



Universidad de las Ciencias Informáticas

FACULTAD 10

Título

**Módulo para la construcción de tableros de juegos didácticos de la
herramienta educativa SMProg para el Entorno Virtual de Aprendizaje
Moodle.**

**Trabajo de Diploma para optar por el Título de
Ingeniero en Ciencias Informáticas**

Autores

Ludiel Castro Barrios

Tutores

Lic. Dunia Suárez Ferreiro

Ing. Yenisel Valdés Hernández

Ciudad de La Habana

Diciembre de 2008

Declaración de Autoría

Declaración de autoría

Declaro que soy el único autor del trabajo titulado:

Módulo para la construcción de tableros de juegos didácticos de la herramienta educativa SMProg para el Entorno Virtual de Aprendizaje Moodle.

y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Ludiel Castro Barrios

Lic. Dunia Suárez Ferreiro

Ing. Yenisel Valdés Hernández

Datos de Contacto

Agradecimientos

A Dios en primer lugar, ya que sin su bendición nada es posible.

A mi mamá le agradezco por una vida de sacrificios, esfuerzo constante y por darme esa paz que sólo encuentro en sus palabras.

Al pancho y mi papa además de ser mis padres, por darme tanto apoyo durante mi formación profesional.

A mi hermana, al pichi y a la tuti, gracias por no escatimar esfuerzo alguno y por sacrificar parte de sus vidas para apoyarme en mi formación.

A mis tutoras, por su generosidad al darme la oportunidad de recurrir a sus capacidades y experiencias en un marco de confianza, afecto y amistad.

A Daynel, por brindarme su asesoramiento, apoyo y ayuda incondicional.

A mi novia por aguantarme todas mi malcriadeces en todo este año y estar pendiente de mi vida, también a su hermosa familia que a sabido ganarse un lugar en mi vida.

A mis abuelos Picoco, Armando y las dos mima, que seria una vida sin esos viejitos que desde pequeño están trabajando en nuestra formación.

A mis hermanos, a mis tias, tios, primos, toda la familia que ah confiado en mí.

A mis amigos tanto como los del barrio (al gordo, liber y Richard) como a los de la Universidad, a mi querido grupo que siempre me a ayudado en especial mi negrones y a Susi.

A Fidel Castro por permitirme formar parte de este sueño suyo que es la creación de la UCI.

Dedicatoria

Esta tesis va dedicada a las tres pétalos que conforman mi vida, tres mujeres que han estado en cada palabra y en cada línea de código, tres mujeres que por su magnitud aun no he sabido comprenderlas, ellas son mi Madre, mi Hermana y mi Novia.

Resumen

En la Universidad de las Ciencias Informáticas (UCI) se utiliza para la gestión de las asignaturas un Entorno Virtual de Aprendizaje (EVA) y se desarrollan aplicaciones, herramientas y módulos para apoyar y mejorar el proceso de enseñanza-aprendizaje (PEA) desde la plataforma de teleformación.

Entre los módulos en desarrollo se encuentra SMProg que, desde una óptica diferente, permite la interacción directa profesor-estudiante, mediante juegos didácticos, lo que propicia la motivación. Para lograr esta interacción es preciso que los profesores creen sus propios medios de enseñanza, en particular juegos didácticos de tablero, lo que puede ser posible mediante un módulo que permita diseñar tableros de juegos con las características necesarias.

Con la presente investigación se pretende desarrollar una herramienta, sobre software libre, que permita diseñar tableros para juegos didácticos para teleformación, como parte de un Proyecto de Innovación Pedagógica en la Universidad de las Ciencias Informáticas.

Introducción	13
Capítulo 1. Fundamentación teórica.....	17
1.1 ¿Qué es un Entorno Virtual de Aprendizaje?	19
1.1.1 Moodle.	21
1.2 Juegos Didácticos.	22
1.3 Módulos de Juegos Didácticos para Moodle.	24
1.4 ¿Qué es SMProg?	25
1.4.1 Módulo de Juegos didácticos con Tableros.	26
1.4. 1.1 Diseño de tableros.	27
1.5 Metodología de desarrollo de Software.....	28
1.5.1 Proceso Unificado de Rational (RUP).	29
1.6 Tecnologías actuales a considerar.	30
1.6.1 ¿Por qué se elige Java para el desarrollo de esta aplicación?	30
1.6.2 Herramienta de desarrollo.	32
1.6.3 Herramienta de modelado.	33
Capítulo 2. Características del sistema.	36
2.1 Modelo de Dominio.....	36
2.2 Levantamiento de requisitos.	37
2.2.1 Requerimientos funcionales	37
2.2.2 Requerimientos no funcionales	39
2.3 Modelo de Casos de Uso.	39
2.3.1 Casos de uso del sistema.....	41
2.3.2 Descripción de Casos de Uso	43
Capítulo 3. Análisis y diseño del sistema.	64
3.1 Características fundamentales del Análisis.	64

3.1.1 Diagrama de Clases del Análisis.....	65
3.2 Diseño.....	68
3.2.1 Diagrama interacción.....	68
3.2.2 Diagrama de Clases.....	70
3.2.3 Descripción de clases.....	73
3.2.4 Tratamiento de errores.....	93
3.2.5 Interfaz.....	93
Conclusiones.....	95
Recomendaciones.....	96
Referencias Bibliográficas.....	97
Bibliografía.....	98
Anexos.....	99

Figura 1: Diagrama del Modelo de Dominio	37
Figura 2: Diagrama de Caso de Uso del Sistema.....	40
Figura 3: Diagrama de Clases del Análisis del CU Gestionar Tablero.....	65
Figura 4: Diagrama de Clases del Análisis del CU Gestionar Casilla.	66
Figura 5: Diagrama de Clases del Análisis del CU Gestionar Ficha.....	66
Figura 6: Diagrama de Clases del Análisis del CU Gestionar Texto.	67
Figura 7: Diagrama de Clases del Análisis del CU Modificar Dado.....	67
Figura 8: Diagrama de Clases del Análisis del CU Gestionar Imágenes.....	68
Figura 9: Diagrama de Colaboración del CU Gestionar Tablero.....	69
Figura 10: Diagrama de Colaboración del CU Gestionar Casilla.	70
Figura 11: Diagrama de Clases.	72
Figura 12: Diagrama de Colaboración del CU Gestionar Ficha.	99
Figura 13: Diagrama de Colaboración del CU Gestionar Imágenes.....	100
Figura 14: Diagrama de Colaboración del CU Gestionar Dado.....	100
Figura 15: Diagrama de Colaboración del CU Gestionar Texto.	101

Índice de Tablas

Tabla 1: Actor del sistema	40
Tabla 2: Caso de Uso Gestionar Tablero	41
Tabla 3: Caso de Uso Insertar Casilla	41
Tabla 4: Caso de Uso Modificar Casilla	42
Tabla 5: Caso de Uso Modificar Dado	42
Tabla 6: Caso de Uso Gestionar Fichas	42
Tabla 7: Caso de Uso Gestionar Imágenes	43
Tabla 8: Caso de Uso Gestionar Texto	43
Tabla 9: Descripción del Caso de Uso Gestionar Tablero	44
Tabla 10: Descripción del Caso de Uso Insertar Casilla	47
Tabla 11: Descripción del Caso de Uso Modificar Casillas	54
Tabla 12: Descripción del Caso de Uso Modificar Datos	57
Tabla 13: Descripción del Caso de Uso Gestionar Fichas	58
Tabla 14: Descripción del Caso de Uso Gestionar Imagen	60
Tabla 15: Descripción del Caso de Uso Gestionar Texto	62
Tabla 16: Descripción de la clase JFrame	73
Tabla 17: Descripción de la clase Tablero	74
Tabla 18: Descripción de la clase Casilla	78
Tabla 19: Descripción de la clase Obstaculo	80
Tabla 20: Descripción de la clase Avanzar	81
Tabla 21: Descripción de la clase Didactica	82
Tabla 22: Descripción de la clase Texto	84
Tabla 23: Descripción de la clase TextoPanel	85

Índice de Tablas

Tabla 24: Descripción de la clase Seleccionado	86
Tabla 25: Descripción de la clase Ficha	87
Tabla 26: Descripción de la clase Dado	88
Tabla 27: Descripción de la clase Tabla.....	89
Tabla 28: Descripción de la clase Grafo_XML	91
Tabla 29: Descripción de la clase XML_Grafo	91

Introducción

En la actualidad las redes de computadoras y especialmente Internet, han dado lugar al surgimiento de espacios virtuales propicios para el intercambio, el debate y la discusión entre personas de diversas latitudes del planeta. Por otra parte, el mejoramiento experimentado de los software educativos y la búsqueda incesante por parte de los creadores para mejorar la enseñanza de los estudiantes, hacen de la computadora un importante medio de enseñanza, que a diferencia del resto permite una comunicación bidireccional con los profesores y estudiantes.

Universidades de varios países han recurrido a la utilización de las Tecnologías de la Información y de la Comunicación (TIC) con el fin de mejorar el proceso de enseñanza-aprendizaje, ejemplo de esto es la utilización de Entornos Virtuales de Aprendizaje (EVA) que no son más que espacios con accesos restringidos, concebidos y diseñados para que las personas que acceden a él, desarrollen procesos de incorporación de habilidades y conocimientos, mediante sistemas telemáticos permitiendo que los usuarios identificados (tutores, alumnos y administradores) puedan comunicarse entre sí en cualquier momento, enviar los trabajos y recibir los resultados de sus ejercicios (ConocimientosWeb.net). Ejemplos de EVAs utilizados en las Universidades del mundo son: Dokeos, ILIAS, Sakai, Claroline, ATutor, LRN, BlackBoard y Moodle, entre otros. De estos EVAs el más utilizado es Moodle porque se adapta mejor a los cursos basados en la interacción entre los participantes.

