

Universidad de las Ciencias Informáticas
Facultad 4



Desarrollo de los juegos Sopa de letras, Crucigrama,
Sudoku y Pasatiempo matemático para la colección de
juegos educativos MundoClick

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

Autores

Oscar Manuel Concepción Nápoles

Sandy Rodriguez De la Paz

Tutor

Ing. Osmany Montes de Oca Rodríguez

Co-tutor

Ing. Javier Bandomo Ruíz



Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Oscar Manuel Concepción Nápoles

Sandy Rodriguez De la Paz

Autor

Autor

Ing. Osmany Montes de Oca Rodríguez

Ing. Javier Bandomo Ruíz

Tutor

Tutor



“Si quieres trabajadores creativos, dales tiempo suficiente para jugar”

John Cleese



Dedicatoria

A mis padres por confiar en mí desde que me trajeron al mundo, a mi hermano y mis primos para que le sirva de ejemplo en un futuro. A mi novia que siempre me dio su apoyo incondicional, a mis abuelos y en especial a mi abuelo Pochocho que ya no está entre nosotros.

Oscar Manuel Concepción Nápoles

A mi mamá por apoyarme y amarme en todo momento, a mi abuela y mis tías por toda su entrega, amor y crianza pues son otras madres para mí. En general a mi familia por toda la ayuda que me ha brindado a lo largo de estos años y en la vida. A los que me quieren como un hijo a pesar de no compartir lasos de sangre, pues me han demostrado su amor incondicional.

Sandy Rodriguez De la Paz



Agradecimientos

A mis padres y a mi hermano por apoyarme y confiar en mí en todo momento.

A mi novia por estar a mi lado en cada momento brindándome su apoyo.

A mis profesores que durante toda la carrera me han ayudado a formarme profesionalmente.

A mis amigos del 135-306 por haber podido compartir con ellos estos 5 años de universidad.

A mis compañeros de universidad, a los de mi grupo en primer año aunque algunos no pudieron continuar, a los del 4501 y a todas aquellas personas con las que he podido compartir algún momento durante toda la carrera.

A mis tutores por siempre estar pendiente de los avances de la tesis y por tener siempre su disposición para lo que hiciera falta, muchas gracias por todo.

Oscar Manuel Concepción Nápoles

A toda mi familia por todo el apoyo y el amor que me ha dado, pues sin ella no hubiera sido posible estar aquí hoy.

A aquellos que son familia aunque no compartamos lazos de sangre pues me acogieron como uno más de los suyos.

A mis amigos pues son algo para atesorar toda la vida y me soportan a pesar de mis defectos.

A mis compañeros de universidad por hacer más fácil el día a día y por los momentos que compartí con ellos.

A mis profesores por ayudarme a formarme como profesional.

A mis tutores por su disposición, por su ayuda y por el tiempo que nos dedicaron.

A todos ellos muchas gracias.

Sandy Rodriguez De la Paz



Resumen

El proyecto Multisaber-Navegante, perteneciente al Centro de Tecnologías para la Formación (FORTES), se encuentra en pleno desarrollo de una colección de juegos educativos llamada MundoClick. El presente trabajo aborda el desarrollo de los juegos Sopa de letras, Crucigrama, Sudoku y Pasatiempo matemático, los cuales dotan a la colección de medios para contribuir al desarrollo de las habilidades de razonamiento lógico, matemático y verbal en los niños de edad comprendida entre 6 y 12 años. La estructura del trabajo se organizó a partir de la definición de cuatro capítulos que están encaminados al estudio del estado del arte, a las características del sistema, al desarrollo del diseño, la implementación y prueba de los juegos. Como lenguajes de desarrollo se utilizaron HTML, CSS y JavaScript apoyados en la plataforma Pointer la cual está orientada al desarrollo de juegos con estas tecnologías. El proceso de desarrollo estuvo guiado por la metodología RUP y fueron generados artefactos correspondientes a los flujos de trabajo Requisitos, Diseño, Implementación y Pruebas.

Palabras clave: juegos educativos, MundoClick.



Índice General

Introducción.....	1
Capítulo 1. Fundamento Teórico	5
1.1 Las TIC en el proceso educativo destinado a los niños.....	5
1.2 Software educativo	5
1.3 Los juegos como software educativo	7
1.4 Habilidades	8
1.5 Análisis de soluciones similares.....	10
1.6 Tecnologías y herramientas.....	15
1.7 Conclusiones del capítulo	25
Capítulo 2. Características del sistema	26
2.1 Juegos propuestos.....	26
2.2 Especificación de los requisitos del software	28
2.3 Modelo conceptual.....	36
2.4 Modelo de casos de uso del sistema	41
2.5 Conclusiones del capítulo	46
Capítulo 3. Diseño del sistema.....	47
3.1 Patrón arquitectónico	47
3.2 Modelo de diseño.....	48
3.3 Estándares de codificación	55
3.4 Conclusiones del capítulo	57
Capítulo 4. Implementación y prueba	58
4.1 Modelo de implementación	58
4.2 Pruebas de software	59
4.3 Conclusiones del capítulo	64
Conclusiones generales	66
Recomendaciones.....	67
Referencias Bibliográficas	68
Anexos	74



Introducción

El desarrollo vertiginoso de la ciencia y la tecnología que ha tenido lugar desde la segunda mitad del siglo XX hasta la actualidad está produciendo importantes transformaciones en la sociedad, hasta el punto de marcar la característica distintiva de este momento histórico con relación al pasado. Uno de sus aspectos más relevantes está asociado al surgimiento de las Tecnologías de la Información y las Comunicaciones (TIC) (1).

La informática, unida a las comunicaciones, posibilita prácticamente a todo el mundo el acceso inmediato a la información. En los últimos años la producción de software ha alcanzado gran auge, provocado fundamentalmente por el creciente desarrollo de estas ramas, tanto así, que hoy es común que en todas las esferas de la sociedad se haga uso de estos productos para darle solución a infinidad de problemas que se presentan a diario (1).

En la esfera educativa se ha impulsado el uso de las TIC para mejorar la calidad del proceso de aprendizaje, permitiendo recibir los conocimientos de una forma amena, interactiva e integradora. Un ejemplo de esto es el uso de juegos educativos como un tipo de software educativo (2).

El software educativo como apoyo a las actividades docentes en la escuela cubana, se ha implementado teniendo en cuenta los avances tecnológicos en este sentido, es decir, a medida que avanza la informatización de la sociedad. Estos evidencian un cambio favorable en el Sistema Educativo, pues constituyen una alternativa válida para ofrecer a los alumnos y estudiantes un ambiente propicio para la construcción del conocimiento (1).

La Universidad de las Ciencias Informáticas (UCI) tiene como una de sus misiones servir de soporte a la industria cubana de la informática. Esta cuenta con varios centros, entre los cuales se encuentra el centro de Tecnologías para la Formación (FORTES) encargado de satisfacer las demandas en cuanto a software educativo se refiere, brindando servicios y productos informáticos que sirven de apoyo al proceso docente-educativo en los diferentes niveles de enseñanza (3).

Hoy en día los niños conocen más en cuanto a tecnología se refiere, los avances en ésta han hecho que el mundo cambie, que la comunicación sea más efectiva y que los pequeños se interesen por cosas que hace algunos años parecían no importarles. Así pues, algunos prefieren estar sentados frente a la computadora jugando o el televisor a otras opciones, su interés ha cambiado, por lo que el modo de enseñarles también debe hacerlo (4).



Debido a estas tendencias el proyecto Multisaber-Navegante perteneciente al centro FORTES se dio a la tarea de crear una colección de juegos educativos denominada MundoClick. Dicha colección agrupará una variedad de juegos didácticos que contribuirá al desarrollo de habilidades en los niños de edad comprendidas entre 6 y 12 años de manera práctica, sencilla e interactiva.

Hasta el momento de los 27 juegos que deben integrar la colección, el grupo de desarrollado que se encarga de la realización del proyecto, ha logrado incorporar 19 entre los que se encuentran: El elevador, El parchís, Buscando parejas, Escalera de palabras, Memoria de trayectorias, Siguiendo la pista y muchos otros que potencian una serie de habilidades como la agilidad mental, el cálculo, la memoria, el uso del teclado y el mouse, todas ellas con la misión de fortalecer los contenidos de variadas temáticas y motivar el estudio en los alumnos. A pesar de contar con una parte importante del objetivo propuesto, la colección no cuenta con juegos que contribuyan a fomentar habilidades relacionadas con el pensamiento lógico, la comprensión y uso de contenidos matemáticos y el empleo adecuado del lenguaje, tales como el razonamiento lógico, matemático y verbal respectivamente. Según lo planteado, MundoClick actualmente no dispone de una solución capaz de despertar o fortalecer las destrezas mencionadas anteriormente.

Teniendo en cuenta lo planteado se define como **problema a resolver**: ¿Cómo contribuir al desarrollo de habilidades de razonamiento lógico, matemático y verbal en los niños desde la colección de juegos educativos MundoClick? Siendo así el **objeto de estudio** el desarrollo de juegos educativos para niños. Enmarcándose en el **campo de acción** desarrollo de juegos educativos que potencian habilidades de razonamiento lógico, matemático y verbal en los niños. Todo ello encaminado a cumplir con el siguiente **objetivo general**: Desarrollar los juegos: Sopa de letras, Crucigrama, Sudoku y Pasatiempo matemático para contribuir al desarrollo de habilidades como el razonamiento lógico, matemático y verbal en los niños desde la colección de juegos educativos MundoClick.

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

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



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

- Valoración crítica de los conceptos, tecnologías, metodologías y herramientas relacionadas con el objeto de estudio de la investigación.
- Levantamiento de los requisitos funcionales y no funcionales que formarán parte de la propuesta de solución asociada a la presente investigación.
- Modelado de los juegos: Sopa de letras, Crucigrama, Sudoku y Pasatiempo matemático acorde a la arquitectura de la plataforma Pointer.
- Implementación de los juegos: Sopa de letras, Crucigrama, Sudoku y Pasatiempo matemático acorde a la arquitectura de la plataforma Pointer.
- Integración de los juegos: Sopa de letras, Crucigrama, Sudoku y Pasatiempo matemático acorde a la arquitectura de la plataforma Pointer.
- Realización de pruebas funcionales a los juegos: Sopa de letras, Crucigrama, Sudoku y Pasatiempo matemático acorde a la arquitectura de la plataforma Pointer.

Como **posible resultado** se espera, la obtención de los juegos Sopa de letras, Crucigrama, Sudoku y Pasatiempo matemático integrados a la colección MundoClick.

Métodos científicos de la investigación

El método científico de investigación es la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones. Se clasifican en teóricos y empíricos, los cuales están dialécticamente relacionados (5).

En el desarrollo de la presente investigación se hace uso de los siguientes métodos científicos:

Métodos Teóricos

Histórico-lógico: Se utilizó durante el análisis de soluciones similares los conceptos y aplicaciones asociadas a la informática educativa, específicamente en el campo del software y los juegos educativos.

Analítico-sintético: Se utilizó durante la revisión de la documentación disponible asociada al objeto de la investigación, con el fin de realizar un análisis de la problemática a resolver, herramientas y tecnologías a utilizar, y sintetizar luego, una propuesta de solución adecuada al problema planteado.

Métodos Empíricos



Observación: Se empleó para identificar los aspectos significativos y las deficiencias presentes en las soluciones similares, para tenerlas en cuenta a la hora de desarrollar los juegos.

Estructura del documento:

Capítulo 1: Fundamentación teórica. Este capítulo recoge la fundamentación teórica de la investigación y muestra el estado del arte del tema que se investiga. Se argumenta además la selección de las herramientas y metodologías a utilizar para cumplir con el objetivo de esta investigación.

Capítulo 2: Características del sistema. Se identifican los requisitos funcionales y no funcionales, se definen los casos de usos del sistema y se realiza la descripción de cada uno de ellos.

Capítulo 3: Diseño del sistema. Se realiza el diseño del sistema a desarrollar con el objetivo de definir la arquitectura, los componentes y las interfaces del mismo. Se describe el sistema a partir de diferentes diagramas que representan su funcionamiento y estructura interna.

Capítulo 4: Implementación y pruebas. En este capítulo se describe cómo está implementado el sistema, a través de los diagramas de componentes y el diagrama de despliegue. Se documenta el proceso de implementación y la estrategia de prueba a utilizar para verificar la calidad de las funcionalidades implementadas.



Capítulo 1. Fundamento Teórico

Introducción

En este capítulo serán analizadas disímiles soluciones existentes identificando características que pueden ser utilizadas para dar solución al problema de la investigación. Se abordarán las ventajas del uso de las TIC en la educación, principalmente los juegos educativos. También se estudiarán las diferentes tecnologías y herramientas que serán utilizadas en el desarrollo de la solución en la presente investigación.

1.1 Las TIC en el proceso educativo destinado a los niños

El uso de las TIC en la esfera educativa representa un gran avance en el desarrollo del proceso enseñanza – aprendizaje, su uso mejora la productividad del trabajo entre profesores y alumnos permitiendo que los profesores puedan hacer llegar de una forma más amena los contenidos a los estudiantes.

La puesta en práctica del uso de las TIC en la educación es muy beneficiosa para todos los estudiantes ya sean de alta, media o baja motivación y habilidades, estos se ven favorecidos en el aprendizaje debido a las ventajas que adquieren como son (6):

- Aumento del interés por la materia estudiada.
- Mejora la capacidad para resolver problemas.
- Aprenden a trabajar en grupo y a comunicar sus ideas.
- Adquieren mayor confianza en sí mismos.
- Incrementan su creatividad e imaginación.

Frente a los tradicionales libros, vídeos y juegos, los nuevos contenidos educativos creados con recursos tecnológicos permiten presentar la información de otra forma.

Los contenidos se hacen más dinámicos (interactividad) más atractivos (presentación simultánea de texto, sonidos e imágenes) y más variados. Estas nuevas prestaciones pueden facilitar el aprendizaje y permitir mejoras cognitivas sobre todo de los alumnos con dificultades, al aplicar metodologías más activas y menos expositivas (6).

1.2 Software educativo

Se define como software educativo cualquier programa computacional cuyas características estructurales y funciones sirvan para apoyar el proceso de enseñar, aprender y administrar, es



decir, un material de aprendizaje especialmente diseñado para ser utilizado en una computadora en los procesos de enseñar y aprender (7).

Los software educativos pueden tratar las diferentes materias (matemáticas, idiomas, geografía, dibujo,...), de formas muy diversas (a partir de cuestionarios, facilitando una información estructurada a los alumnos, mediante la simulación de fenómenos...) y ofrecer un entorno de trabajo más o menos sensible a las circunstancias de los alumnos y más o menos rico en posibilidades de interacción; pero todos comparten cinco características esenciales (8):

- Son materiales desarrollados con finalidad didáctica.
- Utilizan la computadora como medio para realizar sus actividades.
- Son interactivos, dan respuestas inmediatas a las acciones de los estudiantes, permiten el intercambio de información entre la computadora y el estudiante.
- Se adaptan al ritmo de trabajo de cada estudiante, mostrándole ejercicios según las habilidades demostradas en sus actuaciones con el software.
- Son fáciles de usar, se requiere solo conocimientos mínimos en informática para su uso.

Tipos de software educativo

Algorítmicos: donde predomina el aprendizaje vía transmisión del conocimiento, pues el rol del alumno es asimilar el máximo de lo que se le transmite. Considerando la función educativa se pueden clasificar en (9):

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

Heurísticos: donde el estudiante descubre el conocimiento interactuando con el ambiente de aprendizaje que le permita llegar a él. Considerando la función educativa se pueden clasificar en (9):



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

1.3 Los juegos como software educativo

Un juego educativo está formado por un conjunto de actividades que los usuarios deben realizar. Cada una de estas actividades tiene unos objetivos, que representan el conocimiento que los usuarios deben adquirir durante la realización de la misma. Una actividad puede tener asociados uno o varios juegos informáticos, juegos con los que los usuarios interactúan para adquirir los conocimientos o habilidades descritas en los objetivos de la actividad (10).

Los videojuegos son un material muy motivador para la mayoría de los alumnos lo que ayuda a crear situaciones de aprendizaje altamente significativas. Además de los aspectos motivacionales, en numerosas experiencias se ha podido constatar que los juegos de ordenador aportan múltiples posibilidades educativas que van desde la motivación hasta el desarrollo de procedimientos tales como la adquisición de habilidades, la resolución de problemas, la toma de decisiones, etc. (11).

En los juegos educativos los usuarios pueden demostrar y aplicar sus conocimientos adquiridos en el aula o en el mismo juego. Estos se verán incitados a seguir ejercitando su mente para lograr finalizar dichos juegos y captar la mayor cantidad de información de los mismos.

El juego constituye una actividad voluntaria, que el alumnado desempeña libremente, y a través de actividades lúdicas, éste aprende a desarrollar habilidades sociales, vivir nuevas experiencias de forma individual o bien conjunta con otros alumnos y alumnas que le ayude a realizar descubrimientos, le favorece la adquisición del lenguaje incrementando su vocabulario así como facilita la iniciación del diálogo con aquellos con los cuales comparte el juego a la vez que desarrolla su imaginación y su creatividad (12).

Algunas de las principales características de los juegos según Catalina Huertas Ponce en su libro “El juego como Recurso Educativo” son:



- Todo juego es una actividad libre y espontánea, que el usuario decide cuando iniciarla y cuando terminarla.
- El juego implica seriedad y esfuerzo.
- Es placentero y divertido.
- Es expresivo, comunicativo, productivo, explorador y comparativo.

Los juegos educativos tienen que ser emocionales, creativos, al finalizar los mismos el usuario tiene que adquirir algún conocimiento. Por muchos años los juegos fueron caracterizados por ser solo para diversión, pero hoy en día es uno de los recursos didácticos de gran efectividad.

1.4 Habilidades

En la presente investigación se hace uso del término habilidad. Álvarez de Zayas manifiesta "La habilidad es el componente del contenido que refleja las realizaciones del hombre, en una rama del saber propia de la cultura. Es, desde el punto de vista psicológico, el sistema de acciones y operaciones dominados por el sujeto y que responden a un objetivo" (13).

Existen varios tipos de habilidades asociadas a distintas ramas del saber, pero es interés abordar solo aquellas que influyen en el campo de acción correspondiente a la problemática a tratar:

Primeramente se debe entender, qué son las **habilidades cognitivas**:

Son aquellas que se ponen en marcha para analizar y comprender la información recibida, cómo se procesa y cómo se estructura en la memoria. Desde el punto de vista cognitivo, se concibe el aprendizaje como un conjunto de procesos que tienen como objeto el procesamiento de la información (14).

El **razonamiento verbal** tiene un papel fundamental en el progreso cognitivo, ya que implica la habilidad para entender, estructurar, organizar y comprender el lenguaje, unas destrezas imprescindibles para alcanzar un buen desempeño en las diferentes etapas académicas y en la vida. Proporciona al hablante los medios necesarios para hacer un uso adecuado de la lengua y procesar la información que recibe de su entorno. No consiste sólo en tener un vocabulario amplio, sino que implica también la capacidad para clasificarlo, ordenarlo y relacionar entre sí los significados (15).

La habilidad para el razonamiento verbal no es una capacidad innata en los niños, aunque algunos tengan más potencial que otros. Lo fundamental es que cuenten con un entorno que



les ayude a desarrollarla. Una de las mejores formas de estimular el desarrollo de las habilidades lingüísticas de los niños es a través de los juegos verbales (15).

Juegos de palabras como crucigramas o sopas de letras son herramienta factibles para alcanzar el objetivo de desarrollar el razonamiento verbal. Estas actividades, realizadas de forma habitual, reportan muchos beneficios al hablante. Por una parte, favorecen la adquisición del vocabulario, les ayuda a organizarlo en torno a categorías y refuerzan su proceso cognitivo. También favorecen la fluidez de la expresión oral y el desarrollo de la memoria, además de servir para progresar en competencias del lenguaje tan importantes como la escritura, la ortografía o la redacción (15).

Juegos Lógicos Matemáticos (JLM)

Son medios didácticos u objetos de conocimientos que han sido creados para contribuir a estimular y motivar de manera divertida, participativa, orientadora y reglamentaria el desarrollo de habilidades, capacidades lógico-intelectuales y procesos de razonamiento analítico-sintético, inductivo-deductivo, concentración, entre otros beneficios. Educadores, psicólogos e investigadores sociales señalan que los JLM pueden convertirse en una poderosa herramienta formativa para estimular y motivar el proceso enseñanza- aprendizaje, si son incluidos en el proceso de formación del estudiante; pues no se trata de hacer “jugar” a niños y niñas de modo improvisado, sino de manera deliberada y planificada para lograr resultados. Entre los principales factores que se pueden destacar se encuentran (16):

- Favorecen la comprensión y uso de contenidos matemáticos en general y al desarrollo del pensamiento lógico en particular.
- Relacionan la matemática con una situación generadora de diversión.
- Permiten realizar cálculos mentales.
- Los practicantes adquieren flexibilidad y agilidad mental jugando.
- Promueven el ingenio, creatividad e imaginación.
- Estimulan el razonamiento inductivo-deductivo.
- Contribuyen al desarrollo de la perseverancia y la paciencia.

Dentro de los JLM se encuentran los siguientes:

Sudoku: ejercicio mental que ayuda a mejorar y potenciar una amplia gama de habilidades dentro de las que se encuentra:



- **Razonamiento lógico:** Es un proceso mental que implica la aplicación de la lógica. A partir de esta clase de razonamiento, se puede partir de una o de varias premisas para arribar a una conclusión que puede determinarse como verdadera, falsa o posible (17). Al rellenar las filas y columnas con los números sin poderlos repetirlos, se hace necesario el usar la lógica para resolver el sudoku (18).

Tablero Matemático: ejercicio que ayuda a mejorar y potenciar habilidades matemáticas tales como:

- **Razonamiento matemático:** “Consiste en la habilidad para utilizar y relacionar los números, sus operaciones básicas, los símbolos y las formas de expresión y razonamiento matemático, tanto para producir e interpretar distintos tipos de información, como para ampliar el conocimiento sobre aspectos cuantitativos y espaciales de la realidad, y para resolver problemas relacionados con la vida cotidiana y con el mundo laboral. Forma parte de la competencia matemática la habilidad para interpretar y expresar con claridad y precisión informaciones, datos y argumentaciones, lo que aumenta la posibilidad real de seguir aprendiendo a lo largo de la vida, tanto en el ámbito escolar o académico como fuera de él, y favorece la participación efectiva en la vida social...” (19).

1.5 Análisis de soluciones similares

Colección de juegos educativos GCompris

GCompris es un programa de cómputo educacional especialmente diseñado para niños de entre 2 y 10 años. Es software libre y está disponible para Windows, Mac y Linux. Contiene una colección formada por más de 100 actividades y juegos dedicados a tratar todo tipo de temas (20):

- Uso del ordenador y sus periféricos: acostumbrarse al teclado, ratón...
- Álgebra: memoria, números, tablas, imágenes...
- Ciencia: el ciclo del agua, electricidad, canales, el submarino...
- Geografía: países y mapas.
- Juegos: ajedrez, memoria, fútbol, rompecabezas...
- Lectura: diferentes ejercicios de lectura.
- Otros: las horas, dibujos vectoriales...



Sudoku, coloca símbolos únicos en un cuadrado - GCompris

El objetivo del sudoku es introducir un número del 1 al 9, letra o símbolos dependiendo del modo de juego donde cada uno de estos debe ser único en cada fila, columna y en cada región (subtableros de 3x3 casillas). Para el primer nivel utiliza símbolos de colores y para los niveles más altos, se pulsa en una casilla vacía para darle el foco del teclado para luego introducir un número o letra posible. El juego no deja introducir datos no válidos. Completar el puzle requiere paciencia y habilidad lógica.

EPasatiempos¹

EPasatiempos es un sitio web de entretenimiento en línea que contiene pasatiempos y juegos sobre variadas temáticas con el objetivo de ayudar a ejercitar la mente. Contiene una amplia gama de mini juegos donde se pueden encontrar: crucigramas, sudokus, sopas de letras, laberintos, juegos de diferencias, jeroglíficos, ahorcado, puzles, tableros matemáticos, buscaminas, juegos de reconocimiento visual y juegos de concentración los cuales se pueden realizar o bien obtener su versión para imprimir (21).

Sopas de letras – ePasatiempos

Los juegos de sopa de letras son uno de los disímiles pasatiempos que se pueden encontrar en la aplicación ePasatiempos. Al acceder a éste nos dan tres variantes a escoger para jugar (22):

- Sopa de letras Temático: Se brinda una lista de sopas de letras temáticas disponibles ordenadas por número y fecha de publicación, donde se muestra el tema que aborda, dándole la posibilidad de elegir al usuario la sopa de letras que desee realizar independiente de si realizó o no las anteriores a la escogida.
- Sopa de letras con Imágenes: Nos brinda un listado ordenado por número y fecha de publicación de las sopas de letras con imágenes disponibles. Al seleccionar alguna se muestra la sopa de letras con las imágenes representativas de las palabras a buscar alrededor de esta y un botón con las instrucciones.
- Sopa de letras por Palabras al azar: Nos brinda un listado ordenado por número y fecha de publicación de las sopas de letras que contienen palabras al azar.

Mico - La Sopa de Letras v.2.51

Mico - La Sopa de Letras tiene diversos modos de juego. Está desarrollado en idioma español, su licencia es gratuita, disponible para todas las versiones de Windows y su mayor virtud es la

¹ <http://epasatiempos.es>



variedad infinita de sopas de letras que contiene pues nunca se repite la misma dos veces. Este juego incluye más de 3.000 palabras agrupadas en 100 categorías, permite configurar el nivel de dificultad, incorpora algunos mini-juegos extra, ejemplos, y además la posibilidad de personalizar las categorías y palabras con añadidos propios (23).

Crucigramas - ePasatiempos

Los juegos de crucigramas de la aplicación ePasatiempos están dirigidos a ampliar la cultura, retar la memoria, ejercitar la mente y ayudar a prevenir enfermedades como el Alzheimer. Se muestra un listado ordenado por número y fecha de publicación de los crucigramas disponibles y el tamaño del mismo. Cada crucigrama contiene dos listas de definiciones numeradas por filas y columnas y agrupadas según la orientación además de los elementos de ayuda como las instrucciones y posibilita comprobar errores (24).

