUMENTOS/TECNOLOGIA%20EDUCATIVA/TICs/T8%20VIDEOJUEGOS/08%20LOS%20VIDEOJUEGOS.pdf>.
5. *Informe Horizon.* 2002 al 2013.
6. **Marín, Mareela.** [En línea] septiembre de 2011. [Citado el: 5 de enero de 2013.] <http://www.rcysostenibilidad.telefonica.com/blogs/documentoscatedras/files/2012/02/CuadernoRed2011-VideojuegosAprendizaje.pdf>.
7. [En línea] [Citado el: 15 de junio de 2013.] <http://www.guiainfantil.com/1098/las-capacidades-y-habilidades-de-los-ninos.html>.
8. **Dr. Patrick Felicia, Profesor, Investigador, Departamento de Ciencia.** Videojuegos en el aula. Bélgica : European Schoolnet, 2009.
9. [En línea] [Citado el: 22 de enero de 2013.] <http://proyectoediris.wordpress.com/2010/03/10/popiparchis-ortografico-para-primaria-interactivo-2/>.
10. [En línea] [Citado el: 15 de enero de 2013.] http://newton.cnice.mec.es/materiales_didacticos/fyqhundirlaflota/index.html.
11. [En línea] http://www.ehowenespanol.com/juegos-reforzar-graficacion-puntos-plano-coordenadas-info_216035.
12. **Alarcos.** [En línea] [Citado el: 5 de enero de 2013.] Disponible en: <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pd>.
13. **Martini Rodríguez, Jorge Luis Bradshaw Bemzat, Maikel.** *Análisis, diseño e implementación del Portal WAP del SIIPOL Móvil.* 2010.
14. **Paradigm, Roberto Canales Mora.** *Modelado UML con Visual.* [En línea] 2007. [Citado el: 2013 de enero de 2013.]

15. [En línea] [Citado el: 5 de diciembre de 2013.] <http://catedraprogramacion.foroactivos.net/t83-definicion-de-lenguaje-de-programacion-tipos-ejemplos>.
16. **Cantón, Alejandro Castillo**. *Manual de HTML5*.
17. desarrolloweb.com. [En línea] [Citado el: 6 de enero de 2013.] <http://www.desarrolloweb.com/manuales/CSS3.html>.
18. Librosweb. [En línea] [Citado el: 6 de diciembre de 2013.] <http://www.librosweb.es/javascript/>.
19. mejorando.la. [En línea] [Citado el: 6 de enero de 2013.] <https://mejorando.la/videos/curso-introduccion-node-js/>.
20. **Rebecca Murphey**. [En línea] [Citado el: 7 de enero de 2013.] <http://librojquery.com/>.
21. [En línea] [Citado el: 7 de enero de 2013.] <https://sites.google.com/site/vigomiciudad/otras-cosa-interesantes/ide-entornos-de-desarrollo-integrado>.
22. Globedia. [En línea] [Citado el: 7 de enero de 2013.] <http://cu.globedia.com/aspectos-teoricos-entorno-desarrollo-integrado-ide>.
23. softonic. [En línea] [Citado el: 9 de enero de 2013.] <http://netbeans-ide.softonic.com/>.
24. Tigris.Subversion. [En línea] [Citado el: 15 de diciembre de 2013.] <http://subversion.tigris.org>.
25. **González, Anaisa Hernández**. *Identificación de los procesos de negocio*. Ciudad de La Habana, Cuba : s.n., 2004.
26. *IEEE Standard Glossary of Software Engineering Terminology*. 1990. IEEE Std 610.12.
27. **Hall, Pretince**. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*.
28. Especificación de requerimientos. [En línea] <http://www.mitecnologico.com/Main/EspecificacionesDeRequerimientos>.
29. **Ana Fernández Vilas**. [En línea] marzo de 20 de 2001. [Citado el: 2 de marzo de 2013.] <http://tvdi.det.uvigo.es/~avilas/UML/node24.html>.
30. **Ivar Jacobson, Grady Booch, James Rumbaugh**. *Proceso unificado de desarrollo de software*. Madrid : s.n., 2000. ISBN.
31. **Guerrero, Luis**. *UML Diagramas de Interacción*. Chile:Departamento de Ciencias de la Computación : s.n., 2008.
32. **José Jorge Márquez Gómez**. [En línea] [Citado el: 13 de marzo de 2013.] <http://jorge.queideas.com/wp-content/uploads/2011/11/Arquitectura-MVC.pdf>.
33. [En línea] [Citado el: 2013 de 13 de marzo.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/rivera_l_a/capitulo2.pdf.
34. **Ramiro Agras Hechevarría, Danay Delgado del Río**. *Análisis, diseño e implementación de la versión 2.0 de los juegos del primer ciclo de la Colección Multisaber*. Ciudad Habana. : s.n., 2010.