A partir del año 2005 en la Universidad de las Ciencias Informáticas (UCI) se comienza a utilizar la plataforma de entorno virtual de aprendizaje Moodle, desde ese entonces ha brindado una fuente de información e interacción, ya que el diseño y el desarrollo de Moodle está basado en la filosofía de que “la colaboración entre profesor-alumno transforma al alumno en protagonista del proceso de enseñanza-aprendizaje” (Pages, 2006). Muchos desarrolladores trabajan en la integración de módulos para Moodle con

el fin de ampliar las diferentes formas en que se puede manifestar la enseñanza, así como los medios que se utilizan para lograrlo, como los juegos didácticos.

El empleo de juegos didácticos, si se realizan de forma planificada, teniendo en cuenta las características de los alumnos y una adecuada orientación metodológica pueden ser un método efectivo dentro del proceso de enseñanza-aprendizaje, al incorporar elementos de motivación, competencia, espontaneidad, participación y emulación (Mondeja González, y otros, 2003).

En la universidad de las Ciencias Informáticas se desarrolla un proyecto de innovación pedagógica para implementar módulos de juegos didácticos para Moodle, en particular la herramienta SMProg, que está diseñada para utilizarse como actividad dentro de la plataforma de teleformación. Esta herramienta se encuentra en fase de desarrollo, en su primera versión solo permitirá los juegos didácticos de tablero, por ser juegos conocidos y de buena aceptación general. SMProg aún no cuenta con un módulo que les permita a los profesores diseñar los diferentes tipos de tableros de juegos didácticos, en dependencia de las características del jugador (estudiante) y las necesidades didácticas que requiera.

A partir de la situación problemática existente se plantea el siguiente **problema científico**:

¿Cómo permitir a los profesores de la Universidad de las Ciencias Informáticas (UCI) la creación de tableros de juegos didácticos para la herramienta SMProg?

Como **Objeto de estudio** tenemos las Herramientas para el Entorno Virtual de Aprendizaje (EVA) Moodle y como **Campo de acción**, Módulos de juegos didácticos para el Entorno Virtual de Aprendizaje (EVA) Moodle.

Se ha trazado el siguiente **Objetivo General** de la investigación: Diseñar e implementar una aplicación para la creación de tableros de juegos didácticos para la herramienta educativa SMProg.

Idea a defender: con la creación de un módulo para la construcción de tableros de juegos didácticos se garantizará que los profesores creen y personalicen los tableros de juegos didácticos en la herramienta SMProg.

Para el desarrollo de la investigación se proponen los siguientes **objetivos específicos**:

1. Realizar un estudio para fundamentar la investigación.
2. Diseñar una aplicación para la creación de tableros de juegos didácticos.
3. Implementar una aplicación para la creación de tableros de juegos didácticos.

Para darle respuesta a los objetivos anteriores es necesaria la realización de las siguientes **tareas investigativas**:

1. Realizar un estudio del estado del arte. Investigar las aplicaciones similares que existen a nivel mundial.
2. Diseñar la interfaz visual de la aplicación.
3. Diseñar las funcionalidades de la aplicación.
4. Implementar la aplicación con las características diseñadas.

Métodos teóricos utilizados

Durante el desarrollo de la investigación se emplea el método Histórico-Lógico para realizar un estudio acerca del desarrollo de software educativos que incluyen juegos didácticos y entornos de aprendizaje virtual, con el objetivo de lograr una correcta implementación de la herramienta.

Se utiliza **Análisis y Síntesis** para el análisis de las características de los diferentes tipos de juegos con tableros para lograr un correcto diseño de la aplicación.

Se utilizó también el método de **Modelación** para representar el objeto de estudio y la posible solución del problema científico. Con este método se realiza el diseño de la herramienta a implementar.

El presente documento se encuentra estructurado en introducción, tres capítulos, conclusiones, recomendaciones y anexos. En el **Capítulo 1** se presenta la fundamentación teórica acerca de los sistemas de aprendizajes didácticos y de las herramientas que permiten el diseño de mapas o tableros para juegos que existen en el mundo, haciendo un acercamiento al ámbito nacional y de la UCI en general y las posibles soluciones que se le dan a estos problemas. Se tratará la descripción de las características generales de la aplicación.

En el **Capítulo 2** se presenta el modelo de negocio y el levantamiento de requisitos, se definen los casos de uso, actores, trabajadores y se muestran los artefactos correspondientes.

En el **Capítulo 3** se presenta el análisis y diseño de la herramienta DesBoard (Designer Board) para su posterior implementación.

Capítulo 1. Fundamentación teórica.

La didáctica es el arte de enseñar y constituye la rama fundamental de la Pedagogía. Su objeto de estudio está bien delimitado: el Proceso de Enseñanza – Aprendizaje (PEA). El PEA está compuesto por dos procesos en sí complejos, la enseñanza y el aprendizaje. En este proceso el docente puede actuar como guía o facilitador del aprendizaje de los estudiantes a partir de los métodos de enseñanza que utilice, creando un proceso de interacción entre los conocimientos que trasmite y la forma en que los estudiantes se apropian del mismo (Suárez Ferreiro, 2009).

El PEA está compuesto por categorías didácticas que se clasifican en personales y no personales. Las personales son el profesor y el estudiante y las no personales son los objetivos, contenido, método, medios de enseñanza, evaluación y formas de organización. Las formas de organización reflejan las relaciones entre profesor y estudiantes en la dimensión espacial y temporal del proceso, que en la actualidad, con el perfeccionamiento de la tecnología educativa, a partir del incremento y auge de las TIC, y por ende el Software Educativo, el espacio y los recursos para el aprendizaje no están limitados.

En sus inicios los software educativos eran considerados programas participativos con ejercicios repetitivos que funcionaban en las PC, pero hoy en día con el desarrollo de las TIC y la evolución de este tipo de software, se ha demostrado que el software educativo es muy práctico y ayuda tanto a profesores como a estudiantes, pues estos programas se pueden diseñar de acuerdo a las necesidades docentes de cada cual.

Sánchez J., en su Libro “Construyendo y Aprendiendo con el Computador”, define el concepto genérico de Software Educativo como cualquier programa computacional

cuyas características estructurales y funcionales sirvan de apoyo al proceso de enseñar, aprender y administrar.

Con la evolución del software educativo se fueron definiendo algunas características que los mismos deben permitir:

- La interactividad con los estudiantes, retroalimentándolos y evaluando lo aprendido.
- Facilitar las representaciones animadas.
- Incidir en el desarrollo de las habilidades a través de la ejercitación.
- Permitir simular procesos complejos.
- Reducir el tiempo de que se dispone para impartir gran cantidad de conocimientos facilitando un trabajo diferenciado, introduciendo al estudiante en el trabajo con los medios computarizados.
- Facilitar el trabajo independiente y a la vez el tratamiento individual de las deficiencias de los estudiantes.

Sin embargo, la tecnología educativa ha evolucionado mucho a lo largo de la historia y es posible contar con nuevas formas de organización como el e-learning que constituye una modalidad de educación a distancia (forma de organización) con una base tecnológica fuerte destinada a sustentar el PEA en red, con potencialidades de comunicación y colaboración en espacios geográficos distantes. En la actualidad son muy utilizados los entornos de aprendizaje virtual destinados para este fin, lograr el e-learning.

1.1 ¿Qué es un Entorno Virtual de Aprendizaje?

Un entorno virtual de aprendizaje (EVA) consiste en el conjunto de las distintas funcionalidades asociadas a la actividad formativa. EL EVA permite la creación y mantenimiento de comunidades virtuales, proporcionando los servicios con los que cada comunidad se identifica, y que garantizan la integración, enriquecimiento y la fiabilidad de sus usuarios (ConocimientosWeb.net).

Un entorno Virtual de Aprendizaje no es más que un sitio donde se puede encontrar documentación, cursos online y tutoriales entre otros documentos organizados y clasificados, orientados a la enseñanza. Cuenta con un servicio de autenticación y permite el intercambio de informaciones e ideas, facilitando la interacción entre distintos usuarios.

Para que un sitio cumpla con las características de ser un EVA debe cumplir con una serie de requisitos que lo distinguen los cuales se muestran a continuación (ConocimientosWeb.net).

1. Permitir búsquedas rápidas de contenidos.
2. Contar con una base de datos de preguntas y respuestas más frecuentes.
3. Foro de noticias de hechos más recientes.
4. Preguntas y pruebas de autoevaluación.
5. Clasificación de exámenes de forma inmediata y marcando los errores cometidos en el examen.
6. Referencias y enlaces a otras páginas Web recomendadas.
7. Glosario terminológico.
8. Publicaciones de los eventos más significativos.

9. Actualización de la documentación.

Los EVAs tienen una gran importancia ya que facilitan el trabajo tanto a estudiantes como a profesores, permiten el intercambio de ideas y conocimientos entre los usuarios. No solo sirven para impartir o recibir un contenido determinado, sino que es posible encontrar muchas fuentes de información que facilitan una preparación acorde a los deseos de cada estudiante.