Crossword Express v.8.0b

Crossword Express contiene crucigramas de diferentes tipos, formas y temáticas; siendo, además de un entretenimiento, una buena forma de aprender vocabulario e incluso idiomas, pues el programa está solo en inglés. *Crossword Express* dispone de crucigramas de diferentes niveles de dificultad, así como de distintos tamaños. Aparte de los clásicos crucigramas de siempre, también viene integrado con un juego de puzzle en el que se deben buscar palabras ocultas (25).

Tableros matemáticos – ePasatiempos

Brinda la posibilidad de escoger el tablero matemático que se desee realizar dentro de un listado ordenado por número y fecha de publicación. El juego está compuesto por cuatro filas y cuatro columnas y entre éstas se describen los operadores matemáticos (suma, resta, multiplicación, división, igualdad), el objetivo es completar los espacios en blanco con números de forma tal que se cumplan todas las igualdades. Contiene además las instrucciones de cómo proceder, y dos opciones: una de comprobación y otra para ver la solución sin haber terminado el juego (26).

Sudokus - ePasatiempos

La aplicación ePasatiempos brinda dos tipos de variantes de sudokus (27):

- Sudokus clásicos: Muestra un listado con los sudokus disponibles. Cada sudoku cuenta con los elementos de ayuda donde se describen las reglas, métodos de resolución e instrucciones de uso que explican cómo dar solución al juego, además posee las opciones



de: comprobación que verifica si los números están ubicados correctamente (en caso contrario se muestran de color rojo las casillas incorrectas) y la de limpiar casilla para reiniciar desde cero el pasatiempo.

- Mini sudokus matemáticos: El mini sudoku matemático variante del sudoku pero más reducida se compone solamente de 3x3 casillas en lugar del tradicional 9x9. Sin embargo, y a pesar de las diferencias, conserva las reglas básicas que los definen: hay que colocar los números del 1 al 9, uno en cada casilla y sin repetir, y sólo hay una solución posible. Cada juego cuenta con elementos de ayuda como ¿en qué consiste?, ¿cómo se juega? ¿cómo se resuelve?, además de la opción de Comprobar.

Domo-Sudoku²

Sitio web de entretenimiento en línea que contiene disímiles juegos de sudoku y variantes de este. Posee distintos tipos de juego dentro de los cuales se puede seleccionar que sudoku realizar dentro de un rango de 0 hasta 9999 o simplemente resolver uno aleatorio: modo clásico (9x9 casillas), para niños (4x4 casillas), gordo (16x16 casillas), cada uno de estos con tres niveles de dificultad (fácil, medio, difícil). Contiene además las opciones de chequeo y para ver la solución (28).

Pure Sudoku v.1.8

Pure Sudoku es una versión gratuita en español para PC compatible con todas las distribuciones de Windows, es un juego que ayuda a la facilitación de la estrategia y es un modo de agilizar la mente.

Nos ofrece una serie de ventajas frente a la resolución tradicional del puzle en papel. Por ejemplo, poder borrar fácilmente los errores o muestra qué cifras no se puede situar en una determinada casilla (ésta última es opcional). También se puede elegir entre varios niveles de dificultad, crear nuestros propios sudokus y guardar nuestra partida para poder continuar con ella en otro momento (29).

Sudoku! Deluxe

Aplicación de escritorio gratuita, disponible para todas las distribuciones de Windows. Permite disfrutar de los famosos pasatiempos numéricos originarios de Japón, tanto a través de partidas en el PC, como desde pasatiempos impresos en papel.

² <http://domo-sudoku.com>



Cuenta con diez niveles de dificultad ideal para motivar a los principiantes y no defraudar a lo más expertos, brinda la posibilidad de jugar en pantalla completa, conectado o desconectado a internet, permite publicar tu nombre en las tablas de récords y puntuaciones, etc. (30).

Kakuro 1.0

Kakuro es una aplicación de escritorio gratuita y disponible para todas las versiones de Windows que propone un nuevo concepto de pasatiempos-rompecabezas, producto de la mezcla del concepto de los crucigramas y de los sudokus. Tiene como objetivo rellenar un tablero como el de un crucigrama, pero con números y sumas, en vez de palabras y definiciones. La dificultad del juego Kakuro está en que las sumas de las filas y de las columnas deben ser en cada caso la que esté indicada en el tablero. Existen diferentes niveles de dificultad, para elegir y si quedas atascado en cualquier momento, el juego incluye un completo sistema de ayuda (31).

La caja mágica

La caja mágica es una colección de software educativo desarrollada en la UCI por el proyecto Multisaber-Navegante la cual está compuesta por varios productos donde cada uno se encuentra integrado por diversos módulos. Uno de estos es el módulo Juegos que fue diseñado específicamente para apoyar la enseñanza y el aprendizaje en los alumnos de la educación primaria. Dentro de los juegos por los cuales está integrado dicho módulo se puede encontrar la Sopa de letras y el Crucigrama, cada uno como parte de una ejercitación a un contenido recogido en la misma aplicación. Los juegos de Sopa de letras y el Crucigrama permiten seleccionar la cantidad de actividades a realizar así como los temas a tratar brindándole la oportunidad al usuario de gestionar la dificultad del juego. El ejercicio no culminará satisfactoriamente hasta que no supere la meta establecida (cantidad de juegos a realizar).

Al analizar las diferentes soluciones anteriores se destacan varios factores positivos como:

- Posibilidad de guardar el progreso realizado en el juego para luego reanudarlo.
- Contador de tiempo para completar el juego.
- Varios niveles de dificultad para garantizar la complejidad del mismo.
- La capacidad de fomentar en el niño el razonamiento lógico, matemático y verbal, además retar y ejercitar la mente.

A su vez dichas soluciones cuentan con un conjunto de elementos negativos, entre las cuales se pueden encontrar:



- No son multiplataforma (ej. Kakuro, Sudoku! Deluxe, Pure Sudoku, Crossword Express).
- Requieren conexión a Internet (ej. Domo-Sudoku, ePasatiempos).
- No cuentan con una historia que motive al jugador a llegar al final del juego.
- No todos poseen niveles de dificultad asequibles a los niños.

Debido a que erradicar estas dificultades es casi imposible pues el código fuente de estos juegos no se encuentra disponible y que las tecnologías con las cuales están desarrollados no permiten o no son las más indicadas para garantizar su integración a la colección de juegos educativos MundoClick se decide no tomar ninguno de estos como solución y desarrollar los juegos Sopa de letras, Crucigrama, Sudoku y Pasatiempo matemático teniendo en cuenta factores positivos y negativos anteriormente descritos.

1.6 Tecnologías y herramientas

Las herramientas y tecnologías informáticas a utilizar en el desarrollo de los juegos para la colección de juegos educativos MundoClick fueron analizadas y definidas por el equipo de trabajo del Proyecto Multisaber-El Navegante. A cada herramienta se le realizará un estudio de sus principales características y su aporte como herramienta para desarrollar la propuesta de solución.

1.6.1 Metodología de desarrollo

Metodología es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software (32).

Las metodologías de desarrollo de software están divididas en dos grupos (33):

Las tradicionales:

- El cliente interactúa con el equipo de desarrollo mediante reuniones.
- Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos.
- Se promueve que la arquitectura se defina tempranamente en el proyecto.
- Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
- Se espera que no ocurran cambios de gran impacto durante el proyecto.

Las ágiles:

- Cliente es parte del equipo de desarrollo.



- Orientada a proyectos pequeños. Corta duración (o entregas frecuentes), equipos pequeños (< 10 integrantes) y trabajando en el mismo sitio.
- La arquitectura se va definiendo y mejorando a lo largo del proyecto.
- Se esperan cambios durante el proyecto.

La metodología que será utilizada en el proceso de desarrollo de los juegos de esta investigación es RUP. Debido a que fue la definida por el proyecto para el desarrollo de la colección de juegos educativos MundoClick que además permite obtener una documentación detallada del desarrollo de los juegos.

RUP

RUP (*Rational Unified Process* traducido al español como: Proceso Unificado de Desarrollo) tiene 2 dimensiones (34):

- El eje horizontal representa tiempo y demuestra los aspectos del ciclo de vida del proceso.
- El eje vertical representa las disciplinas que agrupan actividades definidas lógicamente por la naturaleza.

La primera dimensión representa el aspecto dinámico del proceso y expresa en términos de fases, de iteraciones y la finalización de las fases. La segunda representa el aspecto estático del proceso: como se describe en términos de componentes de proceso, las disciplinas, las actividades, los artefactos y los roles (34).

La metodología RUP divide el proceso de desarrollo de software en 4 fases y 9 flujos de trabajos como se muestra en la figura 1:

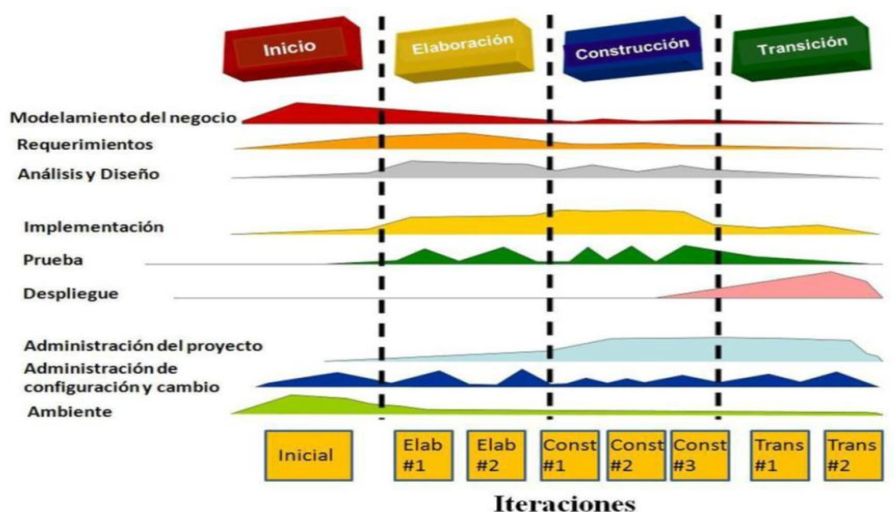


Figura 1. Fases y flujos de RUP.



Fases de RUP (35):

1. **Inicio:** Se desarrolla una descripción del producto final a partir de una buena idea y se presenta un análisis de negocio para el producto. Esencialmente esta fase responde a las siguientes preguntas: ¿Cuáles son las principales funciones del sistema para sus usuarios más importantes? , ¿Cómo podría ser la arquitectura del sistema?, ¿Cuál es el plan de proyecto y cuánto costará desarrollar el producto?
2. **Elaboración:** Se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura del sistema. El resultado de esta base es la línea base de la arquitectura.
3. **Construcción:** Durante esta fase se crea el producto. En esta fase la línea de la arquitectura crece hasta convertirse en el sistema completo.
4. **Transición:** En esta fase un grupo de usuarios con experiencia prueba el producto e informa de defectos y deficiencias. Los desarrolladores corrigen los problemas e incorporan algunas de las mejoras sugeridas en una versión general dirigida a la comunidad de usuarios.

Por cada una de estas fases se transita a través de los nueve flujos de trabajo representados anteriormente, para los que RUP propone una serie de artefactos, actividades y roles que varían de un flujo de trabajo a otro. Su ciclo de vida está caracterizado por ser (35):

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

1.6.2 Lenguaje Unificado Modelado

El lenguaje UML es diseñado para visualizar, especificar, construir y documentar software orientados a objetos. Este modelo es esencial en la construcción de un software porque permite:



- Comunicar la estructura de un sistema complejo.
- Especificar el comportamiento deseado del sistema.
- Comprender mejor lo que se está construyendo.
- Descubrir oportunidades de simplificación y reutilización.

UML estandariza 9 tipos de diagramas para representar gráficamente un sistema desde distintos tipos de vista (36).

Para poder cumplir los objetivos anteriormente planteados UML está compuesto por tres clases de bloques de construcción (37):

Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos y acciones).

Relaciones: Relacionan los elementos entre sí.

Diagramas: Son colecciones de elementos con sus relaciones.

1.6.3 Herramienta CASE de modelado

Las herramientas CASE (*Computer Aided Software Engineering* traducido al español como: Ingeniería de Software Asistida por Computación) son un conjunto de métodos, utilidades y técnicas que ayudan al desarrollo de software. Al utilizar estas herramientas se puede abstraer al código fuente en un nivel donde la arquitectura y el diseño son más fáciles de entender y modificar (38).

Estas herramientas pueden proveer muchos beneficios en todas las etapas del proceso de desarrollo de software, algunas de ellas son (39):

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

Algunas de las ventajas de estas herramientas (39):

- Facilidad para la revisión de aplicaciones.
- Soporte para el desarrollo de prototipos de sistemas.
- Generación de código.
- Soporte interactivo para el proceso de desarrollo.



La herramienta CASE a utilizar es Visual Paradigm for UML debido a que fue definida por el proyecto en que se enmarca la investigación.

Visual Paradigm for UML 8.0

Visual Paradigm for UML es una herramienta CASE profesional que soporta el ciclo de vida completo de desarrollo de software definido en RUP: análisis y diseño orientados a objetos, implementación, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (40).

1.6.4 Lenguajes de desarrollo

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 (41).

Los lenguajes que se emplean en las soluciones son CSS3, JavaScript, HTML 5 y JSON.

CSS3

CSS (*Cascade Style Sheet*, traducido al español como: hojas de estilo en cascada) es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación además es imprescindible para la creación de páginas web complejas (42).

CSS3 trae consigo nuevas características como por ejemplo:

- Tener bordes redondeados con fotos, sombras, etc.
- Se pueden tener múltiples imágenes de fondo con el atributo *background-image*.
- Textos multi-columna.

La novedad más importante que aporta CSS 3, de cara a los desarrolladores de webs, consiste en la incorporación de nuevos mecanismos para mantener un mayor control sobre el estilo con el que se muestran los elementos de las páginas, sin tener que recurrir a trucos o atajos, que a menudo complicaban el código de las web (43).

El uso de CSS 3 permitirá dar un estilo adecuado al juego y algunos elementos comprendidos en éste como son las pistas y los diferentes indicadores de nivel, tiempo y puntuación que se ubicaran en la parte inferior de la pantalla.



JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Es un lenguaje muy usado. Su inigualable popularidad como lenguaje de programación de aplicaciones web se ha extendido a otras aplicaciones y otros entornos no relacionados con la web (44).

Ninguno de los lenguajes de programación del mercado se puede considerar como el más adecuado para la creación de aplicaciones. Cada uno tiene sus ventajas e inconvenientes. Los últimos avances en JavaScript y la proliferación de tecnologías como HTML, Java y Macromedia Flash, han colocado a JavaScript en una posición ventajosa para aprovechar estas herramientas en la creación de grandes soluciones para la Web (45).

JavaScript permitirá un correcto flujo del juego controlando todos los eventos del mismo y validando la entrada y salida de información al sistema.

HTML5

HTML5 es la actualización de HTML (*Hyper Text Markup Language*, traducido al español como: lenguaje de marcado de hipertexto), el lenguaje en el que es creada la web. HTML5 también es un término de marketing para agrupar las nuevas tecnologías de desarrollo de aplicaciones web: HTML5, CSS3 y nuevas capacidades de JavaScript.

Flash en especial ha sido usado en reemplazo de HTML para desarrollar web apps que superaran las habilidades de un navegador: Audio, video, webcams, micrófonos, datos binarios, animaciones vectoriales, componentes de interfaz complejos, entre muchas otras cosas. Ahora HTML5 es capaz de hacer esto sin necesidad de *plugins* y con una gran compatibilidad entre navegadores (46).

Algunas reglas establecidas para HTML5 (47):

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

HTML5 permitirá hacer uso del *canvas* con el cual se podrá dibujar en la página HTML. También será utilizado para mostrar los diferentes mensajes de cada uno de los juegos.



JSON

JSON (*JavaScript Object Notation*, traducido al español Notación de Objetos JavaScript) es un formato de intercambio de datos de texto ligero. Es independiente del lenguaje, se "auto-describe" y es fácil de entender. JSON utiliza la sintaxis de JavaScript para la descripción de objetos de datos, pero sigue siendo un lenguaje con plataforma independiente. Existen analizadores y librerías JSON para muchos lenguajes de programación diferentes. Este se evalúa como objetos JavaScript (48).

JSON se basa en dos estructuras (49):

- Una colección de pares nombre / valor. En varios lenguajes, esto se conoce como un objeto, ficha, estructura, diccionario, tabla hash, lista de clave, o un arreglo de asociación.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se realiza en forma de arreglo, vector, lista o secuencia.

JSON será utilizado para almacenar los datos de cada uno de los juegos y también para guardar los datos relacionados con el estado del jugador tales como la puntuación, el nivel y la dificultad en la que se encuentra el mismo.

1.6.5 Frameworks y librerías

Un **framework** es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto (50).

Algunos de los objetivos de un *framework* son (51):

- Desarrollo rápido.
- Desarrollo estructurado.
- Reutilización de código.
- Disminuir el esfuerzo en el desarrollo.
- Aprovechar las funcionalidades ya implementadas.

En el marco del presente trabajo, se define **librería**, como una colección o conjunto de funciones usadas para desarrollar software (52).



Pointer

Pointer es una plataforma de desarrollo orientada a la implementación y despliegue de juegos basados en HTML 5, CSS 3 y JavaScript. La misma fue desarrollada por el equipo de trabajo de la línea de desarrollo de juegos educativos y simuladores del centro FORTES perteneciente a la UCI con el objetivo de apoyar la implementación de los juegos creando una abstracción de varias funcionalidades comunes entre ellos; esto permitirá acortar el tiempo de desarrollo, estandarizar gran parte del código y ofrecer un ambiente controlado para el funcionamiento de los mismos. Para su funcionamiento Pointer utiliza (53):

- AppJS v.0.20
- NodeJS v.0.8.11
- JQuery v.1.9
- HTML 5
- CSS 3
- JavaScript

Desde el punto de vista arquitectónico utiliza el estilo modelo-vista-controlador y hace uso de algunos patrones de diseño como Active Record para el acceso a datos y el Mediator para el encapsulamiento de las funcionalidades que brinda a los desarrolladores (53).

Bootstrap v.2.3.2

Bootstrap es un Framework desarrollado por el equipo de Twitter que tiene código abierto. Se utiliza en el desarrollo rápido y correcto de aplicaciones y sitios web. Fomenta las buenas prácticas de diseño y desarrollo web. Permite diseñar webs adaptables y fluidas, visualizables correctamente en múltiples dispositivos. Incluye una robusta base de HTML5, CSS3 y JavaScript (54).

También incluye un conjunto de recursos (estilos tipográficos, elementos de formulario, botones, tablas, barras de navegación, y más) que, si bien no se usa en las interfaces de un sitio web público, bien se pueden utilizar en el diseño de administradores, logrando una impronta muy profesional y economizando tiempo de diseño y maquetado (55).

JQuery v.1.10.1

JQuery es una librería de JavaScript rápida, pequeña y rica en funciones. Hace las cosas como documento HTML de recorrido y la manipulación, manejo de eventos, animación y Ajax mucho más simple con un API fácil de usar que funciona a través de una multitud de navegadores. Con



una combinación de versatilidad y capacidad de ampliación, jQuery ha cambiado la forma en que millones de personas escriben JavaScript (56).

La librería jQuery en resumen aporta las siguientes ventajas (57):

- Ahorra muchas líneas de código.
- Hace transparente el soporte de nuestra aplicación para los navegadores principales.
- Provee de un mecanismo para la captura de eventos.
- Provee un conjunto de funciones para animar el contenido de la página en forma muy sencilla.
- Integra funcionalidades para trabajar con AJAX.

NodeJS v.0.8.11

NodeJS 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 v8. Presenta como ventajas que se maneja a base de eventos capaz de escalar simultáneas conexiones a la vez, además trabajar de manera asincrónica (es decir en un segundo plano, mientras las solicitudes van llegando, otras se están ejecutando) y es muy rápido (58).

Node es especialmente bueno en aplicaciones web que necesiten una conexión persistente con el navegador del cliente. Mediante una serie de técnicas llamadas *comet*, puedes hacer una aplicación que envíe datos al usuario en tiempo real; es decir, que el navegador mantenga la conexión siempre abierta y reciba continuamente nuevos datos cuando los haya. Node encajaría perfectamente en aplicaciones como (59):

- Juegos online.
- Gestores de correo online: de esta manera teniendo el navegador abierto se pueden ver notificaciones en tiempo real de nuevos correos recibidos.
- Herramientas de colaboración.
- Chats.
- Redes sociales: por ejemplo para actualizar automáticamente tu muro de novedades.
- Herramientas de traducción en tiempo real.

CreateJS

CreateJS es un conjunto de librerías JavaScript que mejoran la experiencia de usuarios, este interesante recurso *open source* se compone de diferentes librerías especializadas para



trabajar con efectos de movimiento, trabajo con el *canvas* de HTML5, carga elementos de forma asíncrona o extracción de animaciones en formato swf. No tiene requerimientos previos y es compatible con los principales navegadores web, todas las librerías que se encuentran en CreateJS pueden ser usadas de forma individual o en conjunto (60).

Este conjunto de librerías será utilizado para pintar en la web todas las animaciones, figuras y elementos que conforman cada uno de los juegos.

AppJS v.0.20

AppJS es una aplicación multiplataforma, simple y potente que utiliza NodeJS para el desarrollo de aplicaciones de escritorio mediante los lenguajes: HTML, CSS y JavaScript, permitiendo además el uso de tecnologías asociadas a ellos como Bootstrap y JQuery. Dicha herramienta permite utilizar las API de HTML 5 para crear aplicaciones atractivas desde procesadores de texto hasta juegos en tres dimensiones (61).

1.6.6 Entorno de desarrollo

Un entorno de desarrollo integrado (IDE por sus siglas en inglés) es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Un IDE debe tener las siguientes características (62):

- Multiplataforma
- Soporte para diversos lenguajes de programación
- Integración con Sistemas de Control de Versiones
- Reconocimiento de Sintaxis
- Extensiones y Componentes para el IDE
- Integración con Framework populares
- Depurador
- Importar y Exportar proyectos
- Múltiples idiomas
- Manual de Usuarios y Ayuda

NetBeans IDE 7.4

NetBeans es un IDE que permite programar en distintos lenguajes, es ideal para trabajar con el lenguaje de desarrollo JAVA, además ofrece un excelente entorno para programar en PHP.



También se puede descargar una vez instalado NetBeans, los complementos para programar en C++. El IDE de NetBeans es perfecto y muy cómodo para los programadores pues tiene un aceptable completamiento de código (62).

NetBeans IDE 7.4 amplía el soporte avanzado de desarrollo HTML5 introducido en el NetBeans IDE 7.3 para las aplicaciones Java EE y PHP (63).

NetBeans brinda soporte para JQuery y los lenguajes como JavaScript, HTML, CSS, utilizados por la plataforma Pointer y además posee un óptimo completamiento de código para las versiones actualizadas de estos lenguajes.

1.7 Conclusiones del capítulo

Con la realización de este capítulo se pudo concluir que:

- El estudio detallado de los conceptos asociados al objeto de estudio demostró la necesidad de desarrollar los juegos Sopa de letras, Crucigrama, Sudoku y Pasatiempo matemático para dar solución al problema planteado.
- La búsqueda de soluciones similares a las de la investigación dentro del ámbito nacional e internacional arrojó que ninguna de las analizadas cumple con los requisitos necesarios para dar una solución rápida y viable al problema.
- La selección de RUP como metodología de desarrollo, UML como lenguaje de modelado, JavaScript, CSS 3 y HTML 5 como lenguajes para la programación, Bootstrap, jQuery, NodeJS, CreateJS como frameworks y librería, Visual Paradigm y NetBeans como herramientas CASE e IDE respectivamente, conformaron el entorno de desarrollo para dar solución al problema a resolver, garantizando la compatibilidad con la arquitectura de la plataforma Pointer y con la colección MundoClick.



Capítulo 2. Características del sistema

Introducción

En este capítulo se realiza una caracterización de los juegos que se desea implementar para dar solución a la investigación. Se identifican los requisitos funcionales y no funcionales con los que deben cumplir los juegos Sopa de Letras, Crucigrama, Sudoku y Pasatiempo Matemático. Se describen los conceptos fundamentales, las relaciones y los principales atributos que se manejan mediante un modelo conceptual. También se analizan los actores involucrados y se identifican, describen y modelan los casos de uso.

2.1 Juegos propuestos

A continuación se describen los juegos Sopa de Letras, Crucigrama, Sudoku y Pasatiempo Matemático propuestos como solución al problema de contribuir al desarrollo de habilidades de análisis, cálculo numérico y clasificación de palabras en los niños desde la colección de juegos educativos MundoClick.

2.1.1 Juego “Sopa de letras”

El juego consiste en una travesía marítima donde el usuario es el capitán de una flota compuesta por dos Calaveras, un Galeón, una Fragata y un Nao (embarcación de transporte de mercancías). La flota llega a una isla en el medio del Caribe y el capitán decide hacer de esa isla una ciudad. Para esto debe construir diferentes edificaciones que conforman la vida social de la ciudad y ayudan a mejorar las condiciones de vida en la isla. Para completar las edificaciones (granja, escuela, hospital, iglesia, etc.) se deben solucionar distintas sopas de letras relacionadas con la edificación a construir.

El juego tiene 2 niveles de dificultad:

- Marinero: la matriz de la sopa es de 12x12 y tienes que buscar 8 palabras por nivel, el listado de palabras a buscar es con letras.
- Capitán: la matriz de la sopa es de 12x12 y tienes que buscar 8 palabras por nivel, el listado de palabras a buscar es con imágenes.

Cada dificultad cuenta con 7 niveles donde cada uno se relaciona con una edificación distinta que trae consigo una sopa, con palabras relacionadas a esa edificación.

Al final de cada nivel se muestra la puntuación obtenida en el mismo más la puntuación de bonificación si termina la sopa en menos de 3 minutos y si usa o no la pista del nivel.



Al finalizar el juego se mostrará la puntuación obtenida durante todo el juego.