35. [En línea] [Citado el: 6 de mayo de 2013.] https://github.com/rwldrn/idiomatic.js/tree/master/translations/es_ES.
36. **Tracy, Madeline García, Claudia González Soto.** [En línea] [Citado el: 3 de Mayo de 2013.] <http://www.google.com.cu/url?sa=t&rct=j&q=diagrama+de+componentes+caracter%C3%ADsticas&source=web&cd=24&cad=rja&ved=0CHsQFjAX&url=http%3A%2F%2Fvirtual.usalesiana.edu.bo%2Fweb%2Fpractica%2Farchiv%2Fcomponn.doc&ei=-Q6pUdGoGIfKswbjnICgBA&usg=AFQjCNEOLSH4ZCdx>.
37. **Quintero, Juan Bernardo.** *La ingeniería de software aplicada a las bases de datos.* [En línea] [Citado el: 3 de mayo de 2013.] <http://docencia.udea.edu.co/ingenieria/ArquitecturaSoftware/documentos/LaingenieriadesoftwareenlasBD.pdf>.
38. **Binder, Robert.** Object Oriented Software Testing Communications of te ACM September. September 1994.
39. Capítulo 4: Pruebas Funcionales y de Campo. [En línea] [Citado el: 23 de mayo de 2013.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/fuentes_k_jf/capitulo4.pdf.
40. **Pressman, Roger S.** *Ingeniería de Software: Un Enfoque Práctico.* España. Sexta Edición : s.n.
41. slideshare. [En línea] [Citado el: 9 de abril de 2013.] www.slideshare.net/rinconsete/pruebas-de-caja-blanca-y-negra.
42. [En línea] http://developyourdream.net/tutoriales/tesis/tomo_pdf/CAPITULO%20%20-%20MARCO%20TE%20D3RICO.pdf.
43. **D´Vera, Matha Lucía Ramírez.** El mundo del software educativo. [En línea] 1 de Octubre de 2009. [Citado el: 5 de enero de 2013.] <http://www.slideshare.net/MaLuRamirez/el-mundo-del-software-educativo-concelo>.
44. **Elena Rodríguez San Julián, Ignacio Megías Quirós, Ana Calvo Sastre, Esteban Sánchez Moreno, José Navarro Botella.** *Jóvenes y videojuegos. Espacio, significación y conflictos.* Ciudad de Frías, Madrid : s.n., 2002.
45. **J., Sánchez.** Construyendo y Aprendiendo con el Computador. Santiago de Chile, Universidad de Chile. : s.n., 1999.
46. **Velazquez, Norlen Rosales.** *Software educativo Pico Turquino.* La Habana : s.n., 2009.
47. **Ivar Jacobson, Grady Booch, James Rumbaugh.** El proceso unificado de desarrollo. 2001.
48. **Quintero, Juan Bernardo.** [En línea] [Citado el: 3 de mayo de 2013.]
49. **Pressman, Roger S.** *Ingeniería de Software, un enfoque práctico.*
50. —. *Ingeniería de Software: Un Enfoque Práctico.* España. Sexta Edición : s.n. ISBN: 9701054733.
51. **Mediavilla, Elena.** [En línea] http://www.ctr.unican.es/asignaturas/MC_OO/Doc/Casos_de_uso.pdf.