Como se menciona con anterioridad entre los EVA más utilizados actualmente se encuentran:

- Dokeos
- ILIAS
- Sakai
- Claroline
- ATutor
- LRN
- BlackBoard
- Moodle

Los cuales presentan características similares por constituir entornos de aprendizaje virtual, sin embargo SMProg se vinculará con Moodle por ser el CMS que se emplea en la Universidad.

1.1.1 Moodle.

Moodle, por sus siglas en inglés significa *Modular Object-Oriented Dynamic Learning Environment* (Entorno de aprendizaje dinámico modular orientado a objetos), es un sistema de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea. Este tipo de plataformas tecnológicas también se conocen como LMS (Learning Management System)¹.

Moodle es utilizado en todo el mundo en los centros de enseñanza es una herramienta muy útil para estos propósitos ya que soporta disímiles contenidos como son multimedia, apuntes, videos, imágenes ext. Permite fomentar el auto-aprendizaje y el aprendizaje colaborativo y realizar exámenes online y evaluar las tareas de los estudiantes.

Moodle es una herramienta potente que facilita el trabajo para los cursos en línea y para la preparación personal. Es multiplataforma y permite poder incluir o enlazar (link), al igual que múltiples blogs, web-quest, imágenes, videos o documentos, que hacen mucho más rico, variado y dinámico el trabajo.

Entre las principales ventajas que nos ofrece Moodle se encuentra la posibilidad que brinda de que los alumnos participen en la creación de glosarios, en todas las lecciones se generan enlaces a las palabras incluidas en estos, las universidades pueden montar su propio Moodle y adecuarlo de acuerdo a las principales necesidades que estas tengan (Moodle, 2002).

En la UCI el EVA está basado en el software libre Moodle constituyendo un espacio de apoyo al proceso de formación de la Carrera de Ingeniería en Ciencias Informáticas y a

¹ <http://moodle.org/>

los programas de postgrado que tributan a la formación del claustro de profesores de la universidad. Constituye además, un espacio virtual donde los profesores pueden implementar estrategias de enseñanza-aprendizaje complementarias a las clases presenciales, así como diseñar cursos semipresenciales o totalmente a distancia. Los estudiantes disponen de un poderoso medio en el cual pueden obtener, utilizar o compartir materiales didácticos (Teleformación, 2005).

1.2 Juegos Didácticos.

A medida que aumenta la edad de los estudiantes y el nivel de enseñanza que cursan, en las instituciones docentes disminuye la utilización de juegos didácticos, por lo que su aplicación decrece de la enseñanza primaria hacia la universitaria.

El juego es una actividad de aprendizaje; si se dirige y orienta pedagógicamente contribuye a la activación del pensamiento y permite desarrollar las capacidades intelectuales de los alumnos (Ferreiro Cordero, 2008).

Los juegos didácticos se dividen en las siguientes categorías:

- Juegos Creativos.
- Juegos Profesionales.
- Juegos Didácticos.

Juegos Creativos: Permiten desarrollar en los estudiantes la creatividad, estimulan la imaginación y la producción de nuevas ideas.

Juegos Profesionales: Estimulan la rapidez y la motivación para entender y aprender. Se centran en la materia que se va a impartir.

Juegos Didácticos: Resulta un método muy eficaz, sobre todo para la enseñanza problémica. Existen diversas variantes de cómo emplear estos:

De tipo Competitivo: Encuentros de conocimiento, olimpiadas.

De tipo Profesional: Análisis de situaciones concretas, análisis de casos, interpretaciones de papeles, simulación.

De acuerdo con Armando Tesla en su libro: "Aprendizaje sobre juegos", los principios que más caracterizan a los juegos didácticos son los siguientes:

- De participación activa.
- De dinamismo.
- De entrenamiento.
- De interpretación de roles.
- De carácter problémica.
- De obtención de resultados concretos.
- De competencia.

Los juegos didácticos pueden contribuir, en cualquier enseñanza, a elevar la calidad del PEA, siempre que se tengan en cuenta las características que deben cumplir para su efectividad. Entre los resultados que se han obtenido históricamente con su utilización destacan:

- Elevar el estudio individual de los estudiantes.
- Mejorar los resultados docentes.
- Lograr una interacción cooperativa y competitiva entre un colectivo de estudiantes.

- Motivar al estudiantado.
- Encontrar nuevas formas de evaluación.

1.3 Módulos de Juegos Didácticos para Moodle.

En la actualidad se han implementado varias herramientas de juegos didácticos que ayudan a la superación de los estudiantes y sirven de apoyo al PEA, diseñadas con diferentes objetivos, algunas de apoyo para asignaturas determinadas. Ninguna de estas herramientas ofrece la posibilidad de integrarse como un módulo a Moodle, excepto Jclíc que permite el diseño y utilización de algunos tipos de juegos didácticos (crucigramas, sopa de palabras, asociaciones de palabras, etc.) para la enseñanza primaria y secundaria.

Entre sus principales características se encuentran las siguientes:

- Esta implementado sobre la plataforma Java, con un formato de almacenamiento XML.
- Sus componentes son
 - Jclíc- applet: es un applet que permite que se integre a una aplicación Web, en este caso a Moodle.
 - Jclíc-reports: es un módulo de recogida de datos sobre las actividades que realizan los estudiantes.
 - Jclíc-author: permite crear y modificar proyectos Jclíc, en un entorno visual muy intuitivo e inmediato.

- Tiene arquitectura abierta que permite ampliar o adaptar sus funcionalidades.
- Es un proyecto de software libre.

Sin embargo, para los requerimientos del nivel universitario esta herramienta no es suficiente, pues está definida con un alcance menor, no obstante es bueno resaltar que algunas de sus características comunes con respecto a SMProg, se tienen en cuenta para el desarrollo más rápido de esta nueva herramienta educativa.

1.4 ¿Qué es SMProg?

La herramienta educativa SMProg (Software – Motivación - Programación) se desarrollará sobre software libre, en particular sobre el lenguaje de programación Java, y permitirá al profesor definir el sistema de preguntas que desea se muestren a los estudiantes que se encuentren "jugando". Las preguntas podrán agruparse por temas y saldrán en un inicio de forma aleatoria. De acuerdo a los temas en que más se confunda el estudiante, el propio software mostrará un mayor número de preguntas de ese tipo.

Para el diseño e implementación se van a utilizar algoritmos de Inteligencia Artificial y Bases de Datos, así como técnicas de programación orientada a objetos. Se pretende desarrollar a partir de dos enfoques principales, uno orientado a desktop (escritorio), que permita interactuar desde una estación de trabajo particular no necesariamente conectado en red y otro orientado a módulos para incluir dentro de la plataforma de teleformación ya existente en la universidad, permitiendo así la interacción concurrente de los usuarios y el aprendizaje no presencial a partir de la permanente disponibilidad de las actividades en la plataforma. En un futuro, pueden desarrollarse módulos con las características que requieran la asignatura, el profesor o las particularidades del estudiantado.

En la primera versión de la herramienta educativa se incluyen en el diseño e implementación solo Juegos Didácticos con tableros debido a que son bien conocidos y tienen buena aceptación general. En esta primera versión del software se presentan tres módulos diferentes estrechamente vinculados entre sí y con similar ambiente que son los siguientes:

- **Módulo Gestión de Asignaturas:** para la conexión con Moodle en caso de estar vinculado a la plataforma de teleformación y para la definición de temas, sistemas de preguntas y de evaluación de las asignaturas en caso de no estar vinculado a Moodle.
- **Módulo Juegos didáctico con Tableros (lógica de negocio):** Desde este módulo se integran los restantes, es por esto que se nombra módulo principal. Desde el mismo es posible gestionar la asignatura, diseñar los tableros de juegos o iniciar el juego, con sus características propias, además de administrar la aplicación (roles - usuarios), manteniendo la comunicación con Moodle a través de applets y de la base de datos.

1.4.1 Módulo de Juegos didácticos con Tableros.

Este módulo permite que los estudiantes de una forma sencilla y recreativa amplíen sus conocimientos y desarrollen habilidades educativas mediante una interfaz que permite, de forma automática, mostrar las preguntas en orden creciente de complejidad.

Juegos Didácticos con Tableros consta de dos subsistemas:

1. Diseño de Tablero.
2. Implementación de juegos de tableros

El **Diseño de Tablero** permite al profesor diseñar el tablero teniendo en cuenta las características de sus estudiantes, ya sea motivación por la asignatura o dificultades en la misma.

En esta interfaz visual se puede elegir el tipo de juego de tablero deseado para posteriormente ubicar obstáculos dentro de las casillas, así como definir casillas de tipo didáctico.

Una vez diseñada la interfaz visual del juego, es necesario poder interactuar con la misma, aquí es donde entra la Implementación de juegos de tableros que permite a los estudiantes jugar y aprender.

Este subsistema es capaz de guiar a los estudiantes por un sistema de preguntas que al inicio es a través de una pregunta por cada tema, analizando cuál es el primer tema de la asignatura que el estudiante debe ejercitar, dado sus malos resultados en el mismo, para mostrar un mayor número de preguntas de ese tipo.

1.4. 1.1 Diseño de tableros.