2.1.2 Juego “Crucigrama”

El juego trata de un niño que tiene un robot de juguete y quiere que sea su amigo. Desea que se convierta en un niño de verdad para poder jugar con él. Durante la noche lo visita un hada y le dice que para concederle el deseo tiene que enseñarle correctamente el idioma español al robot. Esto debe realizarlo mediante un juego y cuando el robot domine el idioma se convertirá en un niño de verdad y podrá jugar con él miles de juegos divertidos. La forma que el niño encontró para enseñarle español a su juguete fue mediante la solución de crucigramas, donde cada crucigrama le enseñaba una clasificación distinta de palabras.

El juego posee 2 niveles de dificultad:

- FÁCIL: Se muestran todas las consonantes por las que están compuestas las palabras a buscar y el jugador solo tiene que introducir las vocales para completarlas y así resolver el crucigrama.
- Normal: No se muestra ningún dato y el jugador tiene que introducir todas las letras de las palabras para completar el crucigrama.

Cada nivel de dificultad cuenta con 7 niveles donde cada uno es un crucigrama distinto relacionado con una clasificación distinta de las palabras. El usuario pasará de nivel al solucionar correctamente cada crucigrama.

Al final de cada nivel se muestra la puntuación obtenida en el mismo. Adicionalmente, otorga una puntuación de bonificación, si termina el crucigrama antes de 10 minutos y si no usa las pistas de cada nivel.

Al finalizar se mostrará la puntuación total obtenida durante todo el juego.

2.1.3 Juego “Sudoku”

El juego consiste en un viaje espacial donde el jugador es un astronauta que se encuentra en el planeta Neptuno con su nave averiada y necesita regresar a la Tierra para reencontrarse con su familia. Pero para lograr su objetivo debe reparar su nave mediante la solución de distintos sudokus antes de que se le agote el aire con el que dispone en sus tanques de oxígeno para reparar su nave (límite de tiempo).

El usuario solo pasará de nivel al culminar el sudoku correctamente. El juego está estructurado por 8 niveles, cada nivel es en un planeta distinto donde tiene que reparar su nave para poder continuar con su viaje. Los sudokus tienen 4 niveles de dificultad:



- Figuritas Espaciales: el tablero tiene una matriz de 4x4 y el jugador tiene que colocar 4 figuritas en el tablero sin que se repitan.
- Figuritas Galácticas: el tablero tiene una matriz de 9x9 y el jugador tiene que colocar 9 figuritas en el tablero sin que se repitan.
- Números Galácticos: el tablero tiene una matriz de 9x9 y el jugador tiene que colocar los números del 1 al 9 sin que se repitan.
- Números Espaciales: el tablero tiene una matriz de 4x4 y el jugador tiene que colocar los números del 1 al 4 sin que se repitan.

Al final de cada nivel se muestra la puntuación obtenida en el mismo más la puntuación de bonificación si termina el sudoku antes de que finalice el tiempo.

Al finalizar se mostrará la puntuación total obtenida durante todo el juego.

2.1.4 Juego “Pasatiempo Matemático”

El juego se basa en un niño que le encanta las matemáticas y desea ser el mejor matemático del mundo. Su profesora al enterarse de su deseo le dice que para cumplir su sueño necesita realizar los 8 tableros matemáticos del mítico libro de Albert Einstein que nadie en el mundo ha logrado resolver.

El jugador dispone de un tiempo límite para completar cada tablero y solo pasará de nivel al culminar el mismo correctamente. Al finalizar cada tablero se guardará la puntuación obtenida por el jugador y se aumentará el coeficiente matemático (porcentaje del juego completado).

El juego tiene 2 niveles de dificultad:

- Aprendiz: Todas las igualdades tienen una sola operación.
- Genio: Todas las igualdades están compuestas por 2 operaciones matemáticas.

Al finalizar el juego se muestra la puntuación total obtenida durante todo el transcurso del mismo.

2.2 Especificación de los requisitos del software

Para que un proceso de software sea exitoso es imprescindible una descripción detallada, correcta y exhaustiva de los requisitos, ya que es necesario saber qué debe hacer el sistema. Los requisitos expresan lo que los usuarios quieren del producto, estos son especificados por escrito como parte de un documento formal. Los requisitos (o requerimientos) son condiciones que deben estar presentes en el software, estos se clasifican en: funcionales y no funcionales.



Los requisitos funcionales definen las acciones que el sistema será capaz de realizar, son capacidades o condiciones que debe cumplir para satisfacer las necesidades funcionales del cliente. Los requisitos no funcionales describen aquellas propiedades o cualidades del sistema una vez que esté finalizado, como por ejemplo, el rendimiento, interfaces de usuario, fiabilidad, mantenimiento, seguridad, portabilidad y estándares (35).

A continuación se muestran los requisitos funcionales y no funcionales de cada uno de los juegos a desarrollar.

2.2.1 Requisitos Funcionales:

Juego “Sopa de letras”

RF 1: Iniciar juego: El sistema debe mostrar la historia del juego y permitirle al jugador comenzar a jugar.

RF 1.1: Seleccionar modo de juego: El sistema debe permitirle al jugador escoger el tipo de sopa de palabras a resolver.

RF 2: Mostrar información del juego: el sistema debe mostrar información relacionada con el juego.

RF 2.1: Mostrar instrucciones del juego: El sistema debe brindarle al jugador instrucciones de cómo jugar (nivel de entrenamiento) antes de iniciar el juego.

RF 3: Crear entorno por nivel: el sistema debe ser capaz de cargar todos los elementos que corresponden al nivel en que se encuentra el jugador.

RF 3.1: Crear matriz de la sopa de letras: el sistema debe mostrar la matriz de la sopa definiendo el tamaño dado el modo de juego.

RF 3.2: Manejar los datos de la matriz de la sopa de letras: el sistema debe cargar y ubicar en la matriz las palabras correspondientes al nivel y llenar los espacios vacíos con letras aleatorias.

RF 3.3: Cargar estado del jugador: el sistema debe permitirle al jugador continuar jugando a partir del nivel en que se encontraba (solo si el jugador ya jugó con anterioridad y no culminó el juego) o comenzar una partida desde el principio.

RF 4: Realizar sopa de letras: el sistema debe permitirle al jugador realizar la sopa de letras.



RF 4.1: Seleccionar palabra: el sistema debe permitirle al jugador seleccionar palabras en la matriz marcando donde comienza la misma y luego en donde termina dicha palabra.

RF 4.2: Evaluar respuesta: El sistema debe evaluar si la palabra seleccionada por el usuario forma parte del conjunto de solución de la sopa (de ser así esta es resaltada).

RF 4.3: Registrar Tiempo: El sistema debe mostrarle al jugador el tiempo transcurrido.

RF 4.4: Registrar Puntuación: El sistema debe registrar los puntos del jugador en cada nivel.

RF 4.5: Cambiar de nivel: El sistema debe permitirle al jugador pasar al próximo nivel una vez vencido el actual.

RF 4.6: Otorgar bonificación: El sistema debe sumar una bonificación a los puntos obtenidos por el jugador si este cumplió con los requisitos para obtener la misma (bonifica 5 puntos por no utilizar la pista y 20 por terminar antes de 3 minutos).

RF 4.7: Otorgar pista: El sistema debe brindarle al jugador una pista por nivel dependiendo si el jugador cumplió con la condición para obtener esta (descubrir 3 palabras en menos de un minuto).

RF 5: Mostrar resultados finales del juego: El sistema debe mostrarle al jugador al finalizar el juego las conclusiones del mismo.

RF 5.1: Mostrar jugador ganador: El sistema debe mostrar el nombre jugador ganador al finalizar el juego.

RF 5.2: Mostrar puntuación obtenida por el jugador: El sistema debe mostrar la puntuación general obtenida por el jugador. Si la partida es modo multijugador debe mostrar las posiciones obtenidas por cada uno según dicha puntuación.

RF 6: Guardar estado del juego: El sistema debe permitirle al jugador que su estado (nivel en que se encuentra y puntuación hasta el momento) se guarde en el momento que desee salir del juego.

RF 7: Terminar juego: El sistema debe permitirle al jugador terminar el juego en el momento que lo desee.

RF 8: Jugar varios usuarios: El sistema debe permitir jugar hasta 4 jugadores por turnos en una misma PC.



Juego “Crucigrama”

RF 1: Iniciar juego: El sistema debe mostrar la historia del juego y permitirle al jugador comenzar a jugar.

RF 1.1: Seleccionar la dificultad de juego: El sistema debe permitirle al jugador escoger la dificultad de los crucigramas a resolver (fácil, normal).

RF 2: Mostrar información del juego: el sistema debe mostrar información relacionada con el juego.

RF 2.1: Mostrar instrucciones del juego: El sistema debe brindarle al jugador instrucciones de cómo jugar (nivel de entrenamiento) antes de iniciar el juego.

RF 3: Crear entorno por nivel: el sistema debe ser capaz de cargar todos los elementos que corresponden al nivel en que se encuentra el jugador.

RF 3.1: Crear matriz del crucigrama: el sistema debe crear la matriz del crucigrama de acuerdo a las dimensiones que posee cada nivel y dibujar solo las posiciones donde se encuentra una palabra.

RF 3.2: Manejar los datos del crucigrama: el sistema debe mostrar la descripción de la palabra a descubrir y la orientación de la misma al colocar el puntero en la posición inicial de ésta.

RF 3.3: Cargar estado del jugador: el sistema debe permitirle al jugador continuar jugando a partir del nivel en que se encontraba (solo si el jugador ya jugó con anterioridad y no culminó el juego) o comenzar una partida desde el principio.

RF 3.4: Crear alfabeto: el sistema debe mostrar un alfabeto digital donde el jugador puede seleccionar la letra a ubicarlas en la casi seleccionada.

RF 4: Realizar crucigrama: el sistema debe permitirle al jugador realizar el crucigrama.

RF 4.1: Ubicar letra: el sistema debe permitirle al jugador ubicar la letra que desee en la casilla seleccionada tanto por teclado como utilizando el alfabeto del juego.

RF 4.2: Evaluar respuesta: El sistema debe evaluar si la palabra entrada por el jugador se corresponde con la solución (de ser así ésta es resaltada).

RF 4.3: Registrar Tiempo: El sistema debe mostrarle al jugador el tiempo transcurrido.

RF 4.4: Registrar Puntuación: El sistema debe registrar los puntos del jugador en cada nivel.



RF 4.5: Cambiar de nivel: El sistema debe permitirle al jugador pasar al próximo nivel una vez vencido el actual.

RF 4.6: Otorgar bonificación: El sistema debe sumar una bonificación a los puntos obtenidos por el jugador si este cumplió con los requisitos para obtener la misma (bonifica 5 puntos por cada pista sin utilizar y 20 por terminar antes de 10 minutos).

RF 4.7: Utilizar pista: El sistema debe brindarle al jugador tres pistas por nivel y ubicar una letra aleatoria en una casilla vacía al utilizar una pista.

RF 5: Mostrar resultados finales del juego: El sistema debe mostrarle al jugador al finalizar el juego las conclusiones del mismo.

RF 5.1: Mostrar jugador ganador: El sistema debe mostrar el nombre jugador ganador al finalizar el juego.

RF 5.2: Mostrar puntuación obtenida por el jugador: El sistema debe mostrar la puntuación general obtenida por el jugador. Si la partida es modo multijugador debe mostrar las posiciones obtenidas por cada uno según dicha puntuación.

RF 6: Guardar estado del juego: El sistema debe permitirle al jugador que su estado (nivel en que se encuentra y puntuación hasta el momento) se guarde en el momento que desee salir del juego.

RF 7: Terminar juego: El sistema debe permitirle al jugador terminar el juego en el momento que lo desee.

RF 8: Jugar varios usuarios: El sistema debe permitir jugar hasta 4 jugadores por turnos en una misma PC.

Juego “Sudoku”

RF 1: Iniciar juego: El sistema debe mostrar la historia del juego y permitirle al jugador comenzar a jugar.

RF 1.1: Seleccionar modo de juego: El sistema debe permitirle al jugador escoger el tipo de sudoku a resolver.

RF 2: Mostrar información del juego: el sistema debe mostrar información relacionada con el juego.

RF 2.1: Mostrar instrucciones del juego: El sistema debe brindarle al jugador instrucciones de cómo jugar (nivel de entrenamiento) antes de iniciar el juego.



RF 3: Crear entorno por nivel: el sistema debe ser capaz de cargar todos los elementos que corresponden al nivel en que se encuentra el jugador.

RF 3.1: Crear matriz del sudoku: el sistema debe mostrar la matriz del sudoku definiendo el tamaño dado el modo de juego.

RF 3.2: Manejar los datos de la matriz del sudoku: el sistema debe cargar y ubicar en la matriz los números o símbolos y espacios en blanco correspondientes al nivel.

RF 3.3: Cargar estado del jugador: el sistema debe permitirle al jugador continuar jugando a partir del nivel en que se encontraba (solo si el jugador ya jugó con anterioridad y no culminó el juego) o comenzar una partida desde el principio.

RF 4: Realizar sudoku: el sistema debe permitirle al jugador realizar el sudoku propuesto.

RF 4.1: Ubicar números o símbolos introducidos por el usuario: el sistema debe permitirle al jugador seleccionar y ubicar los números o símbolos correspondientes al nivel en la casilla vacía deseada de la matriz.

RF 4.2: Evaluar respuesta: El sistema debe evaluar la respuesta del jugador, notificándole la evaluación sea parcial o general (si el número o símbolo ubicado entra en conflicto con otro este se resalta en rojo, de lo contrario verde).

RF 4.3: Registrar Tiempo: El sistema debe mostrarle al jugador el tiempo transcurrido.

RF 4.4: Registrar Puntuación: El sistema debe registrar los puntos del jugador en cada nivel.

RF 4.5: Cambiar de nivel: El sistema debe permitirle al jugador pasar al próximo nivel una vez vencido el actual.

RF 4.6: Otorgar bonificación: El sistema debe sumar una bonificación a los puntos obtenidos por el jugador si este cumplió con los requisitos para obtener la misma (bonifica 20 puntos por terminar antes de 7 minutos).

RF 5: Mostrar resultados finales del juego: El sistema debe mostrarle al jugador al finalizar el juego las conclusiones del mismo.

RF 5.1 Mostrar jugador ganador: El sistema debe mostrar el nombre del jugador ganador al finalizar el juego.



RF 5.2 Mostrar puntuación obtenida por el jugador: El sistema debe mostrar la puntuación general obtenida por el jugador. Si la partida es modo multijugador debe mostrar las posiciones obtenidas por cada uno según dicha puntuación.

RF 6: Guardar estado del juego: El sistema debe permitirle al jugador que su estado (modo de juego, nivel en que se encuentra y puntuación hasta el momento) se guarde en el momento que desee salir del juego.

RF 7: Terminar juego: El sistema debe permitirle al jugador terminar el juego en el momento que lo desee.

RF 8: Jugar varios usuarios: El sistema debe permitir jugar hasta 4 jugadores por turnos en una misma PC.

Juego “Pasatiempo Matemático”

RF 1: Iniciar juego: El sistema debe mostrar la historia del juego y permitirle al jugador comenzar a jugar.

RF 1.1: Seleccionar el modo de juego: El sistema debe permitirle al jugador escoger el modo de juego de los pasatiempos matemáticos a resolver (aprendiz o genio).

RF 2: Mostrar información del juego: el sistema debe mostrar información relacionada con el juego.

RF 2.1: Mostrar instrucciones del juego: El sistema debe brindarle al jugador instrucciones de cómo jugar (nivel de entrenamiento) antes de iniciar el juego.

RF 3: Crear entorno por nivel: el sistema debe ser capaz de cargar todos los elementos que corresponden al nivel en que se encuentra el jugador.

RF 3.1: Crear matriz del pasatiempo matemático: el sistema debe crear la matriz del pasatiempo matemático de acuerdo a las dimensiones que posee cada modo de juego.

RF 3.2: Manejar los datos del pasatiempo matemático: el sistema debe cargar y ubicar en la matriz los números, espacios en blanco y operadores matemáticos correspondientes al nivel.

RF 3.3: Cargar estado del jugador: el sistema debe permitirle al jugador continuar jugando a partir del nivel en que se encontraba (solo si el jugador ya jugó con anterioridad y no culminó el juego) o comenzar una partida desde el principio.



RF 3.4: Crear tablero numérico: el sistema debe mostrar un tablero numérico donde el jugador pueda seleccionar los números a ubicar en la matriz.

RF 4: Realizar pasatiempo matemático: el sistema debe permitirle al jugador realizar el pasatiempo matemático.

RF 4.1: Ubicar número: el sistema debe permitirle al jugador ubicar el número seleccionado tanto por teclado como utilizando el tablero numérico en la casilla vacía que desee.

RF 4.2: Evaluar respuesta: El sistema debe evaluar si los números entrados por el jugador satisfacen las operaciones matemáticas (si se satisfacen las operaciones matemáticas los resultados se enmarcan en un cuadrado de color verde).

RF 4.3: Registrar Tiempo: El sistema debe mostrarle al jugador el tiempo transcurrido.

RF 4.4: Registrar Puntuación: El sistema debe registrar los puntos del jugador en cada nivel.

RF 4.5: Cambiar de nivel: El sistema debe permitirle al jugador pasar al próximo nivel una vez vencido el actual.

RF 4.6: Otorgar bonificación: El sistema debe sumar una bonificación a los puntos obtenidos por el jugador si este cumplió con los requisitos para obtener la misma (bonifica 20 puntos por terminar antes de 3 minutos).

RF 5: Mostrar resultados finales del juego: El sistema debe mostrarle al jugador al finalizar el juego las conclusiones del mismo.

RF 5.1: Mostrar jugador ganador: El sistema debe mostrar el nombre jugador ganador al finalizar el juego.

RF 5.2: Mostrar puntuación obtenida por el jugador: El sistema debe mostrar la puntuación general obtenida por el jugador. Si la partida es modo multijugador debe mostrar las posiciones obtenidas por cada uno según dicha puntuación.

RF 6: Guardar estado del juego: El sistema debe permitirle al jugador que su estado (nivel en que se encuentra y puntuación hasta el momento) se guarde en el momento que desee salir del juego.

RF 7: Terminar juego: El sistema debe permitirle al jugador terminar el juego en el momento que lo desee.



RF 8: Jugar varios usuarios: El sistema debe permitir jugar hasta 4 jugadores por turnos en una misma PC.

2.2.2 Requisitos no funcionales:

RNF 1: Requisito de portabilidad: El sistema podrá ser utilizado por los sistemas operativos Windows XP o una versión superior y por Linux la distribución Ubuntu12.04 o superior.

RNF 2: Requisito de hardware.

- Requiere 500MB de espacio libre en el disco duro.
- Como mínimo 512 MB de RAM (recomendado 1 GB de RAM o mayor).
- Soporte de video que admita resolución de al menos 1024x780px y 24 bits.

RNF 3: Requisitos de interfaz.

La interfaz del sistema debe presentar rasgos de diseño infantiles, predominando el uso de colores atractivos para captar la atención del niño e imágenes en lugar de textos para sugerir las acciones.

RNF 4: Requisito de soporte.

Se realizará la transferencia tecnológica de la colección a los clientes.

RNF 5: Requisito de usabilidad.

El software será usado por los niños con edad comprendida entre 6 y 12 años.

RNF 6: Restricciones de diseño e implementación.

- Lenguaje de desarrollo: JavaScript.
- Lenguaje de marcado: HTML 5.
- Hoja de estilo: CSS 3.
- Herramientas de desarrollo: jQuery V1.9 y NodeJS v0.8.11.
- Entorno de Desarrollo Integrado: NetBeans v7.4.
- Herramienta CASE: Visual Paradigm.

2.3 Modelo conceptual

Un modelo conceptual explica los conceptos más significativos en un dominio del problema, identificando los atributos y las asociaciones. Representa elementos del mundo real, no componentes del software (37).



A continuación se muestran los modelos conceptuales de cada uno de los juegos y la descripción de los conceptos que lo componen:

Juego “Sopa de letras”

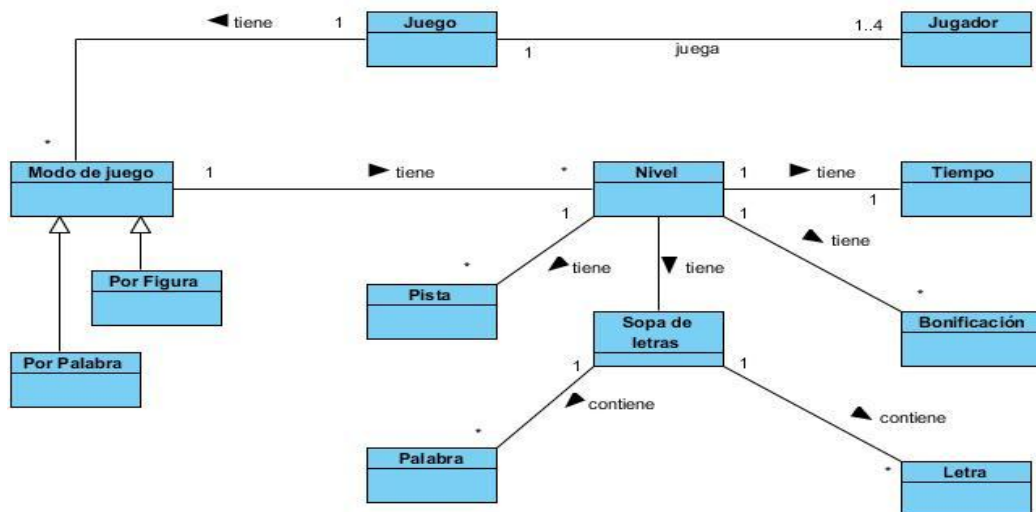


Figura 2. Modelo conceptual del juego Sopa de letras.

Descripción de los conceptos:

Juego: Conjunto de niveles en los que se muestran cada sopa de letras que el jugador debe resolver.

Jugador: Usuario que interactúa con el juego Sopa de letras.

Modo de juego: forma de buscar las palabras en la sopa de letras. Puede ser por figuras o conociendo la palabra.

Por Figura: modo de juego que muestra una imagen por cada palabra a buscar en la sopa de letras.

Por Palabras: modo de juego que muestra un listado de las palabras a buscar en la sopa de letras.

Nivel: Cada nivel representa un desafío diferente donde aumenta la complejidad del juego a medida que el jugador va avanzando.

Sopa de letras: Estructura donde se representan los elementos del juego.

Tiempo: Contador que marca el tiempo transcurrido que el jugador tarda en darle solución a una sopa de letras.



Bonificación: Puntos adicionales que recibe el jugador por cumplir ciertas condiciones durante el juego.

Pista: Elemento de ayuda que se le da al jugador como premio por una ágil ejecución del juego.

Palabras: Elementos a descubrir por el jugador que se encuentran contenidos en las sopas de letras (matrices).

Letras: Elementos que son ubicados aleatoriamente en los espacios de la matriz que no corresponden con una palabra.

Juego “Crucigrama”

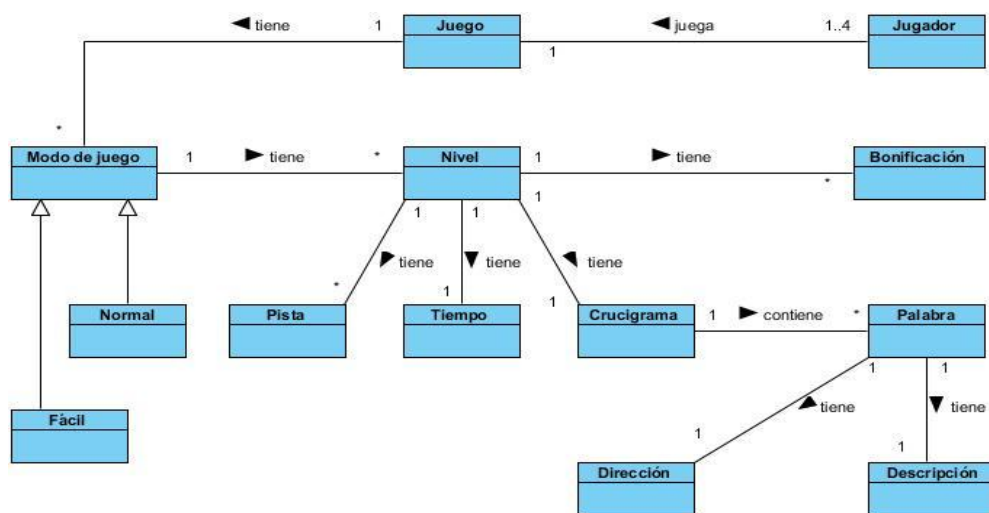


Figura 3. Modelo conceptual del juego Crucigrama.

Descripción de los conceptos:

Juego: Conjunto de niveles en los que se muestran cada uno de los crucigramas que el jugador debe resolver.

Jugador: Usuario que interactúa con el juego Crucigrama.

Modo de juego: dificultad del crucigrama. Puede ser modo fácil o normal.

Fácil: modo de juego que muestra los crucigramas con las consonantes ya colocadas para que el jugador complete las palabras solo utilizando las vocales.

Normal: modo de juego que muestra solamente la cantidad de letras que forman la palabra y su descripción.

Nivel: Cada nivel representa un desafío diferente donde aumenta la complejidad del juego a medida que el jugador va avanzando.



Crucigrama: Estructura donde se representan los elementos del juego.

Palabras: Elementos a descubrir por el jugador que se encuentra completando el crucigrama.

Tiempo: Contador que marca el tiempo transcurrido que el jugador tarda en darle solución a un crucigrama.

Bonificación: Puntos adicionales que recibe el jugador por cumplir ciertas condiciones durante el juego.

Pista: Elemento de ayuda que se le da al jugador como para facilitarle un poco el juego.

Descripción: Datos, significados, conceptos dados al jugador para que pueda descubrir la palabra a la cual hace referencia la descripción.

Dirección: Sentido en el cual está ubicada la palabra en la matriz: horizontal o vertical.