Bibliografía consultada

1. **Rebecca Murphey**. [En línea] [Citado el: 7 de enero de 2013.] <http://librojquery.com/>.
2. Revista de la OMPI. Los videojuegos: arte del siglo XXI. [En línea] [Citado el: 11 de enero de 2013.] http://www.wipo.int/wipo_magazine/es/2012/04/article_0003.html.
3. **Serrano, Dres. Alejandro**. Los videojuegos. Matrimonio y familia. . [En línea] 2013 de enero de 2013. <http://www.aciprensa.com/Familia/videojuegos1.htm>.
4. Los videojuegos estimulan la creatividad. . [En línea] <http://www.muyinteresante.es/tecnologia/articulo/los-videojuegos-estimulan-la-creatividad>.
5. Parchís ortográfico para primaria interactiva. [En línea] [Citado el: 20 de enero de 2013.] <http://proyectoediris.wordpress.com/2010/03/10/popiparchis-ortografico-para-primaria-interactiv>.
6. Introducción a UML. . [En línea] [Citado el: 20 de enero de 2013.] <http://docs.kde.org/stable/es/kdesdk/umbrello/uml-basics.html>.
7. Juegos de batalla naval. [En línea] <http://batalla-naval.chulojuegos.com/>.
8. Visual Paradigm. [En línea] [Citado el: 20 de enero de 2013.] <http://www.softpedia.es/programa-Visual-Paradigm-for-UML-Community-Edition-4164.html>.
9. Slideshare. Análisis y diseño orientado a objetos. [En línea] <http://www.slideshare.net/maria8003/qu-es-el-analisis-y-diseo-de-software-orientado-a-objetos>.
10. **Espinosa., MC José Martín Olguín**. Análisis orientado a objetos. [En línea] <http://yaqui.mx.l.uabc.mx/~molguin/as/IngSoft%201-4.pdf>.
11. El proceso unificado de desarrollo de software. . [En línea] <http://www.chaco.gov.ar/utn/disenodesistemas/apuntes/oo/ApunteRUP.pdf>.
12. Slideshare. Tipo de pruebas software. . [En línea] marzo de 2013. <http://www.slideshare.net/GuillermoLemus/tipos-de-pruebas-de-software>.
13. Pruebas de software. Gestión de Calidad y pruebas de software. . [En línea] <http://pruebasdesoftware.com/laspruebasdesoftware.htm>.
14. Casos de prueba blanca. . [En línea] <http://fcqi.tij.uabc.mx/usuarios/luisgmo/data/8.2%20prb-cal-mant.pdf>.
15. Pruebas de caja negra. [En línea] <http://www.globetesting.com/2012/08/pruebas-de-caja-negra/> .

Glosario de Términos

AppJS: Permite el uso de los API's de HTML 5 para crear aplicaciones atractivas desde procesadores de texto hasta juegos en 3D. La creación de interfaces de usuario personalizadas sólo se limita a tu imaginación.

CVS (del inglés: Concurrent Versions System):

CSS: Del inglés: Cascading Style Shet. Hoja de Estilo en Cascada. Dentro del diseño de páginas Web se presenta esta como la vanguardia en cuanto a definición de estilos dentro de las plantillas de diseño.

HTML: Acrónimo de Hyper Text Markup Language (lenguaje hipertexto de marcas). HTML es el lenguaje con el que se escriben las páginas Web, las que pueden ser vistas por el usuario mediante un tipo de aplicación nombrada "Navegador". Este lenguaje permite aglutinar textos, sonidos e imágenes y combinarlos a gusto. Además, permite la introducción de referencias a otras páginas Web por medio de los enlaces hipertexto.

I/O.Input/Output: Entrada/Salida. Siglas que hacen referencia a dispositivos o puertos de comunicación (serie, paralelo, joystick...) así como a la tarjeta que los controla (si no están integrados en la placa base), y a las instrucciones de los lenguajes de programación e interrupciones que poseen algunos componentes y programas para denotar, realizar o gestionar la función de entrada y salida de datos.

IDE: Del inglés: Integrated Development Environment. Entorno de Desarrollo Integrado. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de gráficas GUI. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de otras aplicaciones existentes.

Plugins: Pequeño programa que añade funcionalidades a otro programa, habitualmente de mayor tamaño que no posee. Un programa puede tener uno o más Plugins.

Descartes web 2.0: Es un proyecto conjunto del Ministerio de Educación de España y el Instituto de Matemáticas de la Universidad Nacional Autónoma de México. Su propósito ha sido desarrollar una herramienta de uso libre para la creación de unidades didácticas interactivas de matemáticas para los niveles educativos medio-superior y superior.