Existen muchas herramientas de diseño Gráfico que se pueden utilizar en el diseño de tableros, desde un programa sencillo como el Paint hasta uno más profesional como el PhotoShop, Gimp, unas privativas y otras libres.

Estas herramientas permiten obtener un buen dibujo del tablero con una buena calidad de la imagen, que sería óptimo para despertar el interés de los jugadores. Estas ventajas que brindan los softwares de diseño son fundamentales en la creación de los tableros de juegos didácticos.

En la realización de la actividad (Diseñar Tablero) se necesita obtener algunos parámetros de esa "Imagen" como es una estructura no lineal (grafo) de casillas, que brinda información clasificada a la hora de la implementación del módulo como son:

- Posición X, Y en la que se encuentra la casilla y otros componentes.
- Selección del tipo de tablero.
- Número máximo de intentos a que tiene posibilidad el estudiante de responder correctamente las preguntas.
- Definición de las propiedades de las diferentes especializaciones de las casillas.

Esta información no se puede obtener de una herramienta de diseño convencional, aunque si es bueno mencionar que puede servir como apoyo a la hora de confeccionar los tableros.

1.5 Metodología de desarrollo de Software.

Las metodologías de desarrollo de software son un conjunto de procedimientos, métodos y técnicas que ayudan a organizar el trabajo y la documentación con el objetivo de facilitar el desarrollo del producto. Es decir, son una guía de pasos a seguir, donde además indican qué persona y qué papel debe tener la misma para realizar una actividad específica en el desarrollo del software. Además detallan la información que se debe producir después de terminada una actividad.

Esta tecnología se divide en dos grandes grupos:

1. Metodologías Ligeras. (XP (Programación extrema) y (Crystal Light Methods)
2. Metodologías Pesadas. (SW-CMM (SoftWare Capability Maturity Model), RUP (Proceso Unificado de Rational).

Las ligeras se basan en conseguir su objetivo por medio del orden y la documentación, mientras que las pesadas se basan en la comunicación directa e inmediata entre aquellos que intervienen en el proceso.

En este caso en particular se utiliza la metodología RUP para el desarrollo de la aplicación, por todas las ventajas de organización que brinda, por dividir el trabajo en roles, por ser adaptable a cualquier tipo de proyecto, ya sean grandes o sencillos y por ser una metodología orientada a objetos. Además de que la aplicación forma parte de un proyecto que se desarrolla con esta metodología.

A continuación se analizan algunas de las características de RUP:

1.5.1 Proceso Unificado de Rational (RUP).

Es un proceso de desarrollo de software que junto con el Lenguaje Unificado de Modelado (UML), conforma la metodología más utilizada en el mundo tanto para el desarrollo de pequeños proyectos como para sistemas de software complejos pues está pensada para adaptarse a diferentes áreas de aplicaciones y diferentes organizaciones.

RUP se divide en cuatro fases (Inicio, Elaboración, Construcción, Transición) y presenta nueve flujos de trabajo de los cuales seis son de ingeniería de software y tres de apoyo, estos son:

1. Modelo del negocio.

2. Requerimientos.
3. Análisis y Diseño.
4. Implementación.
5. Prueba.
6. Instalación o distribución.
7. Configuración y administración del cambio.
8. Administración de proyectos.
9. Ambiente.

Como una de sus principales características está el hecho de que en cada ciclo de iteración se hace exigente el uso de artefactos.

1.6 Tecnologías actuales a considerar.

Para el desarrollo de esta aplicación se utiliza el lenguaje de programación Java que no es más que un lenguaje orientado a objetos desarrollado por Sun Microsystems. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

1.6.1 ¿Por qué se elige Java para el desarrollo de esta aplicación?

Se escogió este lenguaje por dos razones fundamentales:

1. Portabilidad de su plataforma.
2. Se puede incrustar en otro programa.

Portabilidad de su plataforma:

El compilador del lenguaje Java genera un código binario especial (el Java bytecode) que es interpretado por una Máquina Virtual (JVM) sus siglas en inglés significan Java Virtual Machine.

La Máquina virtual de Java (JVM) es un programa nativo, es decir, es un ejecutable en una plataforma específica, capaz de comprender tanto el bytecode como el sistema en el que se va a ejecutar la aplicación. Así, cuando se escribe una aplicación Java, se hace pensando para que se ejecute en una máquina virtual de Java, siendo ésta la que en última instancia convierte de código bytecode a código nativo del dispositivo final.

La gran ventaja de la máquina virtual de Java es su portabilidad, de manera que su creación para diferentes arquitecturas logra que un programa escrito en Windows pueda ser interpretado en un entorno libre, disponiendo solamente de la máquina virtual para dichos entornos. De esto se desprende el famoso axioma que sigue a Java, "escríbelo una vez, ejecútalo en cualquier parte".

Se puede incrustar en otro programa:

Una de las funcionalidades que potencian el uso del lenguaje de programación Java es la incorporación de los applets.

Un applet es un componente de Java que se ejecuta en el contexto de otro programa, por ejemplo un navegador web. El applet, a diferencia de otros programas, no puede ejecutarse de manera independiente, debe ejecutarse siempre en un contenedor. Esta es la principal razón por la que se ha escogido este lenguaje ya que puede permitir que la aplicación desarrollada pueda ser ejecutada por cualquier usuario con tan solo cargar la página Web en su navegador.²

² <http://sun.com>

1.6.2 Herramienta de desarrollo.

Actualmente existe gran variedad de IDEs (*Interface Development Enviroment*) para el desarrollo de aplicaciones Java, como son *WebSphere Studio Application Developer*, *Rational Application Developer*, *JBuilder* entre otros, aunque solo prestaremos atención a los IDEs *Open Source* (código abierto), en especial el NetBeans que ha sido escogido para el desarrollo de la aplicación.

NetBeans es un proyecto exitoso con una gran comunidad en constante crecimiento lo que facilita el trabajo con el mismo, fue fundado como proyecto de código abierto por Sun Microsystems quien continúa siendo su patrocinador principal. Es un entorno de desarrollo con el que se puede escribir, compilar, depurar y ejecutar programas. Está escrito en el propio Java pero se puede utilizar para cualquier otro lenguaje de programación. Cuenta con un número importante de módulos para extenderlo, es un producto libre y gratuito sin restricciones de uso. La Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear grandes aplicaciones de escritorio, ofrece soporte para los plug-in AWT/SWING entre otros, aunque solo estos serán utilizados en la creación de la aplicación. El NetBeans viene integrado con todas las herramientas necesarias para el desarrollo Web o Desktop a diferencia del Eclipse, que es necesario ir instalando plug-ins según sea necesario.

En este trabajo no se pretende demostrar que el NetBeans³ es el mejor IDE Open Source para el desarrollo en Java solo que ofrece la ventajas necesarias para lograr una conformidad en la realización de la aplicación.

³ <http://netbeans.org>

1.6.3 Herramienta de modelado.

El Visual Paradigm⁴ utiliza “UML” como *lenguaje* de modelado, soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite crear todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

Lista de características:

- Soporte de UML versión 2.1
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento
- Modelado colaborativo con CVS y Subversión (nueva característica)
- Interoperabilidad con modelos UML2 (meta-modelos UML 2.x para plataforma Eclipse) a través de XMI (nueva característica).
- Ingeniería de ida y vuelta
- Ingeniería inversa - Código a modelo, código a diagrama
- Ingeniería inversa Java, C++, Esquemas XML, XML,.NET exe/dll, CORBA IDL
Generación de código - Modelo a código, diagrama a código
- Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso
- Diagramas EJB - Visualización de sistemas EJB.
- Generación de código y despliegue de EJB's - Generación de beans para el desarrollo y despliegue de aplicaciones.

⁴ <http://www.visual-paradigm.com/>

- Diagramas de flujo de datos
- Soporte ORM - Generación de objetos Java desde la base de datos
- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación
- Generador de informes para generación de documentación
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML
- Importación y exportación de ficheros XML
- Integración con Visio - Dibujo de diagramas UML con plantillas (esténcils) de MS Visio
- Editor de figuras.

Además permite la integración con las siguientes herramientas Java:

- Eclipse/IBM WebSphere.JBuilder.
- NetBeans IDE.
- Oracle Jdeveloper.
- Sun ONE.

En especial con el NetBeans que es el IDE a utilizar en el desarrollo de la aplicación, aportando facilidad y comodidad en el generador de código o de diagramas.

En la realización de esta etapa del trabajo han sido abordados los principales conceptos que de una forma u otra se relacionan con el trabajo como son las definiciones de:

- Qué es Moodle y sus principales ventajas y desventajas.

- Qué es un (EVA) y su impacto en el mundo.
- La importancia de los juegos didácticos.
- Qué es SMProg.

Partiendo de estos análisis y estudiando la situación problemática planteada se propone realizar un módulo que permita diseñar juegos didácticos. También se hizo un estudio de las principales tecnologías, metodologías y herramientas de diseño a considerar para el desarrollo de la aplicación.

Capítulo 2. Características del sistema.

En este capítulo se presentan las características del sistema que se propone para dar solución al problema existente y se describen elementos del proceso de creación de tableros de juegos didácticos. Se definen conceptos y se identifican entidades que se relacionan en un modelo de dominio, el cual permite determinar los requisitos necesarios que debe cumplir la aplicación para cubrir las necesidades que lo originan.