Juego “Sudoku”

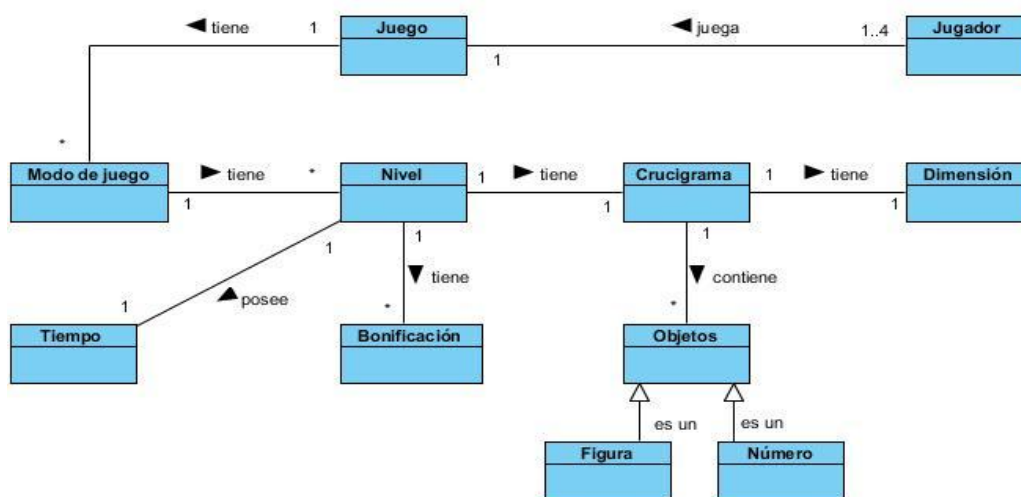


Figura 4. Modelo conceptual del juego Sudoku.

Descripción de los conceptos:

Juego: Conjunto de niveles en los que se muestran cada uno de los sudokus que el jugador debe resolver.

Jugador: Usuario que interactúa con el juego Sudoku.

Modo de juego: forma de jugar el sudoku. Varía según el tamaño de la matriz y los objetos a colocar en ella.



Nivel: Cada nivel representa un desafío diferente donde aumenta la complejidad del juego a medida que el jugador va avanzando.

Sudoku: Estructura donde se representan los objetos.

Objetos: Elementos contenidos en la matriz y utilizados para completar el sudoku.

Número: Elemento utilizado para completar el sudoku.

Figura: Elemento utilizado para completar el sudoku.

Dimensión: Tamaño de la matriz del sudoku.

Tiempo: Delimita un intervalo de tiempo durante el cual el jugador debe resolver un sudoku en cada nivel.

Bonificación: Puntos adicionales que recibe el jugador por cumplir ciertas condiciones durante el juego.

Juego “Pasatiempo Matemático”

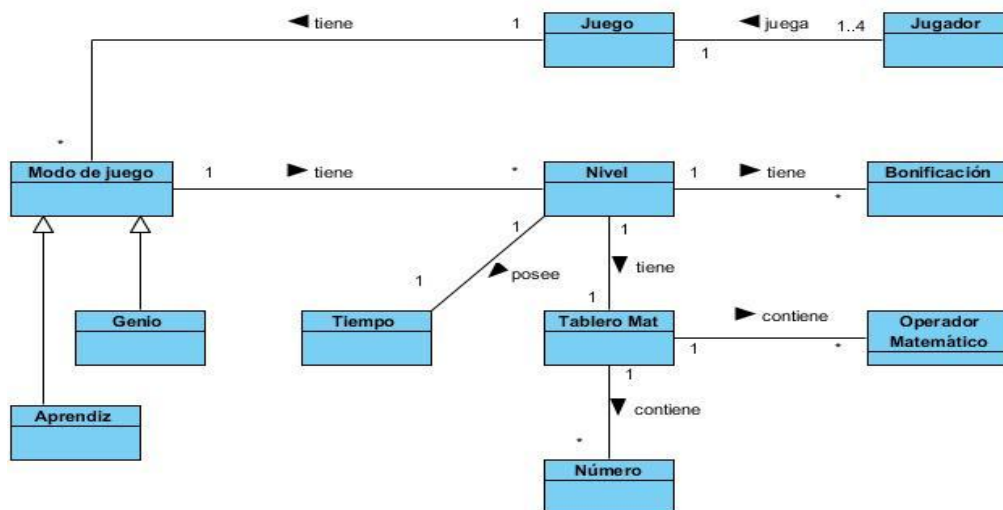


Figura 5. Modelo conceptual del juego Pasatiempo Matemático.

Descripción de los conceptos:

Juego: Conjunto de niveles en los que se muestran cada uno de los tableros matemáticos que el jugador debe resolver.

Jugador: Usuario que interactúa con el juego Pasatiempo Matemático.

Nivel: Cada nivel representa un desafío diferente donde aumenta la complejidad del juego a medida que el jugador va avanzando.



Tablero matemático: Estructura donde se representan los objetos.

Número: Elemento con el cual interactúa el jugador y lo posiciona en las casillas vacías de la matriz.

Operador Matemático: Elemento matemático que contiene la matriz puede ser: suma, resta, multiplicación, división e igualdad.

Dimensión: Tamaño de la matriz del tablero matemático.

Tiempo: Contador que marca el tiempo transcurrido que el jugador tarda en darle solución a un pasatiempo matemático.

Bonificación: Puntos adicionales que recibe el jugador por cumplir ciertas condiciones durante el juego.

2.4 Modelo de casos de uso del sistema

Describen los requerimientos funcionales del sistema en forma de casos de uso (CU). Especifica una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto. Explica gráficamente un conjunto de casos de uso de un sistema, los actores y la relación entre estos y los casos de uso. Tiene por objeto ofrecer una clase de diagrama contextual que permite conocer rápidamente los actores externos de un sistema y las formas básicas en que lo utilizan (35).

A continuación se presentan los diagramas de casos de uso de cada juego y la descripción de un CU. Las restantes descripciones se pueden encontrar en los anexos del presente documento.

Juego “Sudoku”

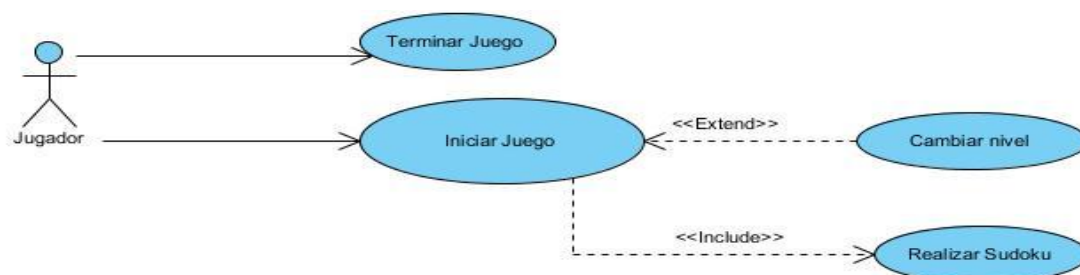


Figura 6. Diagrama de casos de uso del juego Sudoku.



Tabla 1. Descripción del caso de uso “Iniciar juego” del juego Sudoku.

Caso de Uso	<i>Iniciar Juego</i>	
Actores	<i>Jugador</i>	
Resumen	<i>El caso de uso inicia una vez que el jugador selecciona el juego. El sistema carga el juego, muestra el comic inicial y la pantalla principal del juego. Si el jugador selecciona la opción salir el sistema salva los datos y termina juego, finalizando así el caso de uso.</i>	
Complejidad	<i>Alta</i>	
Prioridad	<i>Crítico</i>	
Precondiciones	<i>El jugador debe estar autenticado en la aplicación.</i>	
Postcondiciones	<i>Se inició el juego.</i>	
Flujo normal de eventos		
	Actor	Sistema
	1. Selecciona el juego.	<p>1.1 El sistema analiza el jugador (es) y brinda las siguientes opciones:</p> <ul style="list-style-type: none"> • Si el jugador ha jugado anteriormente el juego y no ha culminado todos los niveles muestra una interfaz con el nivel actual en que se encuentra y las siguientes opciones: <ul style="list-style-type: none"> - “Continuar”: ir al paso 2.1 de la sección 2 “Comenzar a jugar”. - “Comenzar desde el principio”: ir al paso 2 del flujo básico. • Si hay más de un jugador autenticados en el sistema habilita el nivel multijugador: ir a la sección 3 “Multijugador”. • Si el jugador no ha jugado el sistema permite comenzar desde el principio.
	2. Selecciona la opción “Comenzar desde el principio”.	<p>a. Muestra la historia del juego en forma de comic.</p> <p>b. Muestra una interfaz que permite al jugador seleccionar una de las siguientes opciones:</p> <ul style="list-style-type: none"> • Nivel de entrenamiento: ir a la sección 1 “Nivel de entrenamiento”. • Jugar: ir a la sección 2 “Comenzar a jugar”.
Sección 1: “Nivel de entrenamiento”: Flujo Normal de Eventos		
	Actor	Sistema



1. Selecciona la opción “Nivel de entrenamiento”.	<p>1.1 Muestra un sudoku a resolver.</p> <p>1.2 Muestra una serie de instrucciones y pasos que el jugador debe seguir para completar el sudoku.</p>
2. Completa el sudoku siguiendo las instrucciones.	<p>2.1 Muestra un mensaje felicitando al jugador por haber resuelto el sudoku.</p> <p>2.2 Permite seleccionar el modo de juego: ir a la sección 2 “Comenzar a jugar”, paso 1.1.</p> <p>2.3 Termina el caso de uso.</p>
Sección 2: “Comenzar a jugar”: Flujo Normal de Eventos	
Actor	Sistema
1. Selecciona la opción “Iniciar juego”.	<p>1.1 Muestra una interfaz que permite al jugador seleccionar el modo de juego:</p> <ul style="list-style-type: none"> • Figuritas Espaciales. • Figuritas Galácticas. • Números Espaciales. • Números Galácticos.
2. Selecciona el modo de juego “Figuritas Espaciales”.	<p>2.1 Muestra la interfaz del juego.</p> <p>2.2 Muestra un panel con 4 figuras para completar el sudoku.</p> <p>2.3 Muestra una matriz de 4x4 que contiene el sudoku a resolver (ver CU Realizar Sudoku).</p> <p>2.4 Muestra los planetas (niveles) por los que transcurre el jugador.</p> <p>2.5 Muestra el planeta en que se encuentra el jugador.</p> <p>2.6 Muestra el contador del tiempo.</p> <p>2.7 Muestra la puntuación del jugador.</p>
3.	3.1 Termina el caso de uso.
Flujos alternos: 2.a Figuritas Galácticas	
1. Selecciona un modo de juego “Figuritas Galácticas”.	<p>1.1 Muestra la interfaz del juego.</p> <p>1.2 Muestra un panel con 9 figuras para completar el sudoku.</p> <p>1.3 Muestra una matriz de 9x9 que contiene el sudoku a resolver (ver CU Realizar Sudoku).</p> <p>1.4 Muestra los planetas (niveles) por los que transcurre el jugador.</p> <p>1.5 Muestra el planeta en que se encuentra el jugador.</p> <p>1.6 Muestra el contador del tiempo.</p> <p>1.7 Muestra la puntuación del jugador.</p>
2.	2.1 Termina el caso de uso.



Flujos alternos: 2.b Números Espaciales	
1. Selecciona un modo de juego "Números Espaciales".	1.1 Muestra la interfaz del juego. 1.2 Muestra un panel con 4 números para completar el sudoku. 1.3 Muestra una matriz de 4x4 que contiene el sudoku a resolver (ver CU Realizar Sudoku). 1.4 Muestra los planetas (niveles) por los que transcurre el jugador. 1.5 Muestra el planeta en que se encuentra el jugador. 1.6 Muestra el contador del tiempo. 1.7 Muestra la puntuación del jugador.
2.	2.1 Termina el caso de uso.
Flujos alternos: 2.c Números Galácticos	
1. Selecciona un modo de juego "Números Galácticos".	1.1 Muestra la interfaz del juego. 1.2 Muestra un panel con 9 números para completar el sudoku. 1.3 Muestra una matriz de 9x9 que contiene el sudoku a resolver (ver CU Realizar Sudoku). 1.4 Muestra los planetas (niveles) por los que transcurre el jugador. 1.5 Muestra el planeta en que se encuentra el jugador. 1.6 Muestra el contador del tiempo. 1.7 Muestra la puntuación del jugador.
2.	2.1 Termina el caso de uso.
Sección 3: "Multijugador": Flujo Normal de Eventos	
Actor	Sistema
1.	1.1 Muestra el jugador en turno. 1.2 Muestra el contador de tiempo. 1.3 Muestra un panel con los números para completar el sudoku. 1.4 Muestra el sudoku a resolver. 1.5 Muestra el tiempo transcurrido.
2. Selecciona un número	2.1 Marca el número seleccionado.
3. Coloca el número seleccionado en una o varias casillas.	3.1 Verifica si el número está correcto o no. 3.2 Actualiza la puntuación.
4.	4.1 Verifica que el sudoku este completo. 4.2 Muestra la puntuación obtenida por el jugador. 4.3 Le concede el turno al próximo jugador. y se comenzará a



		<p>ejecutar nuevamente el flujo básico de eventos a partir del paso 1.1.</p> <p>4.4 Cuando terminen de jugar todos los jugadores se mostrará los resultados siguientes:</p> <ul style="list-style-type: none"> • Nombre de cada jugador. • Puntos obtenidos. • Lugar alcanzado. <p>4.5 Termina el caso de uso.</p>
Flujos alternos: 4.a Tiempo agotado.		
4.		<p>4.1 Notifica que el tiempo se agotó.</p> <p>4.2 Muestra la puntuación obtenida por el jugador.</p> <p>4.3 Le concede el turno al próximo jugador y se comenzará a ejecutar nuevamente el flujo básico de eventos a partir del paso 1.1 de la sección 3.</p> <p>4.4 Cuando terminen de jugar todos los jugadores se mostrará los resultados siguientes:</p> <ul style="list-style-type: none"> • Nombre de cada jugador • Puntos obtenidos. • Lugar alcanzado. <p>4.5 Termina el caso de uso.</p>
Relaciones	CU Incluidos	Realizar sudoku
	CU Extendidos	Cambiar nivel

Juego “Sopa de letras”

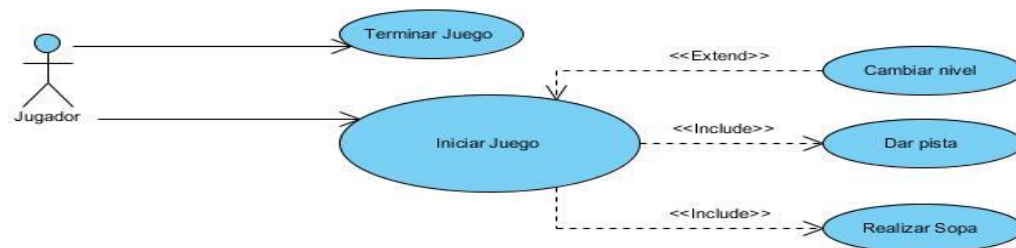


Figura 7. Diagrama de casos de uso del juego Sopa de letras.



Juego “Crucigrama”

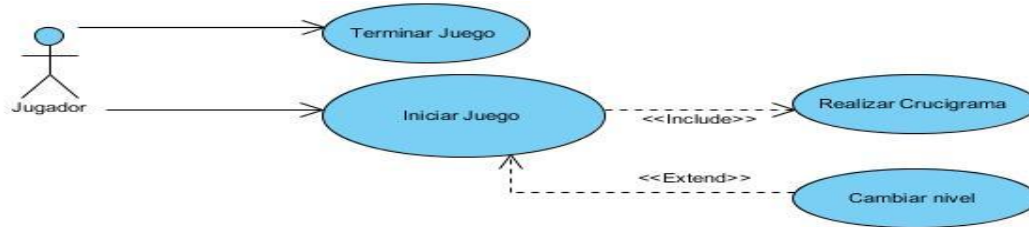


Figura 8. Diagrama de casos de uso del juego Crucigrama.

Juego “Pasatiempo Matemático”

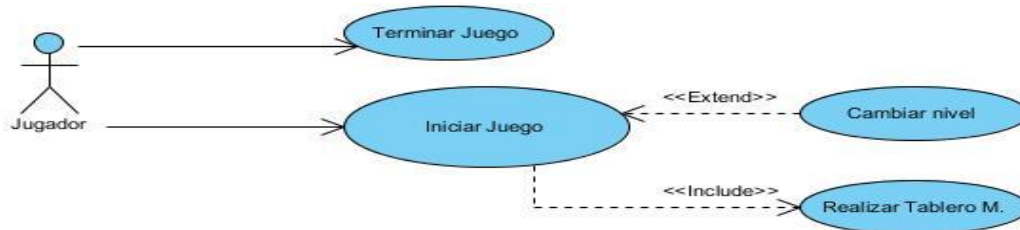


Figura 9. Diagrama de casos de uso del juego Pasatiempo Matemático

2.5 Conclusiones del capítulo

Con la realización de este capítulo se pudo concluir que:

- La descripción de la propuesta del sistema permitió definir los principales conceptos asociados al dominio del problema y ofreció un mejor entendimiento de las soluciones a implementar.
- La identificación de las características y funcionalidades a incluir en el desarrollo de los juegos Sopa de Letras, Crucigrama, Sudoku y Pasatiempo Matemático propició un correcto levantamiento de requisitos, el que permitirá una adecuada elaboración de los artefactos ingenieriles de las diferentes fases del desarrollo.



Capítulo 3. Diseño del sistema

Introducción

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

3.1 Patrón arquitectónico

Los patrones arquitectónicos son patrones de software encargados de definir la estructura de un sistema, estos a su vez se componen de subsistemas con sus responsabilidades. Cuentan con varias directivas para organizar los componentes del propio sistema, con el objetivo de facilitar el diseño del producto (64).

Para la implementación de los juegos se emplea el patrón Modelo-Vista-Controlador.

Modelo Vista Controlador

El patrón MVC (*Model-View-Controller* traducido al español Modelo-Vista-Controlador) es un patrón de arquitectura de software encargado de separar la lógica de negocio de la interfaz del usuario y es el más utilizado en aplicaciones web, ya que facilita la funcionalidad, mantenibilidad y escalabilidad del sistema, de forma simple y sencilla, a la vez que permite "no mezclar lenguajes de programación en el mismo código" (65).

MVC divide las aplicaciones en tres niveles de abstracción (65):

Modelo (objetos de negocio): representa la lógica de negocios. Es el encargado de dar acceso de forma directa a los datos actuando como intermediario.

Vista (interfaz con el usuario u otro sistema): es la encargada de mostrar la información al usuario de forma gráfica y humanamente legible.

Controlador (controlador del flujo de trabajo de la aplicación): es el intermediario entre la vista y el modelo. Es quien controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que ésta, lo presente al usuario, de forma humanamente legible.

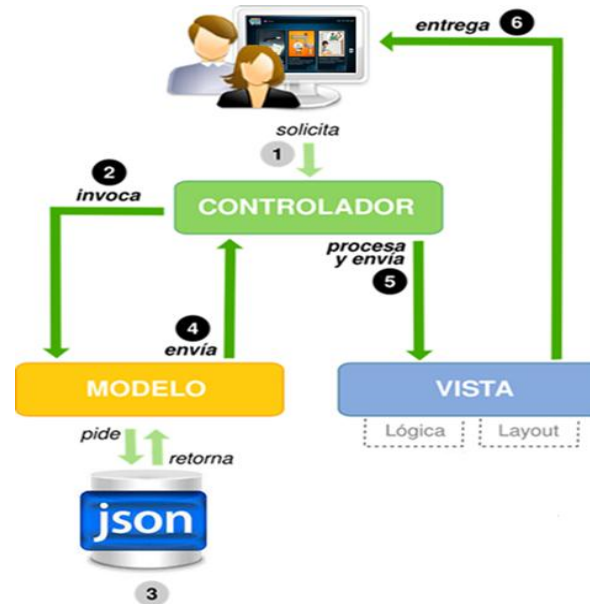


Figura 10. Funcionamiento del patrón modelo-vista-controlador.

El funcionamiento básico del patrón MVC, puede resumirse en (65):

- El usuario realiza una petición.
- El controlador captura el evento.
- Hace la llamada al modelo(s) correspondiente(s).
- El modelo será el encargado de interactuar con la base de datos, ya sea en forma directa, con una capa de abstracción para ello, un *Web Service*, etc. y retornará esta información al controlador.
- El controlador recibe la información y la envía a la vista.
- La vista, una vez procesados los datos, los "acomodará" y los entregará al usuario de forma humanamente legible.

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

3.2 Modelo de diseño

Antes de pasar a las particularidades del diseño en la presente investigación, se hace necesario justificar por qué no se realiza el análisis. Como un elemento determinante se tiene el hecho de que los clientes del producto son personas (estudiantes o profesionales) con conocimientos de informática, por lo que el proceso de captura de requisitos fue claro y explícito. Dentro de la



metodología en uso, el análisis tiene como objetivo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea más fácil de mantener y que ayude a estructurar el sistema entero; de ahí que el equipo de desarrollo decidió ofrecer la visión general del sistema a través del estudio de los resultados del diseño y la implementación (35).

El modelo de diseño es 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. Además el modelo de diseño propicia una de las entradas fundamentales a las actividades de implementación dado que representa una abstracción de la misma (35).

3.2.1 Diagrama de interacción

Los diagramas de interacción son diagramas que describen cómo grupos de objetos colaboran para conseguir algún fin. Estos diagramas muestran objetos, así como los mensajes que se pasan entre ellos dentro del caso de uso. Los diagramas de interacción capturan el comportamiento de los casos de uso (66).

Se expresan de dos formas:

Diagramas de colaboración: Muestra cómo las instancias específicas de las clases trabajan juntas para conseguir un objetivo común. Implementa las asociaciones del diagrama de clases mediante el paso de mensajes de un objeto a otro. Muestra las relaciones sobre los objetos sin mostrar la dimensión temporal de dichas relaciones. La secuencia de los mensajes vendrá dada por números de secuencia (66).

Una colaboración incluye dos tipos de constructores (66):

- Descripción de la estructura estática de los objetos afectados, incluyendo sus relaciones
- Descripción de la secuencia de mensajes intercambiados por los objetos para realizar el trabajo.

Diagramas de secuencia: Muestran las interacciones expresadas en función de secuencias temporales (66).

- Muestra los objetos participantes y los mensajes que intercambian entre ellos a lo largo del tiempo.
- Sin embargo, no muestra los enlaces entre los objetos.
- Son más apropiados para especificar restricciones de interacción en tiempo real.
- Tiene dos dimensiones:



Vertical: Representa el tiempo. La línea vertical representa la existencia de un objeto a lo largo de un determinado tiempo y recibe el nombre de línea de vida del objeto

Horizontal: Representa los distintos objetos. El orden horizontal de aparición de los objetos no tiene ninguna importancia.

- El tiempo avanza desde el comienzo hasta el final de la página.
- Si el objeto es destruido durante el diagrama, se marcaría este evento con una gran "X".
- Una activación muestra el periodo durante el cual un objeto realiza una acción. Se representa como un rectángulo alineado con los momentos en que se inicia y en que finaliza.

A continuación se muestra el diagrama de secuencia del juego Sopa de Letras. Los demás diagramas puede encontrarlos en los anexos:

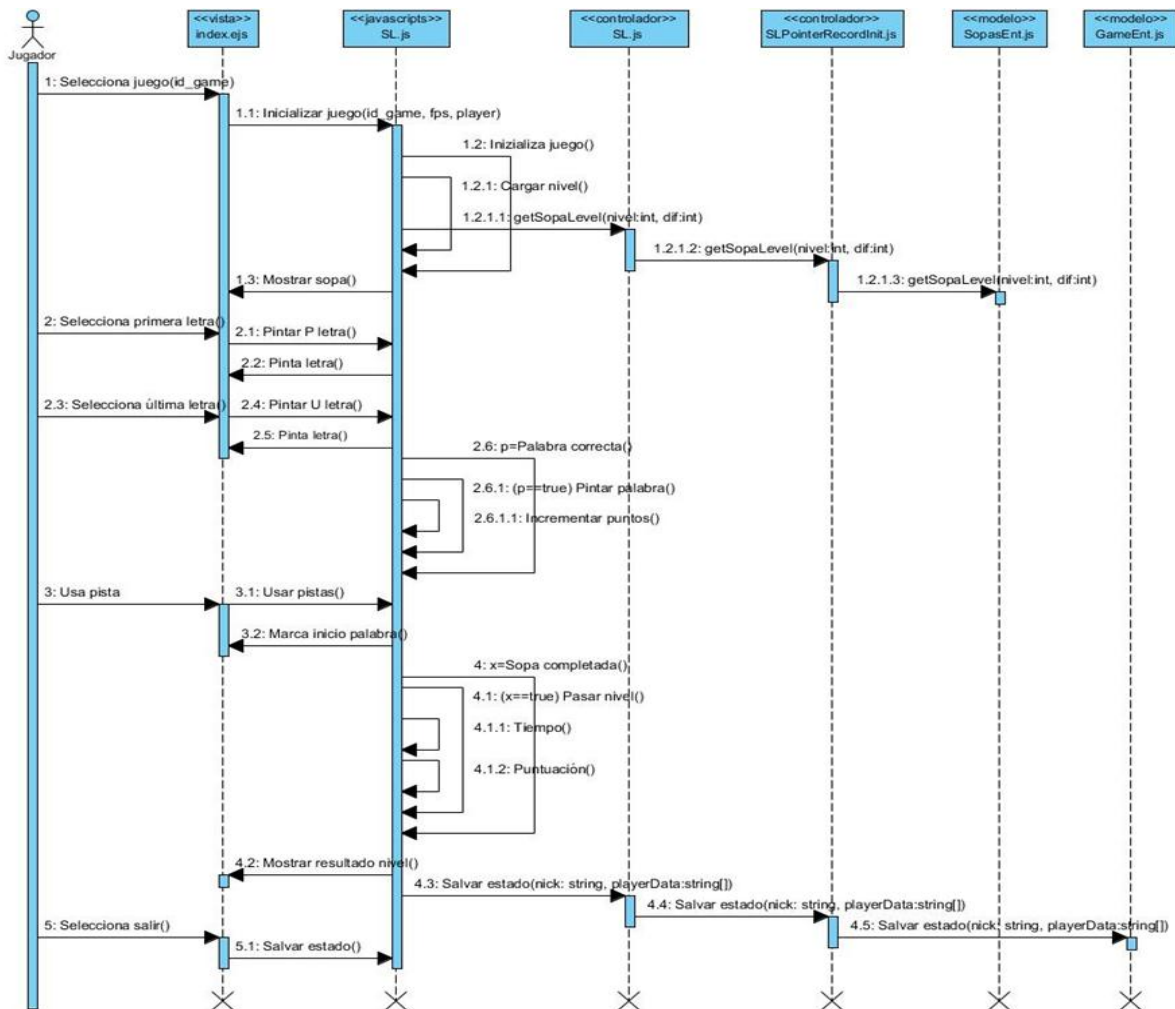


Figura 11. Diagrama de secuencia juego Sopa de letras.