RUP: Del inglés: Rational Unified Process. Proceso Unificado de Racional. Proceso de desarrollo de software que en su modelación define como principales elementos a los trabajadores, las actividades, los artefactos y flujo de actividades. Guía el proceso de desarrollo de software.

UML: Del inglés: Unified Modeling Language. Lenguaje Unificado de Modelado, notación estándar para el modelado de sistemas software.

XML: Del inglés: Extensible Markup Language. Lenguaje de Marcas Extensibles. Constituye un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C) que permite definir la gramática de lenguajes específicos.

Anexos

Anexo 1: Guión del juego “Parchís”

1. Objetivo

Completar todos los tableros.

2. Descripción

Interfaces

- En el momento en que se accede a jugar se mostrarán los tableros que hay que vencer para ganar el juego y se permitirá escoger en cuál quieren jugar.
- Cuando se entra a jugar se mostrará el tablero con las 4 fichas por jugador en sus respectivas casas.
- El fondo debe estar en concordancia con el contenido correspondiente al nivel.
- Luego que una ficha de un jugador está fuera de la casa este deberá contestar una pregunta para poder avanzar excepto cuando todas sus fichas estén en el tramo final.
- De forma aleatoria saldrán en algunas casillas los iconos correspondientes a los bonos, de forma tal que la ficha que llegue a esa casilla pueda obtener el bono.
- Luego de las 10 primeras jugadas salen los bonos y vuelven a salir luego de 15 jugadas y luego de 20 y así aumentando siempre 5 a las jugadas anteriores.
- Los puntos deben mostrarse en forma de barra o algo que sea gráfico en el área correspondiente a la casa o base de los jugadores.

Reglas del juego

- Para poder salir de la base o casa debe tirarse un 5.
- El jugador escogerá con cual o cuales ficha camina los puntos que salieron en los dados.
- Cuando un jugador se come a otro camina 20 casillas extras con la ficha que desee excepto con la que comió.
- Cuando una ficha llega a la meta el jugador puede avanzar 10 casillas extras.
- Cuando sale doble se tira de nuevo.
- Cuando salen tres dobles consecutivos a un mismo jugador la ficha más adelantada que tenga va para la casa excepto las que están en el tramo final y el último doble no se camina.

Secuencia del juego por niveles

- Serán tantos niveles como contenidos distintos tenga el juego y no hay necesidad de completar uno para poder jugar en otro. Cuando se vencen todos los niveles se gana el juego.
- Para completar un nivel hay llegar a la meta.
- Todos los niveles tienen la misma complejidad, lo que cambia es el contenido. A medida que se avanza en el juego las preguntas que salen para contestar y poder avanzar van aumentando de complejidad.

Puntos

- Cada pregunta contestada correctamente aporta 3 puntos.
- Cada pregunta contestada incorrectamente resta -2 puntos.

Bonos

- Escudo protector (para que no te coman).
- Saltar puente (permite pasar un puente que te bloquee el paso).
- Dados mágicos (permite tirar de nuevo) (sale luego de que el jugador haya contestado correctamente 10 preguntas de forma consecutiva).
- Pista (te muestra la respuesta correcta a la pregunta que salió) (sale luego de que el jugador haya contestado correctamente 15 preguntas de forma consecutiva).
- Sacar ficha (permite sacar una ficha de la casa sin tener que tirar los dados)

Forma de obtener bonos: cuando la ficha caiga en la casilla en la que se encuentra el bono y cuando se respondan correctamente de forma consecutiva la cantidad de preguntas exigidas por los bonos “Dados mágicos” y “Pista”.

Forma de activar el bono: dando clic sobre el icono que representa al bono excepto en el de “Escudo protector” que se activa de forma automática cuando van a comerse una ficha del jugador.

Trofeos (se obtienen de acuerdo al cumplimiento de algunas metas, otorgan puntos extras).

- Oro, plata o bronce (se obtiene al responder correctamente de forma consecutiva 20, 15 y 10 preguntas respectivamente).
- Experto en <el contenido del nivel> (si responde correctamente todas las preguntas, se otorga cuando el jugador llega a la meta).