2.1 Modelo de Dominio.

Debido a que en la metodología del proceso de desarrollo del Software no está bien definido el negocio en la fase de inicio, es necesario identificar todos los conceptos que se utilizarán en el diagrama mediante un **glosario de términos**:

El **usuario** es aquella persona que posee los privilegios de acceder a la aplicación.

La **aplicación** es la herramienta de diseño capacitada para diseñar y configurar los tableros de juegos didácticos.

El **tablero** es un fichero que contiene la información necesaria para ser utilizada en el desarrollo del juego didáctico seleccionado.

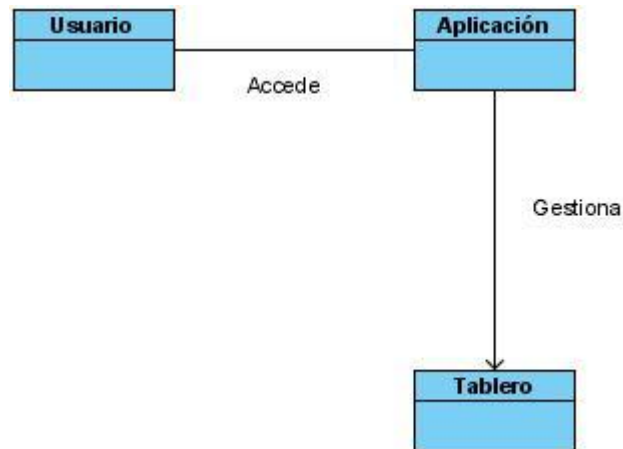


Figura 1: Diagrama del Modelo de Dominio

En la figura No.1 se muestra la relación entre las entidades definidas por el Modelo de Dominio, se puede observar como los usuarios acceden a la aplicación y obtienen como resultado final un fichero, el cual puede ser creado por el usuario o modificado a partir de uno ya existente.

2.2 Levantamiento de requisitos.

En el levantamiento de requisitos se define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

Los requisitos pueden ser de dos tipos, funcionales y no funcionales:

2.2.1 Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Los requisitos funcionales que identifican este sistema son:

R1 El sistema debe gestionar la utilización de ficheros.

R1.1 Permitir crear el tipo de tablero seleccionado.

R1.2 Salvar fichero con formato específico.

R1.3 Actualizar fichero.

R1.4 Abrir un juego de un fichero previamente guardado.

R2 Gestionar el trabajo con las casillas.

R2.1 Ubicar las diferentes casillas dentro del tablero.

R2.2 Modificar las casillas.

R2.3 Eliminar las casillas del tablero.

R3 Modificar dado.

R4 Gestionar fichas.

R4.1 Crea las fichas de los jugadores.

R4.2 Modificar las fichas.

R4.3 Eliminar las fichas de los jugadores.

R5 Oficiar imágenes dentro del tablero.

R5.1 Insertar imágenes dentro del tablero.

R5.2 Eliminar imágenes del tablero.

R6 Gestionar texto.

R6.1 Insertar texto en el tablero.

R6.2 Modificar el texto seleccionado.

R6.3 Eliminar el texto del tablero.

2.2.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que hacen al producto atractivo, usable, rápido o confiable. Los requisitos no funcionales que identifican este sistema son:

Hardware: Se puede utilizar una PC Pentium III o superior de 128 Mb RAM como mínimo para el buen funcionamiento del programa.

Restricciones de la implementación: Se programará en lenguaje Java, en la herramienta NetBeans.

Portabilidad: El sistema deberá ser multiplataforma, para esto se utilizará *Java Virtual Machine* (JVM) 6.0 o superior.

2.3 Modelo de Casos de Uso.

El profesor es el único actor que interactúa con la aplicación, iniciando todos los casos de uso del sistema. En la figura No. 2 se observan los casos definidos para representar el flujo de los eventos que permiten obtener un resultado de valor para los profesores durante el diseño del juego.

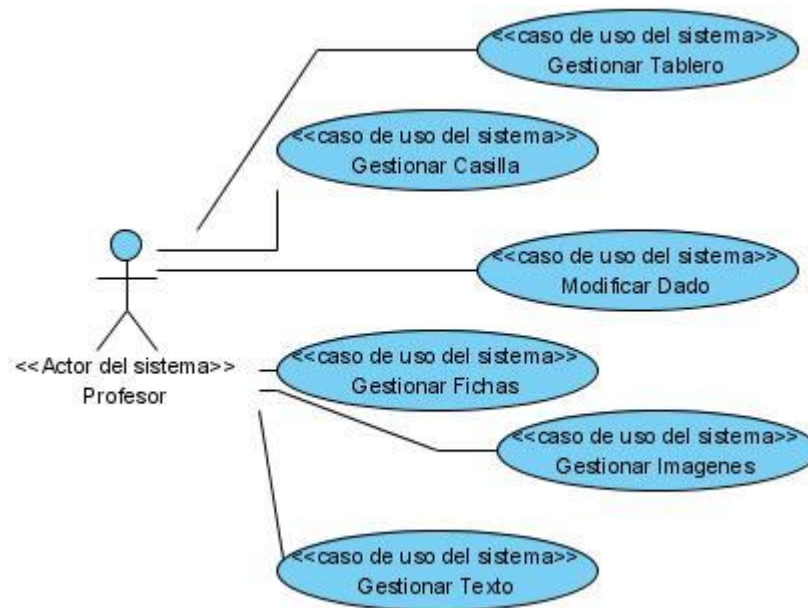


Figura 2: Diagrama de Caso de Uso del Sistema.

2. Actor presente en el sistema:

Tabla 1: Actor del sistema

Actor	Descripción
Profesor	Es el encargado de realizar el diseño de los juegos, así como de gestionar todo lo referente a las asignaturas y definir la forma en que los estudiantes van a jugar.

2.3.1 Casos de uso del sistema.

Tabla 2: Caso de Uso Gestionar Tablero

CU-1	Gestionar Tablero
Actor	Profesor
Descripción	El profesor selecciona el tipo de tablero deseado, fijando el número de intentos permitidos por estudiante en la resolución de preguntas. El tablero puede ser salvado o actualizado una vez creado o abierto previamente.
Referencia	R1

Tabla 3: Caso de Uso Insertar Casilla

CU-2	Insertar Casilla
Actor	Profesor
Descripción	El profesor elige el tipo de casilla deseado, ya sea casilla de avanzar, casilla didáctica o casilla de obstáculo.
Referencia	R2

Tabla 4: Caso de Uso Modificar Casilla

CU-3	Modificar Casilla
Actor	Profesor
Descripción	El profesor elige la casilla que desea modificar, estas pueden ser eliminadas, transformadas o duplicadas
Referencia	R2

Tabla 5: Caso de Uso Modificar Dado

CU-4	Modificar Dado
Actor	Profesor
Descripción	El dado se encuentra previamente seleccionado en dependencia del tipo de tablero, este caso de uso solo permite hacerle modificaciones al dado.
Referencia	R3

Tabla 6: Caso de Uso Gestionar Fichas

CU-5	Gestionar Fichas
Actor	Profesor
Descripción	El profesor selecciona la ficha deseada, el sistema se debe encargar de que no seleccione

	una ficha en uso. Las fichas pueden ser modificadas o eliminadas una vez insertadas.
Referencia	R4

Tabla 7: Caso de Uso Gestionar Imágenes

CU-6	Gestionar Imágenes
Actor	Profesor
Descripción	El profesor inserta una imagen existente, estas imágenes pueden ser eliminadas en caso de no ser requeridas en el diseño de tableros de juegos didácticos.
Referencia	R5

Tabla 8: Caso de Uso Gestionar Texto

CU-7	Gestionar Texto
Actor	Profesor
Descripción	El profesor introduce un texto en el tablero. Los textos pueden ser modificados o eliminados una vez insertados.
Referencia	R6

2.3.2 Descripción de Casos de Uso

Capítulo 3. Análisis y diseño del sistema.

Describe cómo se lleva a cabo y se ejecuta un caso de uso determinado, en términos de las clases del análisis y de sus objetos en interacción.