3.2.2 Diagrama de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación (37). Una clase de diseño es una abstracción de una clase o construcción similar en la implementación del sistema. Por ejemplo (35):

- El lenguaje utilizado para especificar una clase del diseño es el mismo que el lenguaje de programación. Consecuentemente, las operaciones, parámetros, atributos, tipos y demás son especificados utilizando la sintaxis del lenguaje de programación elegido.
- La visibilidad de los atributos y las operaciones de una clase del diseño se especifica con frecuencia, por ejemplo: las palabras *public*, *protected* y *private*.
- Las relaciones de aquellas clases del diseño implicadas con otras clases, a menudo tienen un significado directo cuando la clase es implementada. Por ejemplo, la generalización tiene una semántica que se corresponde con el significado de generalización o herencia en el lenguaje de programación.
- Los métodos de una clase del diseño tienen correspondencia directa con el correspondiente método en la implementación de las clases.

A continuación se muestran los diagramas de clases del diseño y la descripción de cada una de las clases que lo componen de los juegos Sopa de letra y Crucigrama, los demás puede encontrarlos en los anexos del presente documento:



Juego “Sopa de letras”

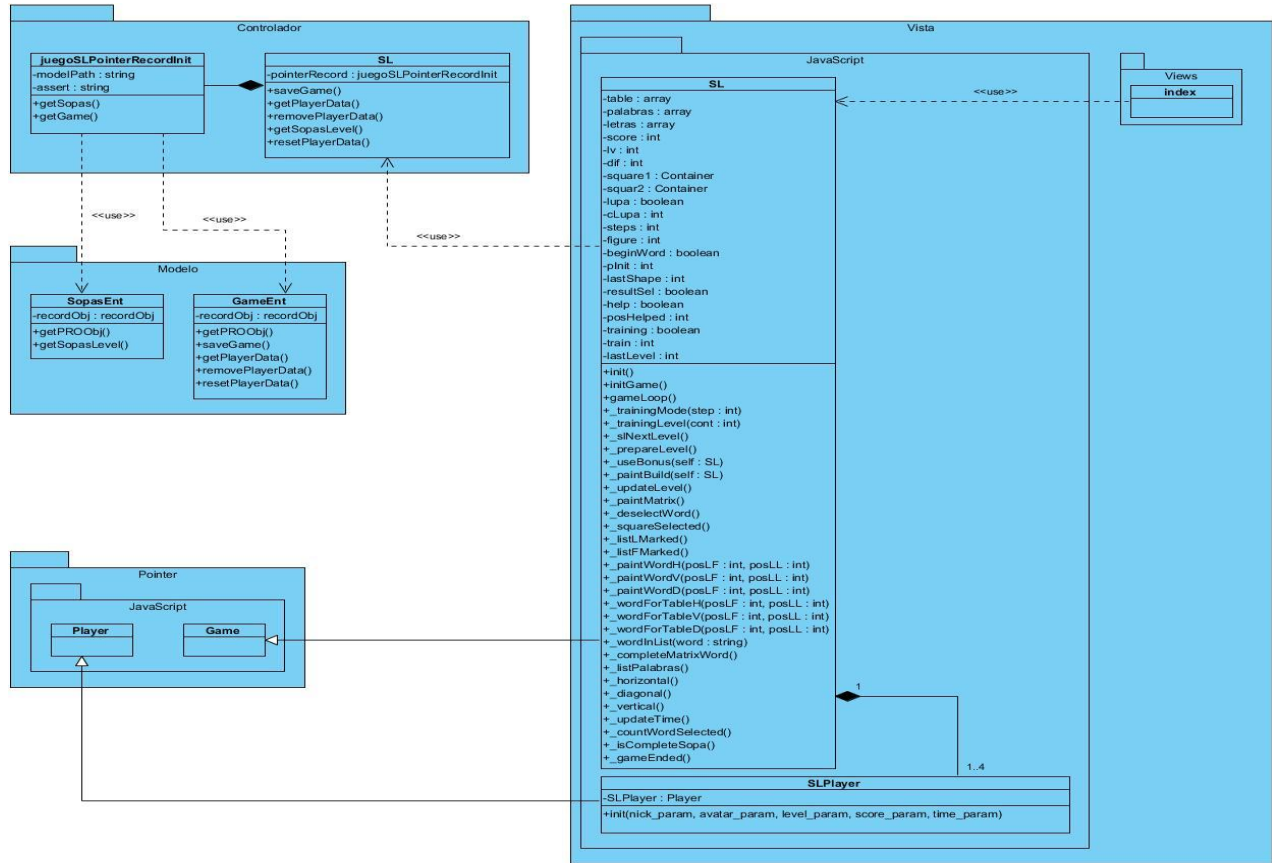


Figura 12. Diagrama de clases del juego Sopa de letras.

Tabla 2. Descripción de las clases del juego Sopa de letras.

Clase	Función
SopasEnt.js	Permite el acceso a los datos guardados en Sopas.json
GameEnt.js	Permite el acceso a los datos guardados en Game.json
juegoSLPointerRecordInit.js	Controla el acceso a los datos mediante las entidades.
SL.js	Controla el acceso a los datos mediante la clase juegoSLPointerRecordInit.js.
SL.js	Controla el flujo del juego.
SLPlayer.js	Gestiona los datos de los jugadores.



Juego “Crucigrama”

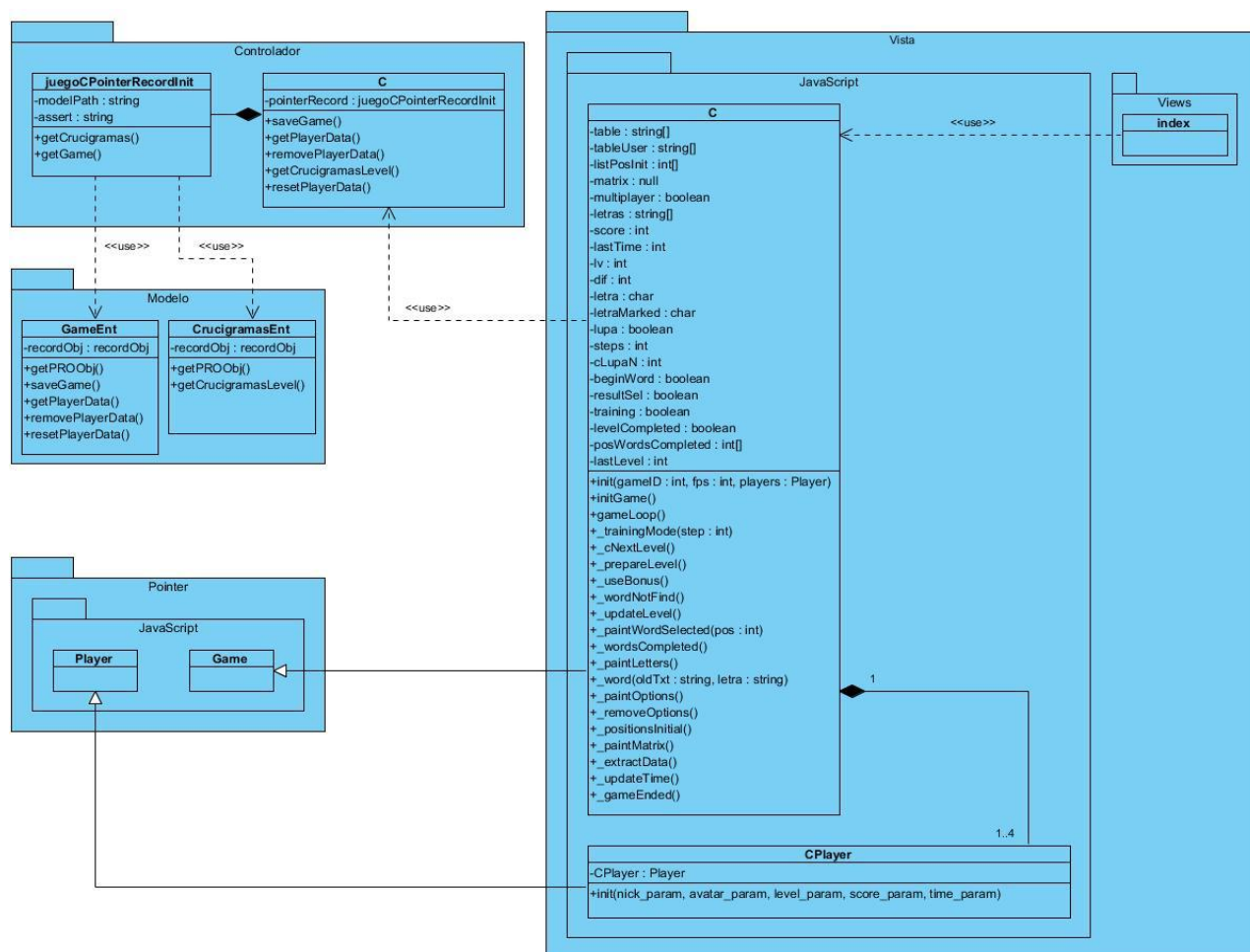


Figura 13. Diagrama de clases del juego Crucigrama.

Tabla 3. Descripción de las clases del juego Crucigrama.

Clase	Función
CrucigramasEnt.js	Permite el acceso a los datos guardados en Crucigramas.json
GameEnt.js	Permite el acceso a los datos guardados en Game.json
juegoCPointerRecordInit.js	Controla el acceso a los datos mediante las entidades.
C.js	Controla el acceso a los datos mediante la clase juegoCPointerRecordInit.js.
C.js	Controla el flujo del juego.
CPlayer.js	Gestiona los datos de los jugadores.



3.2.3 Patrones de diseño

“Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software.” En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Se debe tener presente los siguientes elementos de un patrón: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios) (67).

Patrones GRASP

Los patrones GRASP (*General Responsibility Assignment Software Patterns* traducido al español Patrones generales de software para asignar responsabilidades) describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades (68). La utilización de los patrones GRASP dan origen a sistemas y componentes robustos y fáciles de mantener, entender, reutilizar o extender (69).

Dentro de los patrones GRASP Se destacan 5 patrones principales que son (37):

Tabla 4. Patrones GRASP (68).

Nombre del patrón	Problema	Solución
Experto	¿Cuál es un principio general para asignar responsabilidades a los objetos?	Asignar una responsabilidad específica solo a la clase que cuenta con la información necesaria para cumplir dicha responsabilidad (37).
Creador	¿Quién debería ser el responsable de la creación de una nueva instancia de alguna clase?	Permite identificar quién debe ser el responsable de la instanciación de nuevos objetos (37). Asignar a la clase B la responsabilidad de crear una instancia de clase A si se cumple uno o más de los casos siguientes: <ol style="list-style-type: none">1. B agrega objetos de A.2. B contiene objetos de A.3. B registra instancias de objetos de A.4. B utiliza más estrechamente objetos de A.5. B tiene datos de inicialización que se pasarán a un objeto de A cuando sea creado (por tanto, B es un Experto con respecto a la creación de A).6. B es un creador de los objetos A.



<p>Bajo Acoplamiento</p>	<p>¿Cómo soportar bajas dependencias, bajo impacto del cambio e incremento de la reutilización?</p>	<p>Asignar una responsabilidad de manera que el acoplamiento permanezca bajo. Promueve la implementación de clases lo menos interdependiente posible para que en caso de cambio en alguna de ellas repercuta lo menos posible en el resto (37).</p>
<p>Alta cohesión</p>	<p>¿Cómo mantener la complejidad manejable?</p>	<p>Asignar una responsabilidad de manera que la cohesión permanezca alta. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Una baja cohesión trae como desventaja la dificultad de comprensión, de reutilización y que son constantemente afectadas por los cambios (37).</p>
<p>Controlador</p>	<p>¿Quién debería ser el responsable de gestionar un evento de entrada al sistema?</p>	<p>Objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema (37). Asignar una responsabilidad de recibir o manejar un mensaje de evento del sistema a una clase que representa una de las opciones siguientes:</p> <ul style="list-style-type: none"> • Representa un caso de uso en el que tiene lugar el evento del sistema. • Utilice la misma clase controlador para todos los eventos del sistema en el mismo escenario de caso de uso.

El patrón experto se evidencia en diferentes clases sobre todo en las controladoras como SL, C, SU y PM dado que estas poseen la mayor información necesaria para resolver las funcionalidades requeridas, ellas utilizan su propia información para llevar a cabo sus tareas. También se hace uso del patrón controlador en el diseño de la solución propuesta, esto se evidencia en el manejo de los eventos asociados a las distintas funciones generales del juego, en este caso las clases SL, C, SU y PM son las encargadas de manejar los eventos que permiten inicializar los juegos, almacenar los datos una vez que un jugador ha concluido el juego, controlan el flujo de transición de un jugador a otro cuando se juega en modo multijugador. Los demás patrones son utilizados por la plataforma Pointer.

3.3 Estándares de codificación

Los estilos de código o estándares de codificación son un conjunto de reglas o normas usadas para escribir código y que incluye una gran cantidad de aspectos dentro del proceso de



codificación. Es un estilo de programación homogéneo que permite que los desarrolladores del sistema puedan interpretar de manera eficiente la escritura del código; asegurando que todos trabajen de forma coordinada y en un vocabulario común (70).

Las principales ventajas de utilizar un estándar para escribir el código de las aplicaciones son (71):

- 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.

A continuación se establecen algunas normas de codificación que serán utilizadas en el desarrollo de los juegos Sopa de Letras, Crucigrama, Sudoku y Pasatiempo Matemático:

CamelCase

Es un estilo de escritura que se aplica a frases o palabras compuestas, consistente en utilizar las mayúsculas como separadores de palabras. El nombre se debe a que las mayúsculas a lo largo de una palabra en CamelCase se asemejan a las jorobas de un camello. Si comienza con mayúscula, se denomina *UpperCamelCase* y, si no, *lowerCamelCase* (72).

Nombres de las clases

Para los nombres de las clases se utilizará *UpperCamelCase*, o sea la primera letra de cada una de las palabras es mayúscula. Ejemplo:

```
var PMPlayer;
PMPlayer = ring.create([ Player ], {
  init: function (nick_param, avatar_param, level_param, score_param, time_param, data_difficulty_param) {...}
});
```

Figura 14. Ejemplo de estilo de código UpperCamelCase.

Nombres de los métodos

Para el nombre de los métodos se utilizará *lowCamelCase*. La primera letra de cada una de las palabras es mayúscula a excepción de la primera. Ejemplo:



```
gameLoop: function() {...},
_trainingMode: function(step) {...},
_cNextLevel: function() {
  var self = this;
  self._prepareLevel();
  self.stage.update();
  self.players[self.indexOfActivePlayer].score += self.score;
  self.background.graphics.beginBitmapFill(self.getResource("img_screen" + self.players[self.indexOfActivePlayer].level)).drawRect
  self.players[self.indexOfActivePlayer].chrono.restart();
  if (self.dif == 0) {
    $(".tool").addClass("c_hide");
  }
  self._updateLevel();
  self.startGame();
},
```

Figura 15. Ejemplo de estilo de código lowCamelCase.

3.4 Conclusiones del capítulo

Con la realización de este capítulo se pudo concluir que:

- El modelado de los diagramas de clases del diseño y los diagramas de secuencias proporcionó una mejor comprensión del sistema a la hora de ser implementado.
- El uso de los patrones arquitectónicos Modelo Vista Controlador y de diseño GRASP y *CamelCase* como estilos de codificación evidenciaron la presencia de buenas prácticas de programación en el desarrollo de los juegos Sopa de Letras, Crucigrama, Sudoku y Pasatiempo Matemático.



Capítulo 4. Implementación y prueba

Introducción

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

4.1 Modelo de implementación

El Modelo de Implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se pueden encontrar datos, archivos, ejecutables, código fuente y los directorios. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos. Un diagrama de implementación muestra, entre otros elementos, las dependencias entre las partes de código del sistema (diagramas de componentes) (73).

4.1.1 Diagrama de componentes

Un componente es una parte física de un sistema (pueden ser simples archivos, paquetes, modulo, base de datos, programa ejecutable, etc.). Se puede decir que un componente es la materialización de una o más clases, porque una abstracción con atributos y métodos pueden ser implementados en los componentes (73).

En los diagramas de componentes se muestran los elementos de diseño de un sistema de software. Permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces. El diagrama de componentes (74):

- Se encarga de relacionar los componentes de software en un sistema.
- Describe los elementos físicos del sistema y sus relaciones.
- Muestran las opciones de realización incluyendo código fuente, binario y ejecutable.

A continuación se muestran el diagrama de componentes del juego Sopa de letras, los demás diagramas puede encontrarlos en los anexos:

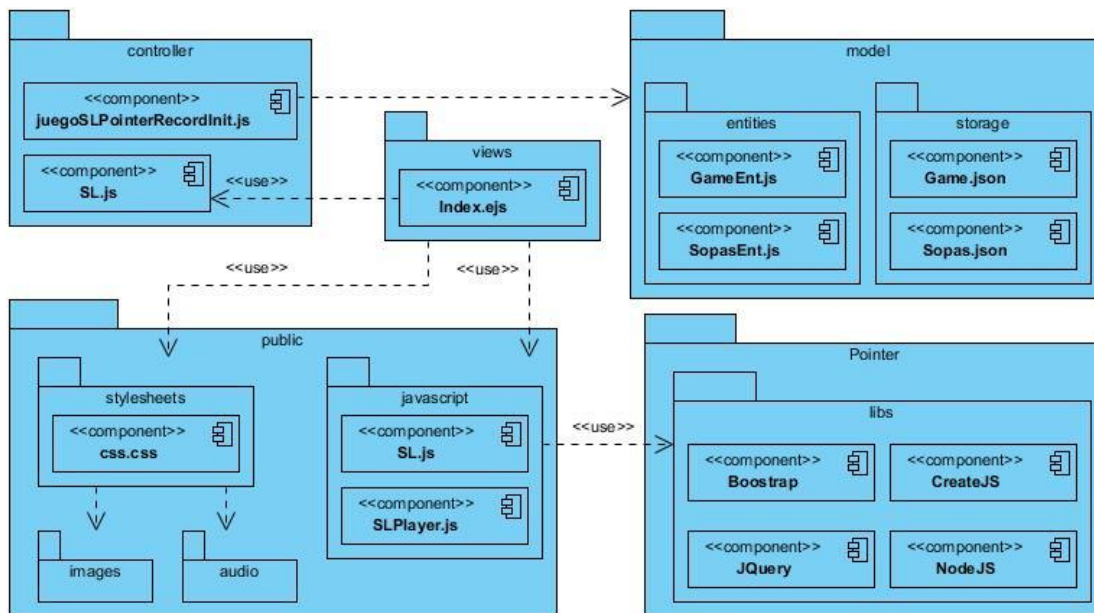


Figura 16. Diagramas de componentes del juego Sopa de letras.

4.2 Pruebas de software

En el proceso de desarrollo de software se define una etapa de pruebas, con el objetivo de analizar si el producto final fue implementado correctamente. Las pruebas demuestran hasta qué punto las funciones del software se corresponden con las especificaciones (75).

Entre sus objetivos están (76):

- 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.

Con el objetivo de garantizar la calidad de los productos desarrollados se hizo necesario verificar su funcionamiento. En el proceso de desarrollo de software se define una etapa de pruebas con los siguientes objetivos (35):

- Planificar las pruebas necesarias en cada iteración (pruebas de integración y pruebas de sistema).
- Diseñar e implementar las pruebas creando los casos de pruebas que especifican qué probar, creando los procedimientos de pruebas que especifican cómo realizar las pruebas y creando componentes de prueba ejecutables para automatizar las pruebas.



- Realizar las diferentes pruebas y manejar los resultados de cada prueba sistemáticamente. Las construcciones en las que se detectan defectos son probadas de nuevo y posiblemente devueltas a otro flujo de trabajo, como diseño o implementación, de forma que los defectos importantes puedan ser arreglados.

Para la elaboración de una adecuada estrategia de prueba es necesario realizar un estudio a los distintos conceptos y representaciones de los mismos en esta área de la ingeniería de software. A continuación se muestran los resultados del análisis realizado por el equipo de desarrollo con las características específicas de la estrategia trazada.

4.2.1 Niveles de prueba

La prueba es aplicada para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. A continuación se describen los niveles de pruebas abarcados (77):

Tabla 5. Prueba de integración.

Indicador	Descripción
Objetivo de la Prueba	Identificar errores introducidos por la combinación de programas probados unitariamente. Verificar que las interfaces entre las entidades externas (usuarios) y las aplicaciones funcionan correctamente. Verificar que las especificaciones de diseño sean alcanzadas. Determina el enfoque para avanzar desde un nivel de integración de los componentes al siguiente.
Descripción de la Prueba	Describe cómo verificar que las interfaces entre las componentes de software funcionan correctamente. Determina el enfoque para avanzar desde un nivel de integración de las componentes al siguiente. Decide qué acciones tomar cuando se descubren problemas.
Técnica	Por cada Caso de Prueba ejecutado: <ul style="list-style-type: none">• Comparar el resultado esperado con el resultado obtenido.• Utilizar la técnica <i>top-down</i>. Se empieza con los módulos de nivel superior, y se verifica que los módulos de nivel superior llaman a los de nivel inferior de manera correcta, con los parámetros correctos.• Utilizar la técnica <i>down-top</i>. Se empieza con los módulos de nivel inferior, y se verifica que los módulos de nivel inferior llaman a los de nivel superior de manera correcta, con los parámetros correctos.



Tabla 6. Prueba de Sistema.

Indicador	Descripción
Objetivo de la Prueba	Asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación.
Descripción de la Prueba	<p>Las pruebas del sistema deben enfocarse en requisitos que puedan ser tomados directamente de casos de uso y reglas y funciones de negocios. El objetivo de estas pruebas es verificar el ingreso, procesamiento y recuperación apropiado de datos, y la implementación apropiada de las reglas de negocios. Este tipo de pruebas se basan en técnicas de caja negra, esto es, verificar el sistema (y sus procesos internos) y analizar las salidas o resultados.</p> <p>En esta prueba se determina qué pruebas de Sistema (funcionalidad, seguridad, esfuerzo, recuperación, etc.) asegurarán que la aplicación alcanzará sus objetivos de negocio.</p>
Técnica	<p>Ejecute cada caso de uso, flujo básico o función utilizando datos válidos e inválidos, para verificar que:</p> <ul style="list-style-type: none">• Los resultados esperados ocurren cuando se utiliza un dato válido.• Los mensajes de error o de advertencia aparecen en el momento adecuado, cuando se utiliza un dato inválido.• Cada regla de negocios es aplicada adecuadamente.

4.2.2 Tipos de pruebas

Dentro de cada nivel de prueba se engloba una técnica de prueba específica según los atributos de calidad que se deseen verificar con las pruebas al software. De acuerdo con el nivel se utilizan varios tipos pruebas, los cuales se exponen a continuación (77):

Tabla 7. Pruebas de funcionalidad.

Indicador	Descripción
Objetivo de la Prueba	Se asegura el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados
Descripción de la Prueba	<p>Las pruebas funcionales deben enfocarse en los requisitos funcionales, las pruebas pueden estar basadas directamente en los Casos de Uso (o funciones de negocio), y las reglas del negocio. Las metas de estas pruebas son:</p> <ul style="list-style-type: none">• Verificar la apropiada aceptación de datos.• Verificar el procesamiento y recuperación y la implementación adecuada de las reglas del negocio. <p>Este tipo de pruebas están basadas en técnicas de caja negra, que consiste en verificar</p>



	la aplicación (y sus procesos internos) y analizar la salida (resultados). Lo que se identifica a continuación es un diseño preliminar de las pruebas recomendadas para cada aplicación.
Técnica	Se ejecuta cada caso de uso, flujo de caso de uso, o función, usando datos válidos e inválidos, para verificar lo siguiente: <ul style="list-style-type: none">• Que los resultados esperados ocurran cuando se usen datos válidos.• Que sean desplegados los mensajes apropiados de error y precaución cuando se usan datos inválidos.• Que se aplique apropiadamente cada regla de negocio.

Tabla 8. Pruebas de regresión.

Indicador	Descripción
Objetivo de la Prueba	Determinar si los cambios recientes en una parte de la aplicación tienen efecto adverso en otras partes.
Descripción de la Prueba	En esta prueba se vuelve a probar el sistema a la luz de los cambios realizados durante la corrección, mantenimiento o desarrollo de la nueva versión del sistema buscando efectos adversos en otras partes.
Técnica	La prueba de regresión es una nueva corrida de casos de prueba previos. Aquellos casos de uso (y los casos de prueba asociados) que descubren defectos tempranamente deben ser incluidos en la prueba de regresión.

4.2.3 Métodos de prueba

Como método de prueba para probar los requerimientos, las funciones y las respuestas de los sistemas se decidió utilizar las pruebas de caja negra.

Las pruebas de **caja negra** o también llamadas pruebas de comportamiento se centran en los requisitos funcionales del software, estas tratan de encontrar errores en las siguientes categorías: funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y de terminación (75).

La **partición equivalente** es una técnica de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos a partir de las cuales pueden derivarse casos de prueba. La partición equivalente se esfuerza por definir un caso de prueba que descubra ciertas clases de errores, reduciendo así el número total de casos de prueba que deben desarrollarse (75).