Anexo 2: Guión del juego “Localiza y Aprende”

1. Historia

Eres el capitán de una flota entera de barcos y te han asignado la misión de llevar segura a puerto dicha flota. Pero la ruta que tienes que seguir está poblada de piratas y otras flotas enemigas que quieren destruirte. Confiamos en que podrás completar la misión.

2. Objetivo

- Cuidar la flota y destruir a los enemigos.
- Desarrollar habilidades en la ubicación de puntos en un eje de coordenadas y ejercitar matemáticas.

3. Descripción

El juego consiste en una batalla naval

- Cada uno de los 5 niveles representa un reto para el jugador sin perder de vista los objetivos del juego.
- Para completar un nivel tiene que haber vencido al jugador contrario (o PC)
- El juego contara con 5 diferentes tipos de barcos (1 – Lancha 2 – FRAGATAS 3 – Buque 4 – Submarino 5- Porta Avión)
- El primer nivel consiste en destruir 8 barcos que se pondrán aleatoriamente en el mapa o eje de coordenada todos del tipo (1, 2,3).
- El segundo nivel consiste en destruir 5 barcos que se pondrán aleatoriamente en el mapa o eje de coordenada todos del tipo (3, 4,5).
- El tercer nivel consiste en destruir 5 barcos que se pondrán aleatoriamente en el mapa o eje de coordenada todos del tipo (1, 2, 3, 4,5).

Detalles del Juego

- El juego se inicia cuando el jugador tira los dados, estos muestran una coordenada (X, Y), ejemplo (6,5), mueve la ficha a la posición $X = 6$ y $Y = 5$.
- Si la posición donde puso la ficha es la correcta se le debe mostrar un ejercicio el cual debe responder correctamente para que su disparo sea certero.
- Si no lo pone en la posición correcta no es válido el disparo y se le descuenta punto (cantidad por definir).
- Si el ejercicio lo responde correctamente el disparo mantiene un 100% de efectividad sino el disparo no es efectivo.

- Cada disparo tiene una fuerza de 100, que al hacer contacto con cada embarcación causa un determinado daño.
- Cada barco tiene asociado una vida .**Ver tabla. 1**
- Los bonos salen aleatorios en el mapa, no puede haber un bono aleatorio en la misma posición de un barco.
- Cada barco tiene asociado un o unos determinados bonos que se activan cuando se destruye una embarcación, pero solo sale uno por partida.

	Tipo de embarcación	Vida
1	LANCHA	100
2	FRAGATAS	200
3	BUQUE	400
4	SUBMARINO	500
5	PORTA AVIÓN	800

Tabla 1

Puntos

- Cada barco destruido aporta una determinada cantidad de puntos.

	Tipo de embarcación	Puntos
1	LANCHA	50
2	FRAGATAS	100
3	BUQUE	150
4	SUBMARINO	200
5	PORTA AVIÓN	300

Bonos

Los bonos se clasifican de la siguiente manera:

Tipo de Bono	Función
--------------	---------

Reparar Barco	Al aplicarse a un determinado barco este aumenta en un 50% el nivel de deterioro que tenga.
Escudo protector	Al utilizarse protege en un área determinada los barcos que se encuentre ahí. El área será de 3 casillas en forma de cuadrado, desde el centro hacia afuera.
Tirar de nuevo	Al utilizarse se le da la opción al jugador de volver a tirar.
Lluvia de misiles	Al utilizarse sucede un bombardeo de forma aleatoria en todo el mapa.
Bomba atómica	Ocasiona un daño en un área de 4 casillas que el usuario escoge.
Elevar potencia	Duplica la fuerza de impacto del disparo.

Nota: en caso de los barcos que puedan tener varios bonos asociados esto se asignan aleatoriamente en cada partida. Por partida cada barco solo tendrá un bono.

Forma de activar el bono: seleccionando el bono correspondiente que está en el área de bonos. Luego seleccionar el barco o área donde se va a utilizar.

Trofeo

Tipo de Trofeo	Cómo se obtiene
Flota intacta	Cuando se ha logrado mantener la flota intacta, o sea el jugador contrario no le ha causado daño a ningún barco
Guardián Naval	Cuando se gana un nivel con más del 95% de efectividad.
Tiro certero	
Franco tirador	Se alcanza cuando todas las jugadas realizadas son efectivas, o sea ha dado siempre en el blanco.