Tabla 9: Descripción del Caso de Uso Gestionar Tablero

CU-1	Gestionar Tablero	
Actores:	Profesor	
Propósito:	Crear, guardar y abrir fichero de tipo tablero	
<p><u>Resumen</u>: Se crea un tablero, seleccionando el tipo de tablero y fijando el número de intentos permitidos por estudiantes en la resolución de preguntas. El tablero puede ser guardado una vez creado o abierto previamente.</p>		
Referencias:	R1	
Precondiciones:	No debe existir ningún fichero (tablero) abierto en la aplicación.	
<u>Acción del actor</u>		<u>Respuesta del sistema</u>
1	El profesor selecciona el menú archivo	1.1 El sistema muestra varias opciones: <ul style="list-style-type: none"> • Nuevo tablero. • Abrir tablero
2	El profesor selecciona la opción deseada.	3 Si el profesor selecciona la opción Nuevo tablero, ver sección Nuevo tablero 4 Si el profesor selecciona la

		opción Abrir tablero, ver sección Abrir tablero
Sección Nuevo tablero		
3	El profesor selecciona la opción Nuevo tablero	<p>3.0.1 El sistema muestra los diferentes tipos de tableros que puede diseñar.</p> <ul style="list-style-type: none"> • Avanzar hasta la Meta. • Sube y Baja. • Bingo
3.1	El profesor selecciona el tipo de tablero que desea diseñar y escoge al número máximo de intentos de resolución de preguntas de similar complejidad	<p>3.1.1 El sistema crea un nuevo tablero y almacena el tipo de tablero y el número de intentos máximo prefijado por el profesor.</p> <p>3.1.2 El sistema muestra las diferentes opciones de casillas, textos e imágenes, entre otros a ubicar en el tablero.</p> <p>5 El sistema activa la opción salvar como, ver sección Salvar como.</p> <p>6) El sistema activa la opción salvar, ver sección Salvar.</p>
Sección Abrir tablero		
4	El profesor selecciona la opción Abrir	4.0.1 El sistema brinda la

tablero	opción de escoger la ruta en la que se encuentra el fichero deseado.
4.1 El profesor selecciona el fichero.	<p>4.1.1 El sistema abre el fichero y muestra las diferentes opciones de casillas e imágenes, entre otros a ubicar en el tablero.</p> <p>4.1.2 Se retorna al paso.</p> <p>4.1.3 3.1.2 del curso normal de los eventos.</p>
Sección Salvar como	
5 El profesor selecciona la opción Guardar como	5.0.1 El sistema brinda la opción de escoger el directorio y de definir el nombre del fichero.
5.1 El profesor escoge la dirección	5.1.1 El sistema guarda el fichero en la ruta especificada.
Sección Salvar	
6 El profesor selecciona la opción Guardar	6.0.1 El sistema actualiza el fichero en el que se encuentra trabajando.
Curso alternativo de los eventos.	

5.1 El profesor escoge la dirección	5.1.0 El sistema muestra un mensaje de error y realiza la petición de la introducción de un nombre para el fichero. 5.1.1 Se retorna al paso 5.0.1 del curso normal de los eventos.
6 El profesor selecciona la opción Guardar	6.0.1 Se retorna al paso 5.0.1 del curso normal de los eventos.
Pos condiciones:	El juego diseñado es almacenado en un fichero con formato específico.

Tabla 10: Descripción del Caso de Uso Insertar Casilla

CU-2	Insertar Casilla
Actores:	Profesor
Propósito:	Crear, personalizar y diseñar.
<p><u>Resumen</u>: Se escoge una casilla previamente diseñada o se crea una nueva, definiendo las propiedades de ésta que pueden ser casillas de obstáculo, avanzar o una casilla didáctica.</p>	
Referencias:	R2
Precondiciones:	Debe estar creado o abierto un fichero (tablero) en la aplicación.

Capítulo 3. Análisis y diseño del sistema.

<u>Acción del actor</u>	<u>Respuesta del sistema</u>
1 El profesor decide la casilla que desea insertar.	1.1 El sistema muestra varias opciones: 1.1.1 Un menú clic primario. 1.1.2 Un menú con clic secundario.
2 El profesor selecciona la opción que desea.	3 Si el profesor selecciona la casilla de Avanzar con el menú clic primario, ver sección Casilla de Avanzar con clic primario. 4 Si el profesor selecciona la casilla de Avanzar con el menú clic secundario, ver sección Casilla de Avanzar con clic secundario. 5 Si el profesor selecciona la casilla de Obstáculo con el menú clic primario, ver sección Casilla de Obstáculo con clic primario. 6 Si el profesor selecciona la casilla de Obstáculo con el menú clic secundario, ver sección Casilla de Obstáculo con clic

		7	secundario. Si el profesor selecciona la casilla Didáctica con el menú clic primario, ver sección Casilla Didáctica con clic primario.
		8	Si el profesor selecciona la casilla Didáctica con el menú clic Secundario, ver sección Casilla Didáctica con clic Secundario.
Sección Casilla de Avanzar con clic primario.			
3	El profesor selecciona la opción Casillas de Avanzar con clic primario.	3.0.1	El sistema inserta la casilla en el lugar del tablero donde se de un clic.
Sección Casilla de Avanzar con clic secundario.			
4	El profesor selecciona la opción Casilla de Avanzar con clic secundario	4.0.1	El sistema despliega una lista de opciones. <ul style="list-style-type: none"> • Casilla (Diseñar Casilla o Casilla predeterminada) • Funcionalidad de la casilla (Salida, Meta, Casilla, Unión y Bifurcación)
4.1	El profesor escoge la opción de casillas que desea.	9	Si el profesor escoge la opción Diseñar Casilla ver sección Diseñar Casilla.

	10	Si el profesor escoge la opción Casilla Prediseñada ver sección Casilla Prediseñada.
4.2 El profesor escoge la funcionalidad de la casilla.	4.2.1	Sigue el curso del paso (3.0.1).
Sección Casilla de Obstáculo con clic primario.		
5 El profesor escoge la opción Casilla de Obstáculo con clic primario.	5.0.1	Al dar clic en el lugar donde se desea se inserte la casilla el sistema muestra una nueva ventana donde se puede seleccionar el tipo de obstáculo y en caso que este sea (avanzar, retroceder, esperar ronda o esperar dados) te permite seleccionar el número de casillas que debes saltar o el número de turnos que debes esperar.
5.1 El profesor escoge la opción que desee.	5.1.1	El sistema inserta la Casilla.
Sección Casilla de Obstáculo con clic secundario		
6 El profesor selecciona la opción Casilla de Obstáculo con clic secundario	6.0.1	El sistema despliega una lista de opciones.

	<ul style="list-style-type: none"> • Casilla (Diseñar Casilla o Casilla predeterminada) • Funcionalidad de la casilla (Casilla, Unión y Bifurcación)
6.1 El profesor escoge la opción de casillas que desea.	<p>9 Si el profesor escoge la opción Diseñar Casilla ver sección Diseñar Casilla.</p> <p>10 Si el profesor escoge la opción Casilla Prediseñada ver sección Casilla Prediseñada.</p>
6.2 El profesor escoge la funcionalidad de la casilla.	6.2.1 El sistema sigue curso del paso (5.1.1).
Sección Casilla Didáctica con clic Primario.	
7 El profesor escoge la opción Casilla Didáctica con clic primario.	7.0.1 Al dar clic en el lugar donde se desea se inserte la casilla el sistema muestra una nueva ventana donde se puede seleccionar el tipo de pregunta que pueden ser: (Ámbito Nacional e Internacional, Docente o Materias Regresivas).
7.1 El profesor selecciona el tipo de pregunta que desea.	7.1.1 El sistema inserta la casilla.

Sección Casilla Didáctica con clic secundario.	
8 El profesor selecciona la opción Casilla Didáctica con clic secundario	8.0.1 El sistema despliega una lista de opciones. <ul style="list-style-type: none"> • Casilla (Diseñar Casilla o Casilla predeterminada) • Funcionalidad de la casilla (Casilla, Unión y Bifurcación).
8.1 El profesor escoge la opción de casillas que desea.	9 Si el profesor escoge la opción Diseñar Casilla ver sección Diseñar Casilla. 10 Si el profesor escoge la opción Casilla Prediseñada ver sección Casilla Prediseñada.
8.2 El profesor escoge la funcionalidad que desea.	8.2.1 El sistema sigue el curso del paso (7.1.1).
Sección Diseñar Casilla	
9 El profesor escoge la opción Diseñar Casilla.	1.6.1 El sistema muestra una nueva ventana donde aparece un diseño estándar de la casilla y le ofrece las opciones de “Siguiete” y “Cancelar”.
1.7 El profesor escoge la casilla y presiona el botón “Siguiete”.	1.7.1 El sistema muestra nuevas opciones donde se definen

	las dimensiones, color y contexto, y ofrece las opciones “Anterior”, “Aceptar” y “Cancelar”.
1.8 El profesor elige la opción “Aceptar”.	1.8.1 El sistema sigue el curso de los pasos (3.0.1), (5.1.1) o (7.1.1) si la casilla seleccionada para diseñar es Avanzar, Obstáculo o Didáctica respectivamente.
1.9 El profesor elige la opción “Anterior”.	1.9.1 El sistema vuelve al paso (9.0.1).
1.10 El profesor elige la opción “Cancelar”.	1.10.1 El sistema anula la nueva casilla.
Sección Casilla Prediseñada.	
10 El profesor escoge la opción “Casilla Prediseñada”.	10.0.1 El sistema muestra una nueva ventana donde ofrece una lista de casillas previamente diseñadas, además de otras opciones como: “Siguiente”, “Cancelar”.
10.1 El profesor escoge la casilla que desea personalizar y presiona el botón “Siguiente”.	10.1.1 El sistema muestra una nueva ventana en la que ofrece opciones para personalizar los valores de “Dimensiones” y

	“Contexto”, además de otras opciones como “Aceptar”, “Cancelar” y “Anterior”.
10.2 El profesor elige la opción “Anterior”.	10.2.1 El sistema vuelve al paso (10.0.1).
10.3 El profesor elige la opción “Cancelar”.	10.3.1 El sistema anula la nueva casilla.
Pos condiciones:	Las casillas diseñadas se guardan en el fichero en el que se encuentra el juego ya previamente diseñado.