4.2.4 Resultados de las pruebas

La estrategia de prueba trazada consistió en abarcar los niveles integración y sistema. Las pruebas a nivel de integración fueron utilizadas para comprobar el intercambio de datos entre componentes del modelo de implementación y que estos operen correctamente cuando son combinados para ejecutar un caso de uso, mientras las pruebas del nivel de sistema se realizaron para verificar el funcionamiento del sistema como un todo. Las pruebas de funcionalidad fueron realizadas en ambos niveles para verificar el cumplimiento de los requisitos funcionales mientras que las pruebas de regresión fueron realizadas solamente en el nivel de sistema con el propósito de asegurar que cuando una no conformidad sea corregida no tenga efectos adversos en otras funcionalidades de los juegos.

A continuación se muestra un resumen de los aspectos esenciales de la estrategia de prueba que guió el proceso de validación de las aplicaciones:

Tabla 9. Estrategia de prueba.

Nivel de prueba	Tipo de prueba	Método de prueba	Técnica
Integración	Funcionalidad	Caja negra	Partición equivalente
Sistema	Funcionalidad Regresión	Caja negra	Partición equivalente

El seguimiento de esta estrategia permitió reconocer no conformidades de los juegos durante las tres iteraciones de prueba que fueron definidas por el equipo de desarrollo. La siguiente tabla refleja las cantidades de no conformidades detectadas durante las pruebas.

Tabla 10. Resultado de las pruebas. No conformidades según iteraciones de prueba.

Juego	Cantidad de no conformidades		
	Iteración I	Iteración II	Iteración III
Crucigrama	5	2	0
Pasatiempo Matemático	4	1	0
Sopa de letras	6	3	0
Sudoku	4	1	0

Las no conformidades detectadas durante la primera iteración fueron en su mayoría de validación. En la segunda iteración las pruebas estuvieron enfocadas en la integración de los componentes del modelo de implementación. En la tercera iteración las pruebas se realizaron encaminadas al manejo de los datos relacionados con cargar y mostrar juegos guardados



dentro de la plataforma y en la comprobación de la solución de las no conformidades anteriormente detectadas.

Algunas de las no conformidades detectadas se describen a continuación:

Juego “Sopa de letras”

- Cuando se marcaba la primera letra de una palabra al utilizar la pista y se encontraba la palabra indicada la letra continuaba resaltada en la sopa.
- En algunos casos no permitía que las palabras fueran marcadas seleccionando la última letra y posteriormente la primera.
- Cuando la puntuación del jugador excedía los 1000 puntos esta se mostraba fuera del contador de puntuación.

Juego “Crucigrama”

- Cuando se utilizaba una pista, se colocaba una letra aleatoria en el crucigrama, pero no se validaba que la letra ya estuviera ubicada o no.
- Las descripciones de las palabras en ocasiones sobresalían del marco que las contenía.
- Cuando se completaba una palabra correctamente y se cambiaba una letra de esta se mantenía la palabra señalada como correcta.

Juego “Sudoku”

- Algunos números pintados de rojo por haber sido colocados en una posición incorrecta no se marcaban y pintaban de verde una vez que el conflicto que los declaraba incorrectos era solucionado.
- Cuando se cambiaba de nivel no se cargaba la imagen de fondo correspondiente al nivel.
- Cuando el jugador llegaba al segundo nivel se salía del juego para la plataforma automáticamente.

Juego “Pasatiempo Matemático”

- Cuando se terminaba el tiempo (10 minutos) para resolver el pasatiempo matemático se detenía el juego y no mostraba el mensaje de “Tiempo Agotado”.
- Los mensajes en el nivel de tutorial no se encontraban en un orden correcto.
- En la interfaz del juego salía una barra de desplazamiento en la parte derecha.

4.3 Conclusiones del capítulo

Con la realización de este capítulo se pudo concluir que:



- La obtención del modelo de implementación posibilitó representar la realización física de los elementos del diseño.
- El estudio realizado sobre el área de las pruebas de software permitió la elaboración de una estrategia para la validación de las soluciones obtenidas, demostrando el cumplimiento de las especificaciones de cada una de ellas.



Conclusiones generales

El presente documento es resultado de la investigación iniciada con el objetivo de desarrollar los juegos educativos Sopa de Letras, Crucigrama, Sudoku y Pasatiempo Matemático para la colección de juegos educativos MundoClick. Al término de la implementación de estas soluciones, el equipo de desarrollo definió las siguientes conclusiones:

- El estudio detallado de aplicaciones similares a las que se desarrollan en la investigación demostró la necesidad de implementar los juegos Sopa de Letras, Crucigrama, Sudoku y Pasatiempo Matemático para la colección de juegos educativos MundoClick.
- La obtención de los artefactos ingenieriles, definidos en cada una de los flujos de trabajo de la metodología RUP, permitió la ejecución de un proceso de desarrollo bien documentado en cada una de sus fases.
- Como principal resultado de la investigación se obtuvo la implementación de los juegos Sopa de Letras, Crucigrama, Sudoku y Pasatiempo Matemático integrados en su totalidad a la colección de juegos educativos MundoClick, los cuales contribuyen a desarrollar el razonamiento lógico, matemático y verbal en los niños.
- La estrategia de pruebas diseñada para los juegos permitió mejorar aspectos relacionados con el diseño y el funcionamiento de los mismos, logrando que el resultado final esté acorde con los guiones que describen cada uno de ellos.



Recomendaciones

Se recomienda al proyecto Multisaber-Navegante:

- Que en conjunto con el MINED incorpore el contenido pedagógico a cada juego y añadirle el diseño gráfico correspondiente a los mismos.
- Desarrollar una aplicación que facilite la incorporación de los contenidos correspondientes a cada uno de los juegos en los ficheros json.



Referencias Bibliográficas

1. **Ecured.** Software educativos en Cuba. [En línea] [Citado el: 6 de 11 de 2013.]
http://www.ecured.cu/index.php/Software_educativos_en_Cuba.
2. **Eumed.net-Enciclopedia Virtual.** Enciclopedia y Biblioteca Virtual de las Ciencias Sociales, Económicas y Jurídicas. *EL SOFTWARE EDUCATIVO UN MEDIO DE ENSEÑANZA EFICIENTE.* [En línea] [Citado el: 11 de 06 de 2013.] <http://www.eumed.net/rev/ced/29/sml.htm>.
3. **GESPRO.** Suite de Gestión de Proyectos. [En línea] [Citado el: 06 de 11 de 2013.]
<http://gespro.fortes.prod.uci.cu/>.
4. **Educapeques.** Portal de educación infantil y primaria. *Aprender más fácil: La importancia de los juegos educativos .* [En línea] [Citado el: 6 de 11 de 2013.]
<http://www.educapeques.com/escuela-de-padres/consejos-educacion-hijos/importancia-juegos-educativos.html>.
5. **Hernández León, Rolando Alfredo y Coello González, Sayda.** *EL paradigma cuantitativo de la investigación científica.* La Habana : s.n., 2002.
6. **Martín-Laborda, Rocío.** Fundación AUNA. *Las nuevas tecnologías en la educación.* [En línea] [Citado el: 22 de 1 de 2014.]
http://biblioteca.ulsu.edu.mx/publicaciones/nuevas_tecnologias.pdf.
7. **L Ramos Pérez, J Domínguez Lovaina, X Gavilondo Mariño, C Fresno Chávez.** Centro Nacional de Información de Ciencias Médicas-Infomed. [En línea] [Citado el: 22 de 1 de 2014.]
http://bvs.sld.cu/revistas/aci/vol18_4_08/aci61008.htm.
8. **Marqués, Pere.** Universidad Autónoma de Barcelona. *El software educativo.* [En línea] [Citado el: 22 de 1 de 2014.]
http://recursos.salonesvirtuales.com/assets/bloques/educativo_de_pere_MARQUES.pdf.
9. **Núñez Rojas, Nemecio.** eumed.net. *BIBLIOTECA VIRTUAL de Derecho, Economía y Ciencias Sociales.* [En línea] [Citado el: 22 de 1 de 2014.] <http://www.eumed.net/libros-gratis/2010a/669/Concepcion%20del%20Software%20educativo.htm>.
10. **Rosa M. Carro, Ana M. Breda, Gladys Castillo, Antonio L. Bajuelos.** Departamento de Matemática, Universidad de Aveiro. [En línea] [Citado el: 23 de 1 de 2014.]
www.researchgate.net/publication/228853579_Generacin_de_juegos_educativos_adaptativos/file/d912f50c2ba31de199.pdf.
11. **Gros, Begoña.** Universidad Pedagógica Veracruzana. *DEL SOFTWARE EDUCATIVO A EDUCAR CON SOFTWARE.* [En línea] [Citado el: 22 de 1 de 2014.]



<http://upvv.clavijero.edu.mx/cursos/EstrategiasAprendizajeCienciasSociales/programa/documentos/Delsoftwareeducativoaeducarconsoftware.pdf>.

12. **Huertas, Catalina Ponce.** CSI-F Central Sindical Independiente y de Funcionarios. *CSI-F Central Sindical Independiente y de Funcionarios*. [En línea] 2009. [Citado el: 23 de 1 de 2014.] [\[csif.es/andalucia/modules/mod_ense/revista/pdf/Numero_19/CATALINA_PONCE_HUERTAS02.pdf\]\(http://www.csi-csif.es/andalucia/modules/mod_ense/revista/pdf/Numero_19/CATALINA_PONCE_HUERTAS02.pdf\).](http://www.csi-</p></div><div data-bbox=)

13. **MÁRQUEZ RODRÍGUEZ, ALEIDA.** *Habilidades: reflexiones y proposiciones para su evaluación*. 1993.

14. **Herreros Martínez, Oscar.** CONSULTA PSICOPEDAGÓGICA PARA PADRES. *QUÉ SON LAS HABILIDADES COGNITIVAS*. [En línea] [Citado el: 10 de 05 de 2014.]

<http://consultaparapadres.blogspot.com/2006/11/qu-son-las-habilidades-cognitivas.html>.

15. **EROSKI CONSUMER.** Desarrollar el razonamiento verbal. [En línea] [Citado el: 10 de 05 de 2014.] <http://www.consumer.es/web/es/educacion/extraescolar/2010/02/21/191265.php>.

16. **Elementos.** Asociación de Estudio e Innovación. *¿Que son los juegos lógico matemáticos?* [En línea] [Citado el: 10 de 05 de 2014.] http://www.elementos.pe/juegos_lm.php.

17. **Definición.de.** Definición de razonamiento lógico. [En línea] [Citado el: 10 de 05 de 2014.] <http://definicion.de/razonamiento-logico/>.

18. **EducaKids.** Los beneficios de aprender a jugar al Sudoku. [En línea] [Citado el: 10 de 05 de 2014.] http://www.educakids.com/educa/articulo_expandido.php?id=221.

19. **Educación en Competencias.** Centro del Profesorado de Córdoba. *Competencia en razonamiento matemático*. [En línea] [Citado el: 10 de 05 de 2014.] <http://competenciasbasicascordoba.webnode.es/razonamiento-matematico/>.

20. **GCompris.** GCompris. [En línea] [Citado el: 9 de 2 de 2014.] <http://gcompris.net/index-es.html>.

21. **Epasatiempos.es.** Pasatiempos electrónicos y juegos gratuitos online que ejercitan la mente. [En línea] [Citado el: 8 de 2 de 2014.] <http://www.epasatiempos.es>.

22. —. Pasatiempos electrónicos y juegos gratuitos online que ejercitan la mente. *Sopas de letras temáticas*. [En línea] [Citado el: 8 de 2 de 2014.] <http://www.epasatiempos.es/sopas-de-letras-tematicas.php>.

23. **Programas-gratis.net.** Sopa de letras. [En línea] [Citado el: 9 de 2 de 2014.] <http://sopa-de-letras.programas-gratis.net/>.



24. **Epasatiempos.es.** Pasatiempos electrónicos y juegos gratuitos online que ejercitan la mente. *Crucigramas*. [En línea] [Citado el: 8 de 2 de 2014.]
<http://www.epasatiempos.es/crucigramas.php>.
25. **Programas-gratis.net.** Crossword express. [En línea] [Citado el: 8 de 2 de 2014.]
<http://crossword-express.programas-gratis.net/>.
26. **Epasatiempos.es.** Pasatiempos electrónicos y juegos gratuitos online que ejercitan la mente. *Tableros matemáticos*. [En línea] [Citado el: 8 de 2 de 2014.]
<http://www.epasatiempos.es/tableros-matematicos.php>.
27. —. Pasatiempos electrónicos y juegos gratuitos online que ejercitan la mente. *Sudokus*. [En línea] [Citado el: 8 de 2 de 2014.] <http://www.epasatiempos.es/sudokus.php>.
28. **Domo-Sudoku.** Sudoku. [En línea] [Citado el: 8 de 2 de 2014.] domo-sudoku.com.
29. **Programas-gratis.net.** Pure sudoku. [En línea] [Citado el: 8 de 2 de 2014.] <http://pure-sudoku.programas-gratis.net/>.
30. —. Sudoku deluxe. [En línea] [Citado el: 8 de 2 de 2014.] <http://sudoku-deluxe.programas-gratis.net/>.
31. —. Kakuro. [En línea] [Citado el: 8 de 2 de 2014.] <http://kakuro.programas-gratis.net/>.
32. **Alarcos.** Universidad de Castilla-La Mancha. *Metodologías de Desarrollo de Software*. [En línea] [Citado el: 23 de 1 de 2014.] <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>.
33. **C&TA.** Ciencia y Técnica Administrativa. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. [En línea] [Citado el: 23 de 1 de 2014.]
http://www.cyta.com.ar/ta0502/b_v5n2a1.htm.
34. **Rueda Chacón, Julio César.** Aplicación de la metodología RUP para el desarrollo rápido de aplicaciones basado en el estándar j2ee. [En línea] 2006. [Citado el: 1 de 2 de 2014.]
http://clasescun.pbworks.com/f/Metodolog%C3%ADa_RUP.pdf.
35. **Jacobson, Ivan, Booch, Grady y Rumbaugh, James .** *El proceso unificado de desarrollo de software*. Madrid : Addison Wesley, 2000. 84-7829-036-2.
36. **BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar.** *El lenguaje unificado de modelado*. s.l. : Addison Wesley, 2000. 84-7829-037-0.
37. **LARMAN, Craig.** *UML y patrones*. s.l. : Pearson, 1999.
38. **Gómez, Ruth Priscila Laderos y del Juncal Huerta, Jorge Luis.** *Herramientas CASE*. 2007.
39. **López Pecho, Ramiro y Cesar Ballesteros, Julio.** *Herramientas CASE. Herramientas CASE*. [En línea] 29 de 7 de 2008. [Citado el: 23 de 1 de 2014.] <http://tpsis324.blogspot.com>.



40. **Sierra, María.** *Trabajando con Visual Paradigm for UML.* Cantabria : Universidad de Cantabria – Facultad de Ciencias : s.n.
41. **Definicion.org.** Definición de lenguaje de programación. [En línea] [Citado el: 1 de 2 de 2014.] <http://www.definicion.org/lenguaje-de-programacion>.
42. **Pérez, Javier Eguíluz.** *Introducción a CSS.* 2008.
43. **Alvarez, Miguel Angel.** *Manual de CSS3.* 2007. s.n.
44. **Pérez, Javier Eguíluz .** *Introducción a JavaScript.*
45. **Brandendaugh, Jerry .** *Aplicaciones JavaScript.* 84-415-1070-9.
46. **Vega, John Freddy y Van Der Henst, Christian .** *El presente de la Web HTML5.* 2011.
47. **W3Schools.** HTML5 Introduction. [En línea] [Citado el: 1 de 2 de 2014.] http://www.w3schools.com/html/html5_intro.asp.
48. —. JSON Tutorial. [En línea] [Citado el: 1 de 2 de 2014.] <http://www.w3schools.com/json/default.asp>.
49. **JSON.org.** Introducción a JSON. [En línea] [Citado el: 1 de 2 de 2014.] <http://www.json.org/json-es.html>.
50. **Ecured.** Framework. [En línea] [Citado el: 1 de 2 de 2014.] <http://www.ecured.cu/index.php/Framework>.
51. **Saavedra, Estaban .** Frameworks de Desarrollo Web. *Frameworks de Desarrollo Web.* [En línea] [Citado el: 2 de 2 de 2014.] <http://www.slideshare.net/estebansaavedra/frameworks-de-desarrollowebgrails>.
52. **ALEGSA©.** Definicion de Biblioteca (informática) - ¿qué es Biblioteca? [En línea] [Citado el: 07 de 06 de 2014.] <http://www.alegsa.com.ar/Dic/biblioteca.php>.
53. **Hernández Nápoles, Yoandys Michel.** *Juegos Buscando parejas y Buscando herramientas para la colección educativa MundoClick.* Ciudad de la Habana : s.n., 2013.
54. **Tutorial-Bootstrap.** Framework, Twitter Bootstrap! [En línea] [Citado el: 1 de 2 de 2014.] <http://internoma.github.io/tutorial-bootstrap/>.
55. **Overprint.** Bootstrap, un framework html/css de Twitter para desarrolladores. [En línea] [Citado el: 2 de 2 de 2014.] <http://overprint.com.ar/desarrollo/bootstrap-un-framework-htmlcss-de-twitter-para-desarrolladores>.
56. **Jquery.com.** Jquery. [En línea] [Citado el: 2 de 2 de 2014.] <http://jquery.com/>.
57. **JavaScript Ya.** ¿Qué es el jQuery? *javascriptya.* [En línea] [Citado el: 2 de 2 de 2014.] <http://www.javascriptya.com.ar/jquery/temarios/descripcion.php?cod=57&punto=1&inicio=0>.
58. **HTML5fácil.** Guía básica para aprender Node.js. [En línea] [Citado el: 2 de 2 de 2014.] <http://html5facil.com/tips/guia-basica-para-aprender-node-js>.



59. **Muñoz, Rafa.** Introducción a Node.js. [En línea] [Citado el: 2 de 2 de 2014.]
<http://www.rmunoiz.net/introduccion-a-node-js.html>.
60. **DesarrolloWeb.com.** CreateJS, herramientas JavaScript para desarrolladores. [En línea] [Citado el: 10 de 05 de 2014.] http://www.desarrolloweb.com/de_interes/createjs-herramientas-javascript-desarrolladores-7003.html.
61. **Benvie, Brandon y Morteza, Milani.** AppJS. *AppJS*. [En línea] 2012. [Citado el: 10 de 05 de 2014.] <http://appjs.org/>.
62. **Ecured.** IDE de Programación. [En línea] [Citado el: 2 de 2 de 2014.]
http://www.ecured.cu/index.php/IDE_de_Programaci%C3%B3n.
63. **Netbeans.** NetBeans IDE 7.4 Information. [En línea] [Citado el: 2 de 2 de 2014.]
<https://netbeans.org/community/releases/74/>.
64. **IBM WebSphere Commerce.** Guías de programación y aprendizaje. [En línea] [Citado el: 30 de 04 de 2014.]
<ftp://ftp.software.ibm.com/software/websphere/commerce/55/es/WC55ProgrammingGuide.pdf>.
65. **Bahit , Eugenia.** Monografias.com. *El paradigma de la Programación Orientada a Objetos en PHP y el patrón de arquitectura de Software MVC*. [En línea] [Citado el: 30 de 04 de 2014.]
<http://www.monografias.com/trabajos89/poo-y-mvc-php/poo-y-mvc-php2.shtml>.
66. **Itescam.** UML y PD. [En línea] 2005-2006. [Citado el: 30 de 04 de 2014.]
<http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r43251.PDF>.
67. **Tedeschi, Nicolás.** ¿Qué es un Patrón de Diseño? *Microsoft*. [En línea] [Citado el: 30 de 04 de 2014.] <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.
68. **Saavedra, Jorge.** PATRONES GRASP (Patrones de Software para la asignación General de Responsabilidad). Parte II. *El Mundo Informático*. [En línea] [Citado el: 30 de 04 de 2014.]
<http://jorgesaaavedra.wordpress.com/2007/05/08/patrones-grasp-patrones-de-software-para-la-asignacion-general-de-responsabilidadparte-ii/>.
69. **Elwebmaster.com.** Top 5 Javascript Frameworks. [En línea] 04 de 10 de 2007. [Citado el: 30 de 04 de 2014.] <http://www.elwebmaster.com/articulos/top-5-javascript-frameworks..>
70. **CISIAD.** Centro de Investigación sobre Sistemas Inteligentes de Ayuda a la Decisión- Universidad Nacional de Educación a Distancia (UNED). *Proyecto Carmen-Estilos de codificación*. [En línea] [Citado el: 30 de 04 de 2014.] <http://www.cisiad.uned.es/carmen/estilo-codificacion.pdf>.
71. **Agras Hechevarría, Ramiro y Delgado del Río, Danay.** *Análisis, diseño e implementación de la versión 2.0 de los juegos del primer ciclo de la Colección Multisaber*. Ciudad Habana : s.n., 2010. s.n.



72. **Marcos Argüelles, Pedro R.** Escritura estilo CamelCase. [En línea] [Citado el: 30 de 04 de 2014.] <http://www.pedrormarcos.com/escritura-estilo-camelcase/>.
73. **Hernandez , Leovigilda .** MODELO DE IMPLEMENTACIÓN. [En línea] [Citado el: 15 de 05 de 2014.] <http://ithleovi.blogspot.com/2013/06/unidad-5-modelo-deimplementacion-el.html>.
74. **Wikiuml.** Diagrama de Componentes. [En línea] [Citado el: 15 de 05 de 2014.] <http://wikiuml.wikispaces.com/Diagrama+de+Componentes>.
75. **Pressman, Roger A.** *Ingeniería de software: Un enfoque práctico*. 5. 2005. pág. 404. 970105473.
76. **Fuentes Krafczyk, Joaquín Federico.** Universidad de las Américas Puebla. *Realidad virtual aplicada al tratamiento del trastorno de lateralidad y ubicación espacial*. [En línea] [Citado el: 15 de 05 de 2014.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/fuentes_k_jf/capitulo4.pdf.
77. **LONDOÑO, Jorge Hernan Abad.** Ingeniería de Software: TIPOS DE PRUEBAS DE SOFTWARE. *Ingeniería de Software*. [En línea] 06 de 04 de 2005. [Citado el: 15 de 05 de 2014.] <http://ing-sw.blogspot.com/2005/04/tipos-de-pruebas-de-software.html>.



Anexos

Anexo 1:

Tabla 11. Descripción del CU “Iniciar juego” del juego Sopa de letras.

Caso de Uso	<i>Iniciar Juego</i>	
Actores	<i>Jugador</i>	
Resumen	<i>El caso de uso inicia una vez que el jugador selecciona el juego. El sistema carga el juego, muestra el comic inicial y la pantalla principal del juego. Si el jugador selecciona la opción salir el sistema salva los datos y termina juego, finalizando así el caso de uso.</i>	
Complejidad	<i>Alta</i>	
Prioridad	<i>Crítico</i>	
Precondiciones	<i>El jugador debe estar autenticado en la aplicación.</i>	
Postcondiciones	<i>Se inició el juego.</i>	
Flujo normal de eventos		
	Actor	Sistema
	1. Selecciona el juego.	<p>1.2 El sistema analiza el jugador (es) y brinda las siguientes opciones:</p> <ul style="list-style-type: none"> • Si el jugador ha jugado anteriormente el juego y no ha culminado todos los niveles muestra una interfaz con el nivel actual en que se encuentra y las siguientes opciones: <ul style="list-style-type: none"> - “Continuar”: ir al paso 2.1 de la sección 2 “Comenzar a jugar”. - “Comenzar desde el principio”: ir al paso 2 del flujo básico. • Si hay más de un jugador autenticados en el sistema habilita el nivel multijugador: ir a la sección 3 “Multijugador”. • Si el jugador no ha jugado, el sistema permite comenzar desde el principio.
	2. Selecciona la opción “Comenzar desde el principio”.	<p>a. Muestra la historia del juego en forma de comic.</p> <p>b. Muestra una interfaz que permite al jugador seleccionar una de las siguientes opciones:</p> <ul style="list-style-type: none"> • Nivel de entrenamiento: ir a la sección 1 “Nivel de entrenamiento”. • Jugar: ir a la sección 2 “Comenzar a jugar”.



Sección 1: “Nivel de entrenamiento”: Flujo Normal de Eventos	
Actor	Sistema
1. Selecciona la opción “Nivel de entrenamiento”.	1.1 Muestra una sopa de letras a resolver. 1.2 Muestra una serie de instrucciones y pasos que el jugador debe seguir para completar la sopa de letras.
2. Completa la sopa de letras siguiendo las instrucciones.	2.1 Muestra un mensaje felicitando al jugador por haber resuelto la sopa de letras. 2.2 Permite seleccionar el modo de juego: ir a la sección 2 “Comenzar a jugar”, paso 1.1. 2.3 Termina el caso de uso.
Sección 2: “Comenzar a jugar”: Flujo Normal de Eventos	
Actor	Sistema
1. Selecciona la opción “Iniciar juego”.	1.1 Muestra una interfaz que permite al jugador seleccionar el modo de juego: <ul style="list-style-type: none">• Marinero• Capitán
2. Selecciona el modo de juego “Marinero”.	2.1 Muestra la interfaz del juego. 2.2 Muestra un listado con las palabras a buscar en la sopa de letras. 2.3 Muestra una matriz de 12x12 que contiene una sopa de letra a resolver (ver CU Realizar Sopa). 2.4 Muestra la construcción en la que se está trabajando. 2.5 Muestra cuando un jugador obtiene una pista (ver CU Usar pista). 2.6 Muestra el contador del tiempo. 2.7 Muestra la puntuación del jugador.
3.	3.1 Termina el caso de uso
Flujos alternos: 2.a Capitán	
1. Selecciona un modo de juego “Capitán”.	1.8 Muestra la interfaz del juego. 1.9 Muestra un listado con imágenes de las palabras a buscar en la sopa de letras. 1.10 Muestra una matriz de 12x12 que contiene una sopa de letra a resolver (ver CU Realizar Sopa). 1.11 Muestra la construcción en la que se está trabajando. 1.12 Muestra cuando un jugador obtiene una pista (ver CU Usar pista). 1.13 Muestra el contador del tiempo.