Tabla 11: Descripción del Caso de Uso Modificar Casillas

CU-3	Modificar Casillas.
Actores:	Profesor
Propósito:	Modificar Casillas.
<u>Resumen</u> : Se escogen la casilla previamente insertada y se modifica.	
Referencias:	R2
Precondiciones:	El tablero debe estar creado y debe existir al menos una casilla insertada.
<u>Acción del actor</u>	<u>Respuesta del sistema</u>

<p>1 El profesor selecciona (mediante el clic “Secundario”) la casilla que desea modificar.</p>	<p>1.0.1 El sistema le proporciona las siguientes opciones:</p> <ul style="list-style-type: none">• Transformar• Rotar• Pasar al Frente• Copiar• Pegar• Eliminar• Propiedades
<p>1.1 El profesor selecciona la opción deseada.</p>	<p>2 Si el profesor selecciona la opción “Transformar”. Ver sección “Transformar”.</p> <p>3 Si el profesor selecciona la opción “Rotar Imagen”. Ver sección “Rotar Imagen”.</p> <p>4 Si el profesor selecciona la opción “Pasar al Frente”. Ver sección “Pasar al Frente”.</p> <p>5 Si el profesor selecciona la opción “Copiar”. Ver sección “Copiar”.</p> <p>6 Si el profesor selecciona la opción “Pegar”. Ver sección “Pegar”.</p> <p>7 Si el profesor selecciona la</p>

	opción “Eliminar”. Ver sección “Eliminar”.
8	Si el profesor selecciona la opción “Propiedades”. Ver sección “Propiedades”.
Sección Transformar.	
2	El profesor selecciona la opción Transformar.
1.0.1	El sistema muestra los diferentes diseños de Casillas.
2.1	El profesor selecciona el tipo de Casilla que desea.
2.1.1	El sistema modifica la forma de la casilla.
Sección Rotar Imagen.	
3	El profesor selecciona la opción Rotar Imagen.
3.0.1	El sistema ofrece las opciones de Rotar la Imagen a: <ul style="list-style-type: none"> • 90° • 180° • 270°.
3.1	El profesor elige la opción que desee.
3.1.1	El sistema Rota la Imagen.
Sección Pasar al Frente.	
4	El profesor selecciona la opción Pasar al Frente.
4.0.1	tema muestra la casilla por delante de las que estén colocadas cercanas a ella.

Sección Copiar.	
5 El profesor elige la opción Copiar.	5.0.1 El sistema almacena la casilla en memoria.
Sección Pegar.	
6 El profesor elige la opción Pegar.	6.0.1 El sistema crea un duplicado de la casilla que esté almacenada en memoria.
Sección Eliminar.	
7 El profesor elige la opción Eliminar.	7.0.1 El sistema elimina la casilla seleccionada.
Sección Propiedades.	
8 El profesor elige la opción Propiedades.	8.0.1 El sistema muestra las propiedades de la casilla seleccionada.
Pos condiciones:	El sistema almacena los cambios hechos a las casillas en el tablero.

Tabla 12: Descripción del Caso de Uso Modificar Datos

CU-4	Modificar Datos.
Actores:	Profesor
Propósito:	Modificar dado.

<u>Resumen</u> : Se escogen los dados previamente seleccionados y se modifican.	
Referencias:	R3
Precondiciones:	El tablero debe estar creado y el dado se encuentra seleccionado.
<u>Acción del actor</u>	<u>Respuesta del sistema</u>
1 El profesor selecciona (con doble clic) el Dado.	<p>1.1 El sistema muestra en una nueva ventana la opción Cambiar color, en los formatos:</p> <ul style="list-style-type: none"> • Muestras • HSB • RGB <p>1.2 El sistema muestra además los botones “Aceptar” y “Cancelar”.</p>
2 El profesor selecciona el color deseado y presiona el botón Aceptar.	1.1 El sistema guarda el color seleccionado.
3 El profesor presiona el botón Cancelar.	3.1 El sistema retira la ventana y mantiene las anteriores propiedades del Dado.

Tabla 13: Descripción del Caso de Uso Gestionar Fichas

CU-5	Gestionar Fichas
------	------------------

Capítulo 3. Análisis y diseño del sistema.

Actores:	Profesor
Propósito:	Seleccionar, modificar o eliminar ficha.
<u>Resumen</u> : Se escogen las fichas deseadas y estas pueden ser modificadas o eliminadas.	
Referencias:	R4
Precondiciones:	Debe estar creado un fichero (tablero) en la aplicación.
<u>Acción del actor</u>	<u>Respuesta del sistema</u>
1 El profesor selecciona el Botón "Ficha".	1.0.1 El sistema muestra una nueva ventana en la que el profesor escogerá la ficha de su preferencia.
1.1 El profesor escoge la ficha deseada y presiona el botón Aceptar.	1.1.1 El sistema inserta la ficha en el panel inferior derecho del tablero y la elimina de la lista de selecciones.
1.2 El profesor selecciona (con clic derecho) la Ficha que desee gestionar	1.2.1 El sistema muestra las opciones: <ul style="list-style-type: none"> • Eliminar. • Modificar
2 El profesor escoge la opción que deseada	3 Si el profesor selecciona la opción Modificar Casilla, ver

		sección Modificar Casilla.
	4	Si el profesor selecciona la opción Eliminar Casilla, ver sección Eliminar Casilla.
Sección Modificar Casilla.		
3	El profesor selecciona la opción Modificar Casilla	3.1 Ver Sesión 1.0.1
Sección Modificar Casilla.		
4	El profesor selecciona la opción Eliminar Casilla.	4.1 El sistema elimina la ficha del panel inferior derecho del tablero y la inserta en la lista de selecciones.
	Pos condiciones:	Las fichas seleccionadas se guardan en el fichero en el que se encuentra el juego ya previamente diseñado y actualiza la lista de fichas a seleccionar según las opciones realizadas.

Tabla 14: Descripción del Caso de Uso Gestionar Imagen

CU-6	Gestionar Imagen.
Actores:	Profesor

Capítulo 3. Análisis y diseño del sistema.

Propósito:	Insertar o eliminar imagen.
<u>Resumen</u> : Se escoge una imagen existente y esta puede ser eliminada en caso de que no se necesite por el tipo de juego.	
Referencias:	R5
Precondiciones:	Debe estar creado o abierto un fichero (tablero) en la aplicación.
<u>Acción del actor</u>	<u>Respuesta del sistema</u>
Sección Insertar Imagen.	
1 El profesor escoge la opción "Insertar Imagen"	1.1 El sistema muestra una nueva ventana donde el profesor puede cargar la dirección de la imagen que desea insertar.
2 El profesor carga la imagen que desea.	2.1 El sistema inserta la imagen y activa la opción eliminar (ver sección "Eliminar").
Sección Eliminar.	
3 El profesor selecciona la opción Eliminar Imagen.	3.1 El sistema elimina la imagen
Pos condiciones:	La imagen escogida se guarda en el fichero en el que se encuentra el juego ya previamente diseñado.

Tabla 15: Descripción del Caso de Uso Gestionar Texto

CU-7	Gestionar Texto.	
Actores:	Profesor	
Propósito:	Insertar modificar o eliminar texto.	
<u>Resumen</u> : Se inserta un texto en el tablero que puede ser modificado o eliminado.		
Referencias:	R6	
Precondiciones:	Debe estar creado o abierto un fichero (tablero) en la aplicación.	
<u>Acción del actor</u>	<u>Respuesta del sistema</u>	
1 El profesor selecciona el menú Texto.	1.1 El sistema muestra varias opciones: 1.1.1 Insertar Texto. 1.1.2 Modificar Texto. 1.1.3 Eliminar texto.	
2 El profesor selecciona la opción deseada.	3	Si el profesor selecciona la opción "Insertar Texto". Ver sección "Insertar Texto".
	4	Si el profesor selecciona la opción "Modificar Texto", ver sección "Modificar Texto".
	5	Si el profesor selecciona

	la opción Eliminar texto” ver sección “Eliminar texto”.
Sección Insertar Texto	
3 El profesor selecciona la opción “Insertar Texto”.	3.1 El sistema crea un fichero de texto. 3.2 El sistema muestra el panel donde será insertado el texto. 3.3 Guarda los cambios ver sección Guardar Cambios.
Sección Modificar Texto.	
4 El profesor escoge la opción “Modificar Texto”.	4.0.1 El sistema muestra el menú “Modificar Texto” donde ofrece las opciones: <ul style="list-style-type: none"> • Cambiar Fuente. • Cambiar color • Cambiar tamaño de la letra.
4.1 El profesor realiza los cambios deseados.	4.1.1 El sistema actualiza el fichero.
Sección Eliminar Texto.	
5 El profesor escoge la opción “Eliminar Texto”.	5.1 El sistema solicita la autorización para eliminar y elimina el fichero.

Pos condiciones:	El texto insertado se guarda en un nuevo fichero en el que se encuentra el juego ya previamente diseñado.
---------------------	---

Capítulo 3. Análisis y diseño del sistema.

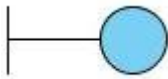
En esta sección se describen las características más importantes del Análisis y Diseño de la herramienta para el diseño de juegos didácticos de tableros, se explica el flujo de los procesos y se ofrece una representación de los Casos de Uso más significativos representadas en UML y diseñadas con la herramienta Visual Paradigm.

3.1 Características fundamentales del Análisis.

Análisis: Consiste en obtener una visión del sistema, sin tomar en cuenta el lenguaje de programación o la plataforma en que se ejecute la aplicación y no precisa como se implementa la herramienta. El Análisis se preocupa solo de ver que hace el sistema, de modo que solo se interesa por los requisitos funcionales, permitiendo estructurar los requisitos de manera que nos facilite su comprensión su modificación y su mantenimiento.