	1.14 Muestra la puntuación del jugador.
2.	2.1 Termina el caso de uso.
Sección 3: "Multijugador": Flujo Normal de Eventos	
Actor	Sistema
1.	1.1 Muestra el jugador en turno. 1.2 Muestra el contador de tiempo. 1.3 Muestra la sopa de letras a resolver. 1.4 Muestra cuando un jugador obtiene una pista (ver CU Usar pista). 1.5 Muestra el tiempo transcurrido.
2. Selecciona una primera letra.	2.1 Marca la letra seleccionada.
3. Selecciona una segunda letra.	3.1 Verifica si la palabra comprendida entre estas dos letras está correcta o no en cualquier dirección (horizontal, vertical, diagonal). 3.2 Marca la palabra. 3.3 Tacha la palabra en el listado de palabras a buscar. 3.4 Actualiza la puntuación.
4. Encuentra todas las palabras.	4.1 Verifica que la sopa de letras está completa. 4.2 Muestra la puntuación obtenida por el jugador. 4.3 Le concede el turno al próximo jugador. y se comenzará a ejecutar nuevamente el flujo básico de eventos a partir del paso 1.1. 4.4 Cuando terminen de jugar todos los jugadores se mostrarán los resultados siguientes: <ul style="list-style-type: none"> • Nombre de cada jugador • Puntos obtenidos. • Lugar alcanzado. 4.5 Termina el caso de uso.
Flujos alternos: 4.a Tiempo agotado.	
4.	4.1 Notifica que el tiempo se agotó. 4.2 Muestra la puntuación obtenida por el jugador. 4.3 Le concede el turno al próximo jugador. y se comenzará a ejecutar nuevamente el flujo básico de eventos a partir del paso 1.1 de la sección 3. 4.4 Cuando terminen de jugar todos los jugadores se mostrarán los resultados siguientes: <ul style="list-style-type: none"> • Nombre de cada jugador



		<ul style="list-style-type: none"> • Puntos obtenidos. • Lugar alcanzado. <p>4.5 Termina el caso de uso.</p>
Relaciones	CU Incluidos	Realizar sopa
	CU Extendidos	Cambiar nivel

Anexo 2:

Tabla 12. Descripción del CU “Usar pista” del juego Sopa de letras.

Caso de Uso	<i>Usar pista.</i>
Actores	<i>Jugador</i>
Resumen	<i>El caso de uso inicia cuando encuentra tres palabras antes del primer minuto de juego. Termina cuando haya utilizado la pista.</i>
Complejidad	<i>Media</i>
Prioridad	<i>Secundario</i>
Precondiciones	<i>Se encontraron tres palabras antes del primer minuto.</i>
Postcondiciones	<i>Se utiliza la pista.</i>
Flujo normal de eventos	
Actor	Sistema
1.	1.1 Verifica que se hayan encontrado tres palabras antes del primer minuto. 1.2 Otorga una pista al jugador.
2. Utiliza la pista.	2.1 Marca el inicio la letra inicial de una palabra en la sopa de letras. 2.2 Disminuye el contador de pistas. 2.3 Permite seguir jugando. 2.4 Termina el caso de uso.

Anexo 3:

Tabla 13. Descripción del CU “Cambiar nivel” (común para todos los juegos).

Caso de Uso	<i>Cambiar nivel</i>
Actores	<i>Jugador</i>
Resumen	<i>El caso de uso inicia cuando el jugador culmina un nivel satisfactoriamente. Termina cuando el jugador vence el último nivel.</i>
Complejidad	<i>Media</i>



Prioridad	<i>Crítico</i>
Precondiciones	<i>Completar correctamente la sopa de letras</i>
Postcondiciones	<i>Cambió el nivel</i>
Flujo normal de eventos	
Actor	Sistema
	1.1 Muestra una interfaz al jugador que contiene: <ul style="list-style-type: none"> • Nivel culminado. • Puntuación obtenida. • Tiempo transcurrido. • Bonificación. 1.2 Brinda la opción “Aceptar” para pasar al siguiente nivel.
1. Selecciona la opción Aceptar.	1.1 Guarda el estado del jugador: <ul style="list-style-type: none"> • Nivel en que se encuentra. • Puntuación obtenida hasta el momento. • Modo de juego. 1.2 Carga los datos del nuevo nivel (Ver CU Iniciar Juego sección 2 “Comenzar a jugar” paso 2.1). 1.3 Termina el caso de uso.
Flujos alternos: 1.b Último nivel.	
1. Completa el juego.	a. Muestra una ventana felicitando al jugador por haber ganado el juego. b. Termina el caso de uso

Anexo 4:

Tabla 14. Descripción del CU “Terminar juego” (común para todos los juegos).

Caso de Uso	<i>Terminar juego</i>
Actores	<i>Jugador</i>
Resumen	<i>El caso de uso inicia cuando el jugador selecciona la opción Salir del juego. Termina cuando se cierra el juego.</i>
Complejidad	<i>Media</i>
Prioridad	<i>Secundario</i>
Precondiciones	<i>El jugador debe encontrarse dentro del juego.</i>
Postcondiciones	<i>Se termina el juego.</i>
Flujo de eventos	
Actor	Sistema



1. Selecciona la opción Salir del juego.	1.1 Muestra un mensaje de confirmación de la acción 1.2 “¿Está seguro que desea salir del juego?”.
2. Selecciona la opción “Si”.	2.1 Guarda el estado del jugador. 2.2 Termina el caso de uso.
Flujos alternos: 2.a. Cancelar salida del juego	
1. Selecciona la opción “No”.	1.1 Continúa en el juego (Ver caso de uso Realizar Juego). 1.2 Termina el caso de uso.

Anexo 5:

Tabla 15. Descripción del CU “Iniciar juego” del juego Crucigrama.

Caso de Uso	<i>Iniciar Juego.</i>
Actores	<i>Jugador</i>
Resumen	<i>El caso de uso inicia una vez que el jugador selecciona el juego. El sistema carga el juego, muestra el comic inicial y la pantalla principal del juego. Si el jugador selecciona la opción salir el sistema salva los datos y termina juego, finalizando así el caso de uso.</i>
Complejidad	<i>Alta</i>
Prioridad	<i>Crítico</i>
Precondiciones	<i>El jugador debe estar autenticado en la aplicación.</i>
Postcondiciones	<i>Se inició el juego.</i>
Flujo normal de eventos	
Actor	Sistema
1. Selecciona el juego.	<p>1.1 El sistema analiza el jugador (es) y brinda las siguientes opciones:</p> <ul style="list-style-type: none"> • Si el jugador ha jugado anteriormente el juego y no ha culminado todos los niveles muestra una interfaz con el nivel actual en que se encuentra y las siguientes opciones: <ul style="list-style-type: none"> - “Continuar”: ir al paso 2.1 de la sección 2 “Comenzar a jugar”. - “Comenzar desde el principio”: ir al paso 2 del flujo básico. • Si hay más de un jugador autenticados en el sistema habilita el nivel multijugador: ir a la sección 3 “Multijugador”. • Si el jugador no ha jugado el sistema permite comenzar desde el principio.



<p>2. Selecciona la opción “Comenzar desde el principio”.</p>	<p>2.1 Muestra la historia del juego en forma de comic. 2.2 Muestra una interfaz que permite al jugador seleccionar una de las siguientes opciones:</p> <ul style="list-style-type: none"> • Nivel de entrenamiento: ir a la sección 1 “Nivel de entrenamiento”. • Jugar: ir a la sección 2 “Comenzar a jugar”.
<p>Sección 1: “Nivel de entrenamiento”: Flujo Normal de Eventos</p>	
<p>Actor</p>	<p>Sistema</p>
<p>1. Selecciona la opción “Nivel de entrenamiento”.</p>	<p>1.1 Muestra un crucigrama a resolver 1.2 Muestra una serie de mensajes explicando reglas del juego e instrucciones del mismo.</p>
<p>2. Completa el crucigrama siguiendo las instrucciones.</p>	<p>2.1 Muestra un mensaje felicitando al jugador por haber completado el crucigrama. 2.2 Permite seleccionar el modo de juego: ir a la sección 2 “Comenzar a jugar”, paso 1.1. 2.3 Termina el caso de uso</p>
<p>Sección 2: “Comenzar a jugar”: Flujo Normal de Eventos</p>	
<p>Actor</p>	<p>Sistema</p>
<p>1. Selecciona la opción “Iniciar juego”.</p>	<p>1.1 Muestra una interfaz que permite al jugador seleccionar el modo de juego:</p> <ul style="list-style-type: none"> • Fácil • Normal
<p>2. Selecciona el modo de juego “Fácil”.</p>	<p>2.1 Muestra la interfaz del juego 2.2 Muestra el crucigrama a resolver (ver CU Realizar Crucigrama). 2.3 Muestra las consonantes de las palabras que contiene el crucigrama. 2.4 Muestra un alfabeto. 2.5 Muestra el contador del tiempo. 2.6 Muestra la puntuación del jugador.</p>
<p>3.</p>	<p>3.1 Termina el caso de uso</p>
<p>Flujos alternos: 2.a Normal</p>	
<p>1. Selecciona un modo de juego “Normal”.</p>	<p>1.1 Muestra la interfaz del juego. 1.2 Muestra el crucigrama a resolver (ver CU Realizar Crucigrama). 1.3 Muestra las pista disponibles en cada nivel (ver CU Usar</p>



	<p>pista).</p> <p>1.4 Muestra un alfabeto.</p> <p>1.5 Muestra el contador del tiempo.</p> <p>1.6 Muestra la puntuación del jugador.</p>
2.	2.1 Termina el caso de uso.
Sección 3: "Multijugador": Flujo Normal de Eventos	
Actor	Sistema
1.	<p>1.1 Muestra el jugador en turno.</p> <p>1.2 Muestra el contador de tiempo.</p> <p>1.3 Muestra el crucigrama a resolver.</p> <p>1.4 Muestra el tiempo transcurrido.</p>
2. Posiciona el puntero en el inicio de una palabra del crucigrama.	2.1 Muestra por cada dirección (horizontal o vertical) el significado de la palabra a completar.
3. Selecciona una casilla del crucigrama	3.1 Marca la casilla.
4. Introduce la letra por teclado o por el alfabeto del juego.	4.1 Coloca la letra en la casilla anteriormente seleccionada.
5.	<p>5.1 Verifica que la palabra completada sea correcta.</p> <p>5.2 Marca la palabra completada.</p>
6. Completa todas las palabras.	<p>6.1 Verifica que el crucigrama está completo.</p> <p>6.2 Muestra la puntuación obtenida por el jugador</p> <p>6.3 Le concede el turno al próximo jugador. y se comenzará a ejecutar nuevamente el flujo básico de eventos a partir del paso 1.1.</p> <p>6.4 Cuando terminen de jugar todos los jugadores se mostrará los resultados siguientes:</p> <ul style="list-style-type: none"> • Nombre de cada jugador • Puntos obtenidos. • Lugar alcanzado. <p>6.5 Termina el caso de uso.</p>
Flujos alternos: 6.a Tiempo agotado.	
6.	<p>6.1 Notifica que el tiempo se agotó.</p> <p>6.2 Muestra la puntuación obtenida por el jugador</p> <p>6.3 Le concede el turno al próximo jugador y se comenzará a ejecutar nuevamente el flujo básico de eventos a partir del paso 1.1.</p>



		<p>6.4 Cuando terminen de jugar todos los jugadores se mostrará los resultados siguientes:</p> <ul style="list-style-type: none"> • Nombre de cada jugador. • Puntos obtenidos. • Lugar alcanzado. <p>6.5 Termina el caso de uso.</p>
Relaciones	CU Incluidos	Realizar crucigrama
	CU Extendidos	Cambiar nivel

Anexo 6:

Tabla 16. Descripción del CU “Realizar crucigrama” del juego Crucigrama.

Caso de Uso	<i>Realizar crucigrama</i>	
Actores	<i>Jugador</i>	
Resumen	<i>El caso de uso inicia cuando el jugador selecciona el modo de juego. Termina cuando haya completado el crucigrama que muestra el juego.</i>	
Complejidad	<i>Alta</i>	
Prioridad	<i>Crítico</i>	
Precondiciones	<i>Se visualiza el crucigrama a completar.</i>	
Postcondiciones	<i>Se completa el crucigrama.</i>	
Flujo normal de eventos		
Actor	Sistema	
1.	1.1 Muestra el crucigrama a resolver.	
2. Posiciona el puntero en el inicio de una palabra del crucigrama.	2.1 Muestra por cada dirección (horizontal o vertical) el significado de la palabra a completar.	
3. Selecciona una casilla del crucigrama	3.1 Marca la casilla.	
4. Introduce la letra por teclado o por el alfabeto del juego.	4.1 Coloca la letra en la casilla anteriormente seleccionada.	
5.	5.1 Verifica que la palabra completada sea correcta. 5.2 Marca la palabra completada.	
6. Encuentra todas las palabras.	6.1 Verifica que todas las palabras hayan sido encontradas. 6.2 Cambia de nivel (ver CU Cambiar nivel). 6.3 Termina el caso de uso.	



Anexo 7:

Tabla 17. Descripción del CU “Usar pista” del juego Crucigrama.

Caso de Uso	<i>Usar pista</i>
Actores	<i>Jugador</i>
Resumen	<i>El caso de uso inicia cuando el jugador utiliza una pista. Termina cuando haya utilizado la pista</i>
Complejidad	<i>Media</i>
Prioridad	<i>Secundario</i>
Precondiciones	<i>Se muestran las pistas al comenzar cada nivel en modo normal.</i>
Postcondiciones	<i>Se utiliza la pista</i>
Flujo normal de eventos	
Actor	Sistema
1.	1.1 Muestra 3 pistas disponibles por cada nivel.
2. Utiliza la pista.	2.1 Ubica una letra de forma aleatoria en cualquier casilla vacía del crucigrama. 2.2 Disminuye en uno el contador de pistas disponibles. 2.3 Termina el caso de uso.

Anexo 8:

Tabla 18. Descripción del CU “Realizar Sudoku” del juego Sudoku.

Caso de Uso	<i>Realizar Sudoku</i>
Actores	<i>Jugador</i>
Resumen	<i>El caso de uso inicia cuando el jugador selecciona el modo de juego. Termina cuando haya completado el sudoku que muestra el juego.</i>
Complejidad	<i>Alta</i>
Prioridad	<i>Crítico</i>
Precondiciones	<i>Se visualiza el sudoku a completar</i>
Postcondiciones	<i>Se completa el sudoku</i>
Flujo normal de eventos	
Actor	Sistema
1.	1.1 Muestra el sudoku a resolver.
2. Selecciona un objeto (número o figura, puede seleccionarlo tanto por teclado como por el tablero de objetos).	2.1 Resalta el objeto seleccionado.



3. Coloca el objeto en una o varias casilla(s).	3.1 Verifica el elemento este bien ubicado. 3.2 Pinta el objeto de color verde.
4. Completa el sudoku.	3.3 Cambia de nivel (ver CU Cambiar nivel). 3.4 Termina el caso de uso.
Flujos alternos: 3.a Elemento mal ubicado	
1. Coloca el objeto en una o varias casilla(s).	1.1 Verifica el elemento este mal ubicado. 1.2 Pinta el objeto de color rojo.
2. Completa el sudoku.	2.1 Cambia de nivel (ver CU Cambiar nivel). 2.2 Termina el caso de uso.
Flujos alternos: 4.a Tiempo agotado	
1.	1.4 Verifica si se acabó el tiempo límite sin que el jugador haya completado el sudoku. 1.5 Muestra una interfaz informando al jugador que su tiempo se ha agotado. 1.6 Brinda las opciones: <ul style="list-style-type: none"> • Reiniciar el nivel. • Salir (ir al CU "Terminar juego" paso 2.1).
2. Selecciona la opción "Reiniciar nivel".	2.1 Carga nuevos datos del mismo nivel (ver CU Iniciar Juego sección 2 "Comenzar a jugar" paso 2.1). 2.2 Termina el caso de uso.

Anexo 9:

Tabla 19. Descripción del CU "Iniciar Juego" del juego Pasatiempo matemático.

Caso de Uso	<i>Iniciar Juego</i>
Actores	<i>Jugador</i>
Resumen	<i>El caso de uso inicia una vez que el jugador selecciona el juego. El sistema carga el juego, muestra el comic inicial y la pantalla principal del juego. Si el jugador selecciona la opción salir el sistema salva los datos y termina juego, finalizando así el caso de uso</i>
Complejidad	<i>Alta</i>
Prioridad	<i>Crítico</i>
Precondiciones	<i>El jugador debe estar autenticado en la aplicación.</i>
Postcondiciones	<i>Se inició el juego.</i>
Flujo normal de eventos	
Actor	Sistema



<p>1. Selecciona el juego.</p>	<p>1.1 El sistema analiza el jugador (es) y brinda las siguientes opciones:</p> <ul style="list-style-type: none"> • Si el jugador ha jugado anteriormente el juego y no ha culminado todos los niveles muestra una interfaz con el nivel actual en que se encuentra y las siguientes opciones: <ul style="list-style-type: none"> - “Continuar”: ir al paso 2.1 de la sección 2 “Comenzar a jugar”. - “Comenzar desde el principio”: ir al paso 2 del flujo básico. • Si hay más de un jugador autenticados en el sistema habilita el nivel multijugador: ir a la sección 3 “Multijugador”. • Si el jugador no ha jugado el sistema permite comenzar desde el principio.
<p>2. Selecciona la opción “Comenzar desde el principio”.</p>	<p>2.1 Muestra la historia del juego en forma de comic. 2.2 Muestra una interfaz que permite al jugador seleccionar una de las siguientes opciones:</p> <ul style="list-style-type: none"> • Nivel de entrenamiento: ir a la sección 1 “Nivel de entrenamiento”. • Jugar: ir a la sección 2 “Comenzar a jugar”.
<p>Sección 1: “Nivel de entrenamiento”: Flujo Normal de Eventos</p>	
<p>Actor</p>	<p>Sistema</p>
<p>1. Selecciona la opción “Nivel de entrenamiento”.</p>	<p>1.1 Muestra un tablero matemático a resolver. 1.2 Muestra una serie de instrucciones y pasos que el jugador debe seguir para completar el tablero matemático.</p>
<p>2. Completa el tablero matemático siguiendo las instrucciones.</p>	<p>2.1 Muestra un mensaje felicitando al jugador por haber completado el tablero matemático. 2.2 Permite seleccionar el modo de juego: ir a la sección 2 “Comenzar a jugar”, paso 1.1. 2.3 Termina el caso de uso.</p>
<p>Sección 2: “Comenzar a jugar”: Flujo Normal de Eventos</p>	
<p>Actor</p>	<p>Sistema</p>
<p>1. Selecciona la opción “Iniciar juego”.</p>	<p>1.1 Muestra una interfaz que permite al jugador seleccionar el modo de juego:</p> <ul style="list-style-type: none"> • Aprendiz. • Genio.



2. Selecciona el modo de juego "Aprendiz".	<p>2.1 Muestra la interfaz del juego</p> <p>2.2 Muestra un panel numérico con los números (1-9) a utilizar para completar el tablero matemático.</p> <p>2.3 Muestra una matriz que contiene un tablero de 2x2 operaciones matemáticas (ver CU Realizar Tablero Matemático).</p> <p>2.4 Muestra el porcentaje del juego completado.</p> <p>2.5 Muestra el contador del tiempo.</p> <p>2.6 Muestra la puntuación del jugador.</p>
3.	3.1 Termina el caso de uso
Flujos alternos: 2.a Genio	
1. Selecciona un modo de juego "Genio".	<p>1.1 Muestra la interfaz del juego.</p> <p>1.2 Muestra un panel numérico con los números (1-9) a utilizar para completar el tablero matemático.</p> <p>1.3 Muestra una matriz que contiene un tablero de 3x3 operaciones matemáticas (ver CU Realizar Tablero Matemático).</p> <p>1.4 Muestra el porcentaje del juego completado.</p> <p>1.5 Muestra el contador del tiempo.</p> <p>1.6 Muestra la puntuación del jugador.</p>
2.	2.1 Termina el caso de uso.
Sección 3: "Multijugador": Flujo Normal de Eventos	
Actor	Sistema
1.	<p>1.1 Muestra el jugador en turno.</p> <p>1.2 Muestra el contador de tiempo.</p> <p>1.3 Muestra el tablero matemático a resolver.</p> <p>1.4 Muestra un panel numérico con los números (1-9) a utilizar para completar el tablero matemático.</p> <p>1.5 Muestra el tiempo transcurrido.</p>
2. Selecciona un número.	2.1 Marca el número seleccionado.
3. Selecciona una casilla.	3.1 Coloca el número en la casilla.
4.	<p>4.1 Verifica que los números colocados satisfagan la igualdad que los condiciona.</p> <p>4.2 Encierra el resultado en un cuadro de color verde.</p>
5. Completa el tablero matemático.	<p>5.1 Verifica que el tablero matemático este completo.</p> <p>5.2 Muestra la puntuación obtenida por el jugador.</p> <p>5.3 Le concede el turno al próximo jugador. y se comenzará a</p>



		<p>ejecutar nuevamente el flujo básico de eventos a partir del paso 1.1.</p> <p>5.4 Cuando terminen de jugar todos los jugadores se mostrará los resultados siguientes:</p> <ul style="list-style-type: none"> • Nombre de cada jugador • Puntos obtenidos. • Lugar alcanzado. <p>5.5 Termina el caso de uso.</p>
Flujos alternos: 5.a Tiempo agotado.		
5.		<p>5.1 Notifica que el tiempo se agotó.</p> <p>5.2 Muestra la puntuación obtenida por el jugador.</p> <p>5.3 Le concede el turno al próximo jugador. y se comenzará a ejecutar nuevamente el flujo básico de eventos a partir del paso 1.1.</p> <p>5.4 Cuando terminen de jugar todos los jugadores se mostrará los resultados siguientes:</p> <ul style="list-style-type: none"> • Nombre de cada jugador • Puntos obtenidos. • Lugar alcanzado. <p>5.5 Termina el caso de uso.</p>
Relaciones	CU Incluidos	Realizar tablero matemático
	CU Extendidos	Cambiar nivel

Anexo 10:

Tabla 20. Descripción del CU “Realizar tablero matemático” del juego Pasatiempo matemático.

Caso de Uso	<i>Realizar tablero matemático</i>
Actores	<i>Jugador</i>
Resumen	<i>El caso de uso inicia cuando el jugador selecciona el modo de juego. Termina cuando haya completado el tablero matemático que muestra el juego.</i>
Complejidad	<i>Alta</i>
Prioridad	<i>Crítico</i>
Precondiciones	<i>Se visualiza el tablero matemático a completar.</i>
Postcondiciones	<i>Se completa el tablero matemático.</i>
Flujo normal de eventos	
Actor	Sistema



1.	1.1 Muestra el tablero matemático a resolver.
2. Selecciona un número.	2.1 Marca el número seleccionado.
3. Selecciona una casilla.	3.1 Coloca el número en la casilla.
4.	4.1 Verifica que los números colocados satisfagan la igualdad que los condiciona. 4.2 Encierra el resultado en un cuadro de color verde.
5. Completa el tablero matemático.	5.1 Cambia de nivel (ver CU Cambiar nivel). 5.2 Termina el caso de uso.
Flujos alternos: 5.a Tiempo agotado	
1.	1.1 Verifica si se acabó el tiempo para resolver el tablero sin que el jugador haya completado el mismo. 1.2 Muestra una interfaz informando al jugador que su tiempo se ha agotado. 1.3 Brinda las opciones: <ul style="list-style-type: none">• Reiniciar el nivel.• Salir(ir al CU "Terminar juego" paso 2.1)
2. Selecciona la opción "Reiniciar nivel".	2.1 Carga nuevos datos del mismo nivel (ver CU Iniciar Juego sección 2 "Comenzar a jugar" paso 2.1). 2.2 Termina el caso de uso.



Anexo 11:

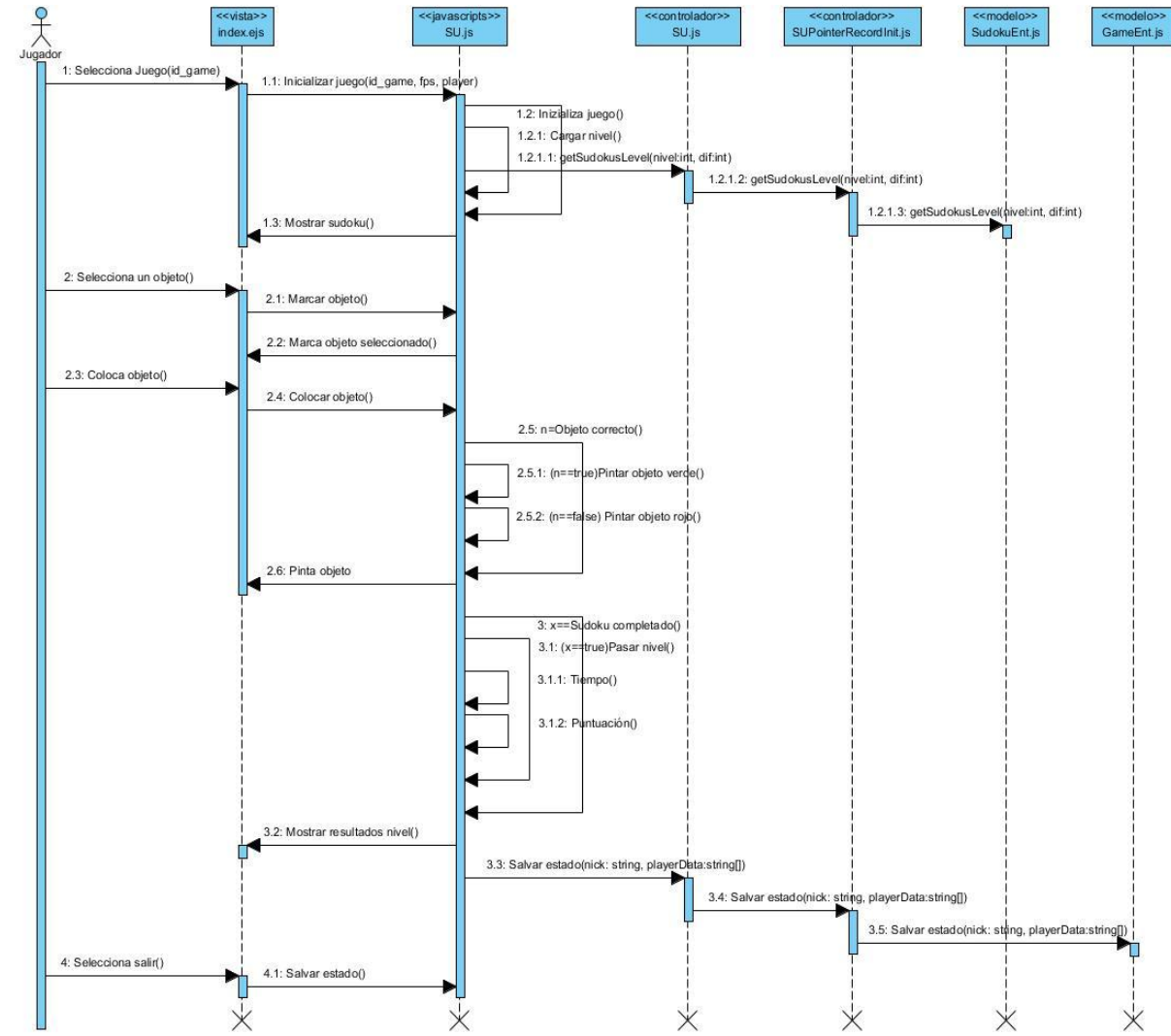


Figura 17. Diagrama de secuencia del juego Sudoku.



Anexo 12:

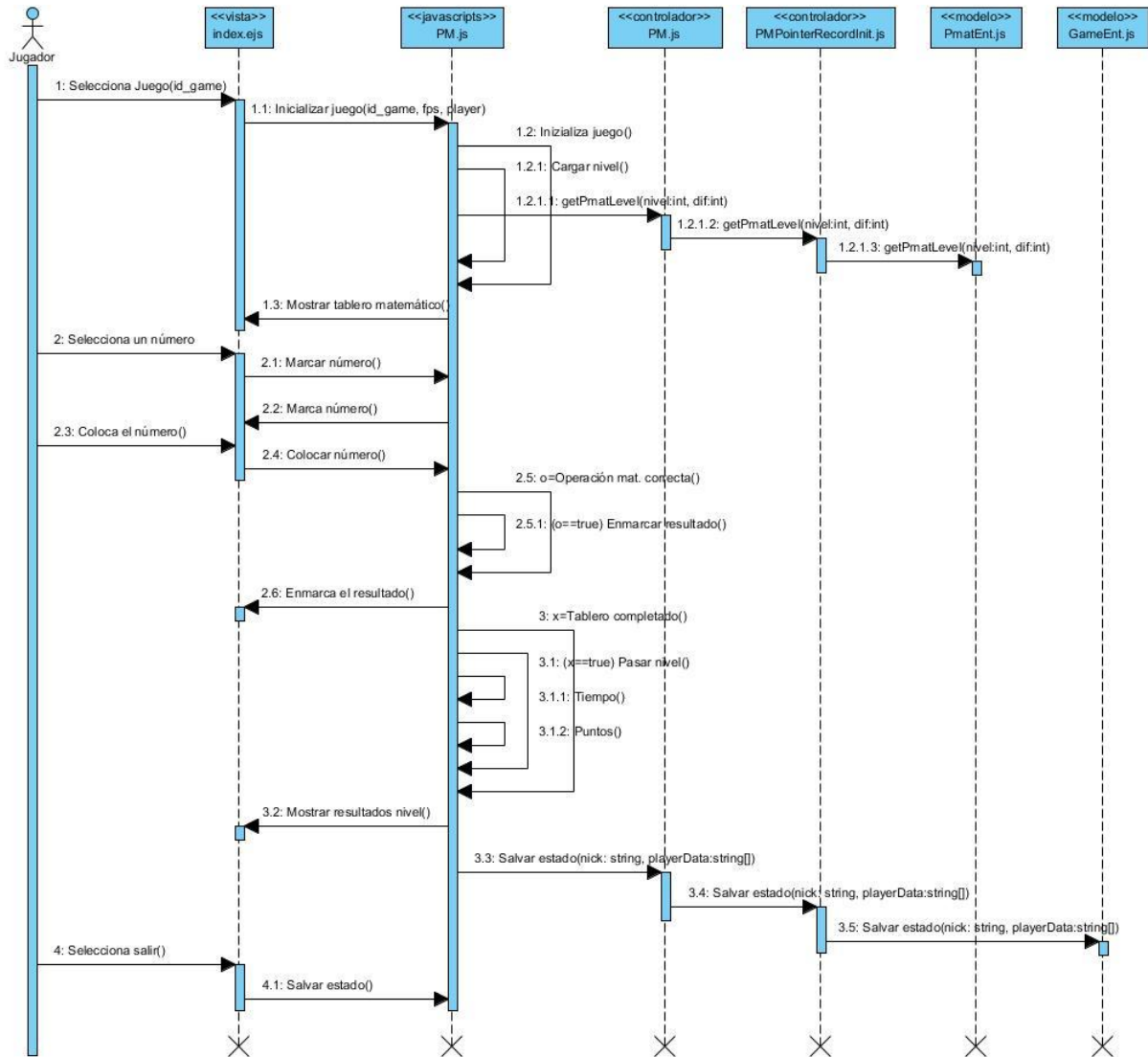


Figura 18. Diagrama de secuencia del juego Pasatiempo matemático.



Anexo 13:

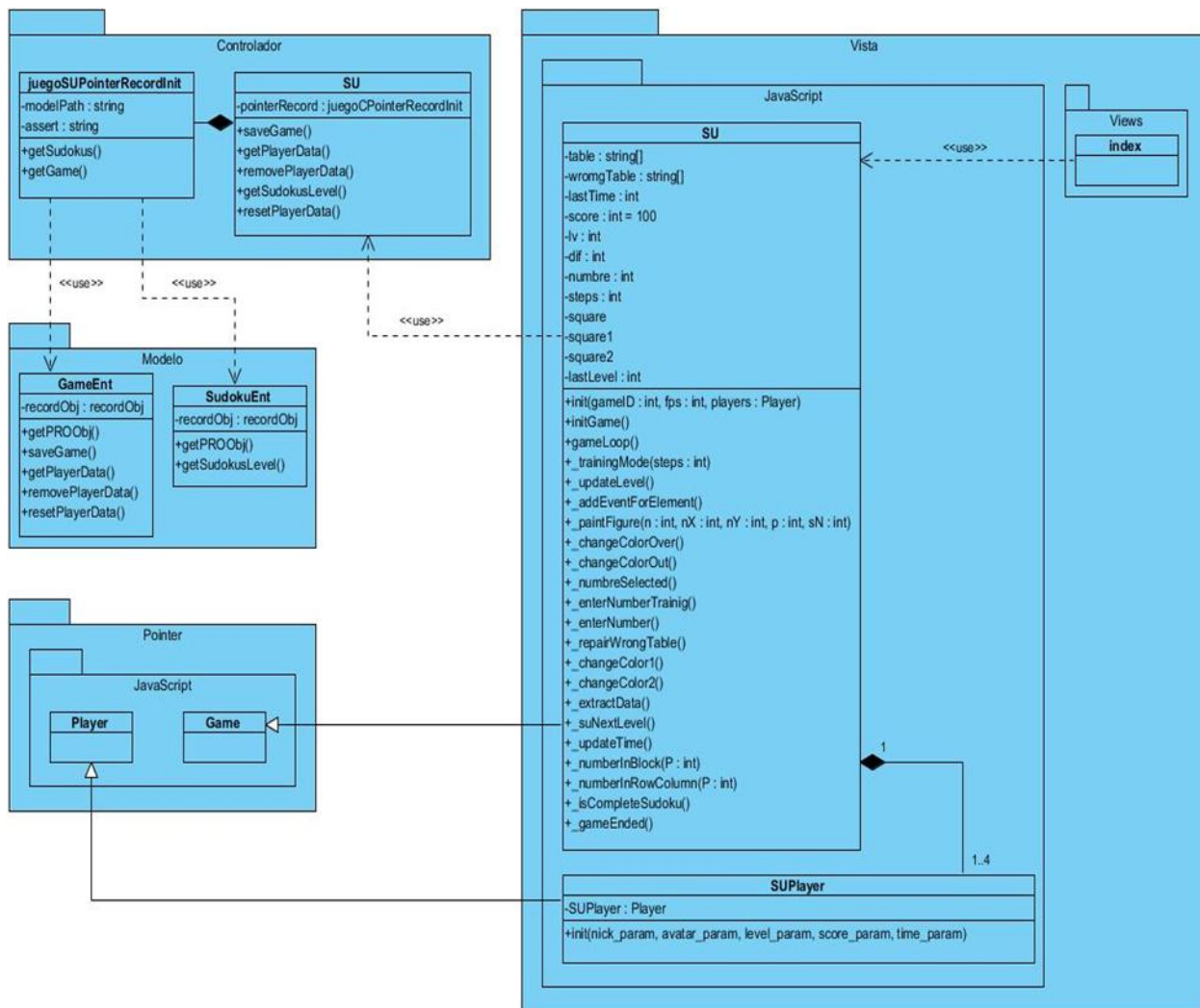


Figura 19. Diagrama de clases del juego Sudoku.

Tabla 21. Descripción de las clases del juego Sudoku.

Clase	Función
SudokuEnt.js	Permite el acceso a los datos guardados en Sudoku.json
GameEnt.js	Permite el acceso a los datos guardados en Game.json
juegoSUPointerRecordInit.js	Controla el acceso a los datos mediante las entidades.
SU.js	Controla el acceso a los datos mediante la clase juegoSUPointerRecordInit.js.
SU.js	Controla el flujo del juego.
SUPlayer.js	Gestiona los datos de los jugadores.



Anexo 14:

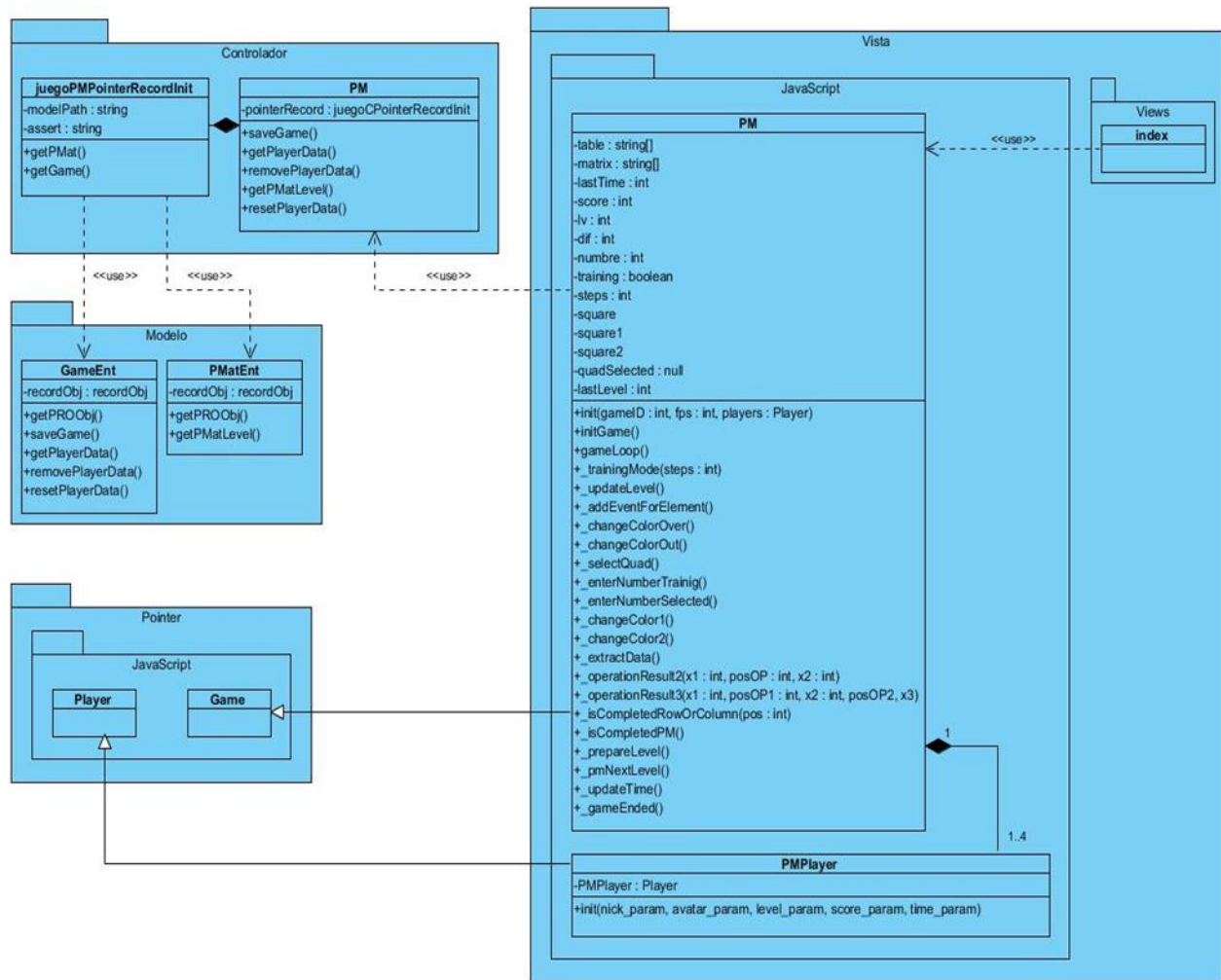


Figura 20. Diagrama de clases del juego Pasatiempo Matemático.

Tabla 22. Descripción de las clases del juego Pasatiempo Matemático.

Clase	Función
PMatEnt.js	Permite el acceso a los datos guardados en PMat.json
GameEnt.js	Permite el acceso a los datos guardados en Game.json
juegoPMPointerRecordInit.js	Controla el acceso a los datos mediante las entidades.
PM.js	Controla el acceso a los datos mediante la clase juegoPMPointerRecordInit.js.
PM.js	Controla el flujo del juego.
PMPlayer.js	Gestiona los datos de los jugadores.



Anexo 15:

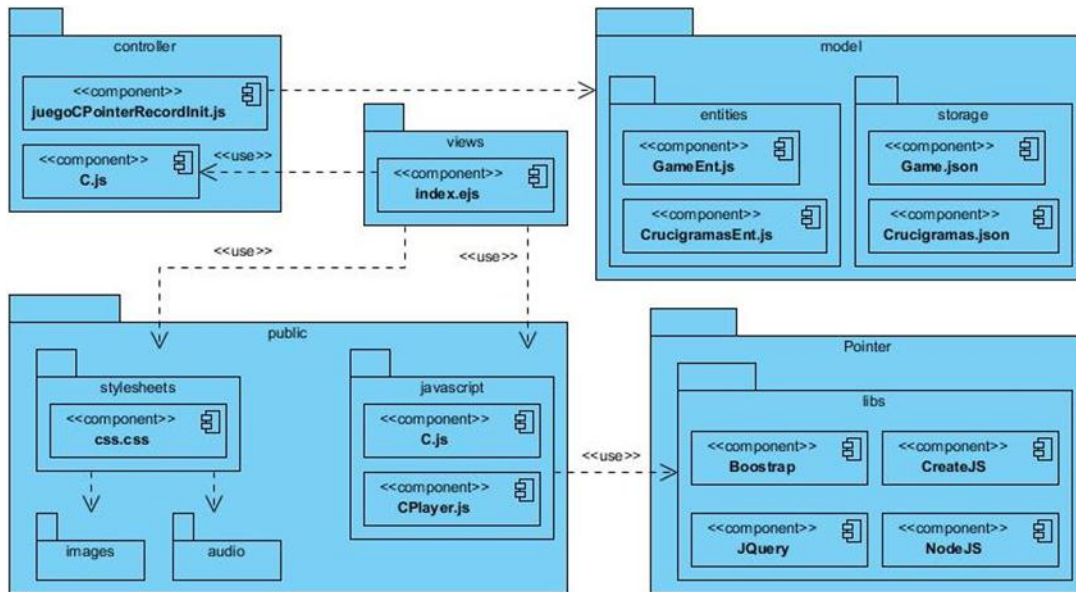


Figura 21. Diagrama de componentes del juego Crucigrama.

Anexo 16:

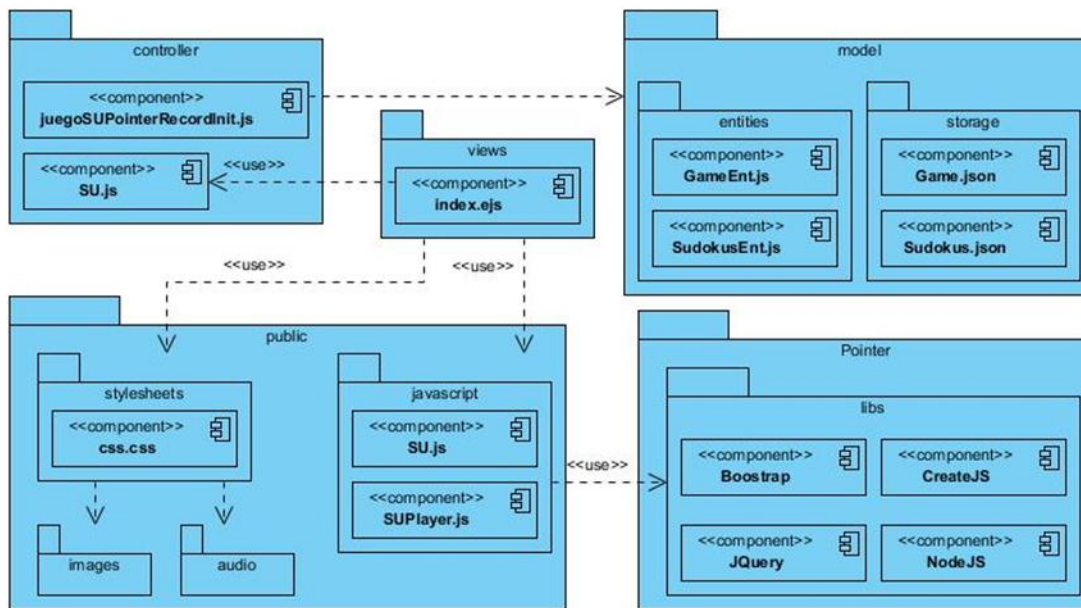


Figura 22. Diagrama de componentes del juego Sudoku.



Anexo 17:

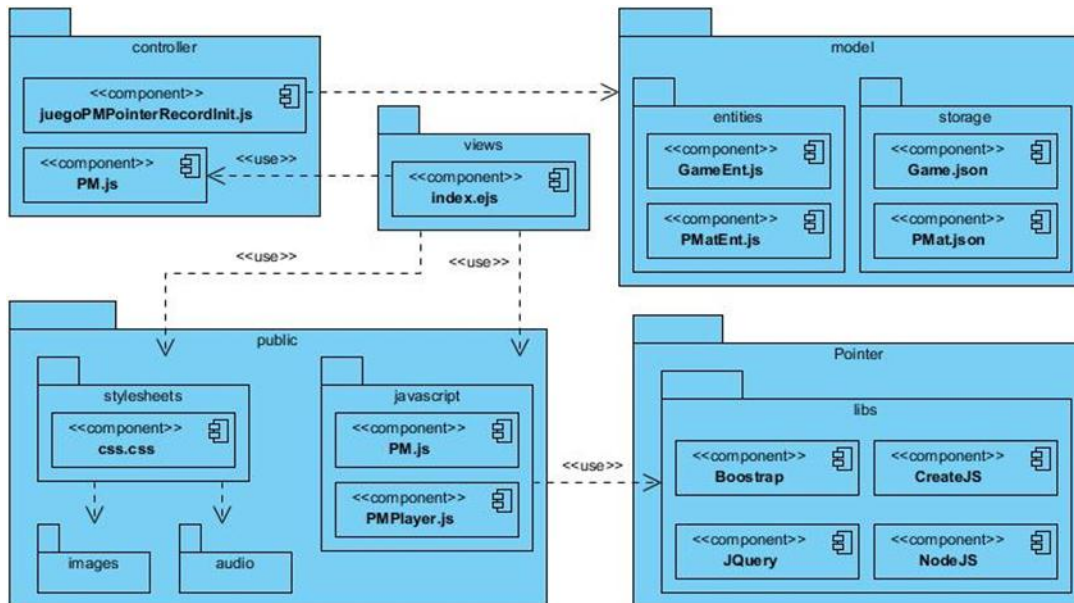


Figura 23. Diagrama de componentes del juego Pasatiempo matemático.

Anexo 18: Guión del juego Sopa de letras para la colección de juegos educativos MundoClick.

Historia

El juego consiste en una travesía marítima en donde eres el capitán de una flota compuesta por dos Calaveras, un Galeón, una Fragata y un Nao (embarcación de transporte de mercancías). La flota llega a una Isla en el medio del Caribe y decides hacer de esa isla una ciudad. Para esto debe realizar distintas sopas de letras relacionadas con la edificación que desea construir para mejorar las condiciones de vida en la isla.

Objetivo

Encontrar todas las palabras escondidas en las sopas de letras relacionadas con la edificación a construir hasta lograr que la isla sea un lugar con perfectas condiciones para vivir.

Flujo de juego

Primero se muestra la pantalla de inicio, donde se narra la historia en forma de comic. Luego en cada nivel el jugador debe resolver una sopa de letras, solo pasará al siguiente nivel cuando la misma esté resuelta correctamente. Al final cada nivel se muestra la puntuación obtenida en el



mismo. Cuando finalice el juego se muestra el desenlace del mismo en forma de comic. Mostrando al final la puntuación total obtenida.

Reglas

- En cada nivel solo puedes usar un objeto Lupa.
- Para marcar una palabra como encontrada primero tienes que hacer clic en la primera letra y después en la última.
- El listado de palabras a encontrar está colocado a la derecha de la pantalla.
- Para pasar el nivel tienes que encontrar todas las palabras.

Puntos

- Se suma la cantidad de letras que tenga cada palabra encontrada.
- Se restan 5 puntos por cada palabra mal seleccionada.

Bonificaciones

- Si soluciona correctamente la sopa de letras antes de 10 minutos se le suma una bonificación de 20 puntos.
- Si encuentra 3 palabras en menos de 1 minuto se activa la lupa que le permite localizar la primera letra de una palabra en la sopa. Si esta lupa no es usada se le suma 5 puntos al finalizar el nivel a la puntuación obtenida.

Secuencia del juego por nivel

Cada uno de los 7 niveles es una sopa de letras que el jugador debe encontrar 8 palabras contenidas en esta para poder hacer la construcción en la isla.

Dificultad Marinero:

Se muestra una sopa de letras con 8 palabras a buscar.

Dificultad Capitán:

Se muestra una sopa de letras con 8 imágenes de las palabras a buscar.



Anexo 19: Guión del juego Crucigrama para la colección de juegos educativos MundoClick.

Historia

El juego consiste en un niño que tiene un robot de juguete y quiere que sea su amigo y que se convierta en un niño de verdad para poder jugar con él. Durante la noche lo visita un hada y le dice que para concederle el deseo tiene que enseñarle correctamente el idioma español mediante un juego y cuando el robot domine este se convertirá en un niño de verdad y podrá jugar con él miles de juegos divertidos. La forma que el niño encontró para enseñarle español a su juguete fue mediante la solución de crucigramas, en donde cada crucigrama le enseñaba una clasificación distinta de palabras.

Objetivo

El objetivo del juego consiste en resolver los crucigramas correctamente para así lograr que tu robot se convierta en un niño de verdad.

Flujo de juego

Primero se muestra la pantalla de inicio, donde se narra en forma de comic la historia. Luego en cada nivel el jugador debe resolver un crucigrama, solo pasará al siguiente nivel cuando lo resuelva correctamente. Al final cada nivel se muestra la puntuación obtenida en el mismo y la puntuación acumulada durante el transcurso del juego. Cuando finalice el juego se muestra el desenlace del mismo en forma de comic. Mostrando al final la puntuación total obtenida.

Reglas

- Para colocar una letra en una casilla tienes que seleccionar la casilla y después la letra que quieres introducir o seleccionar la casilla y teclear la letra.
- Para ver la descripción de las palabras a buscar tienes que pasar el mouse por arriba de la primera casilla de la palabra o darle clic encima.
- En cada nivel tienes 3 objetos lupas que puedes usar.
- En el nivel de dificultad fácil el objeto lupa no aparece.

Puntos

- Cada nivel comienza con 100 puntos en donde cada vez que pasen 10 segundos se disminuye 1 punto, la puntuación será la que obtenga al finalizar el crucigrama.

Bonificaciones



- Si soluciona correctamente el crucigrama antes de 10 minutos se le suma una bonificación de 20 puntos.
- El niño tiene 3 lupas que puede usar en cualquier momento del nivel en donde cada lupa le coloca una letra correctamente en el crucigrama. Al finalizar el nivel se le suma 5 puntos por cada lupa no usada a la puntuación obtenida.

Secuencia del juego por nivel

Cada uno de los 7 niveles es un crucigrama que el jugador debe completar para poder enseñarle al robot el idioma español.

Nivel 1

Dificultad fácil:

Completar las 7 palabras del crucigrama que presentan diptongo solo con las vocales.

Dificultad normal:

Completar las 7 palabras del crucigrama que presentan diptongo.

Nivel 2

Dificultad fácil:

Completar las 7 palabras del crucigrama que tiene hiato solo con las vocales.

Dificultad normal:

Completar las 7 palabras del crucigrama que tiene hiato.

Nivel 3

Dificultad fácil:

Completar las 7 palabras agudas del crucigrama solo con las vocales.

Dificultad normal:

Completar las 7 palabras agudas del crucigrama.

Nivel 4

Dificultad fácil:

Completar las 7 palabras esdrújulas del crucigrama solo con las vocales.

Dificultad normal:



Completar las 7 palabras esdrújulas del crucigrama.

Nivel 5

Dificultad fácil:

Completar las 7 palabras llanas del crucigrama solo con las vocales.

Dificultad normal:

Completar las 7 palabras llanas del crucigrama.

Nivel 6

Dificultad fácil:

Completar las 8 palabras homófonas del crucigrama solo con las vocales.

Dificultad normal:

Completar las 8 palabras homófonas del crucigrama.

Nivel 7

Dificultad fácil:

Completar las 6 palabras parónimas del crucigrama solo con las vocales.

Dificultad normal:

Completar las 6 palabras parónimas del crucigrama.