Las clases que se utilizan para el modelado de los diagramas de clases son de tres tipos: Interfaz, Control y Entidad.

Interfaz: Modelan la interacción entre el sistema y sus actores.



Control: Coordinan la realización de uno o unos pocos Casos de Uso coordinando las actividades de los objetos que implementan la funcionalidad del Caso de Uso.



Entidad: Modelan la información que posee larga vida y que es a menudo persistente.

3.1.1 Diagrama de Clases del Análisis.

Caso de Uso: Gestionar Tablero.

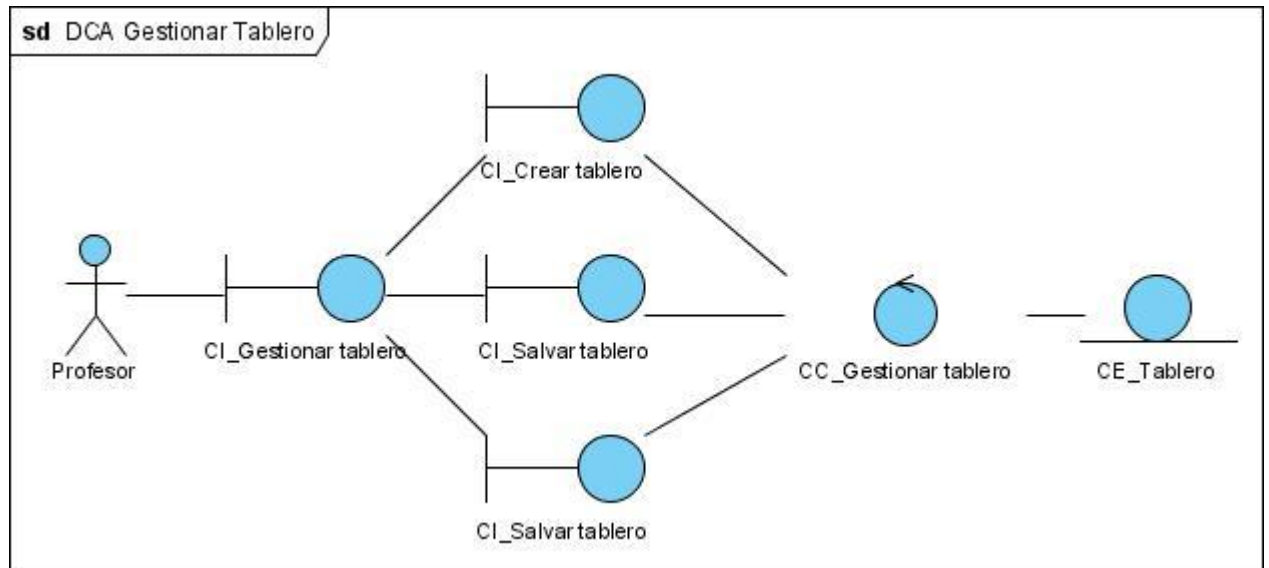


Figura 3: Diagrama de Clases del Análisis del CU Gestionar Tablero.

Caso de Uso: Gestionar Casilla.

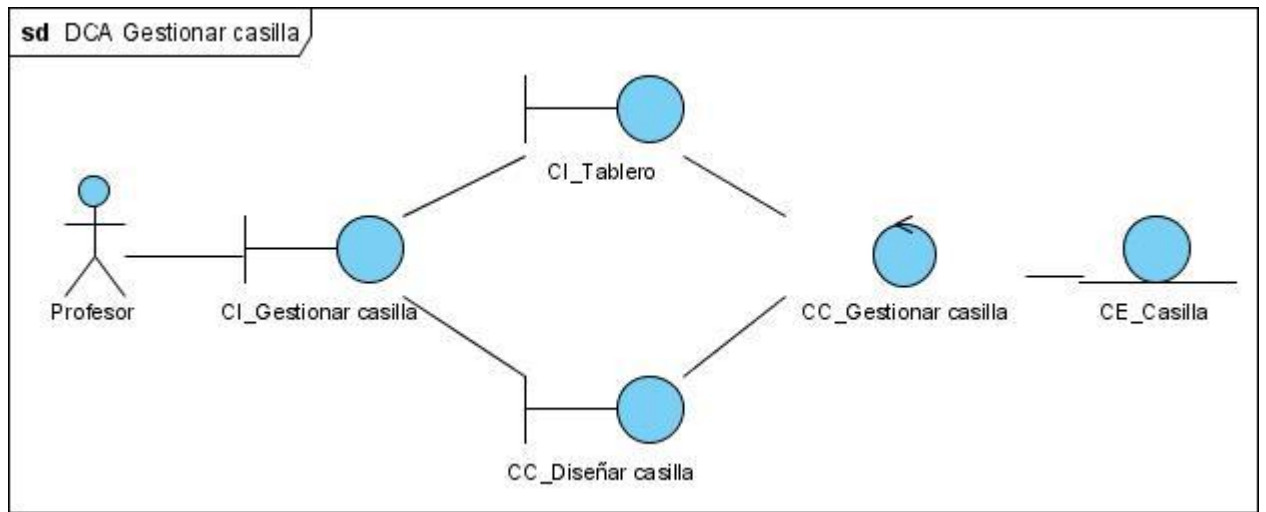


Figura 4: Diagrama de Clases del Análisis del CU Gestionar Casilla.

Caso de Uso: Gestionar Ficha.

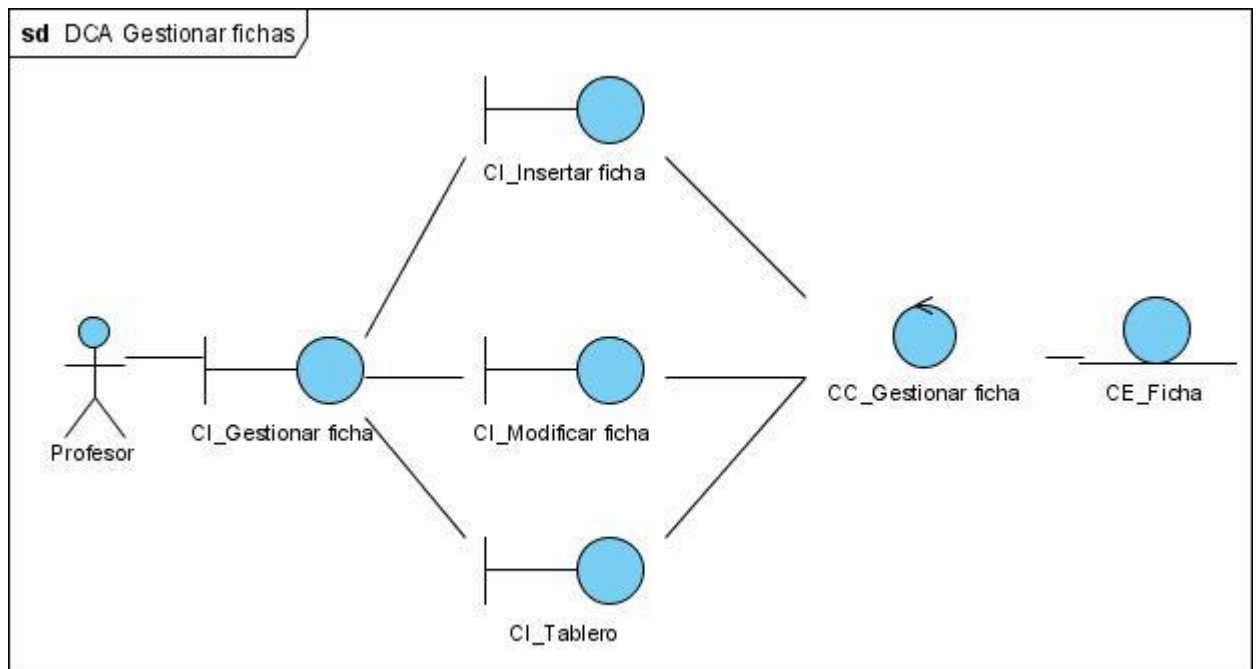


Figura 5: Diagrama de Clases del Análisis del CU Gestionar Ficha.

Caso de Uso: Gestionar Texto.

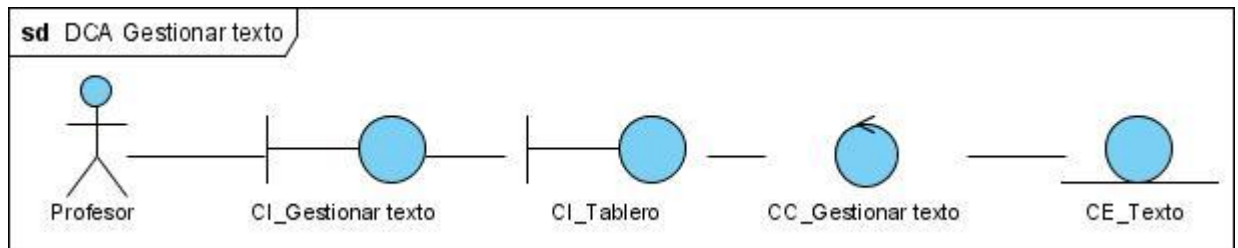


Figura 6: Diagrama de Clases del Análisis del CU Gestionar Texto.

Caso de Uso: Modificar Dado.

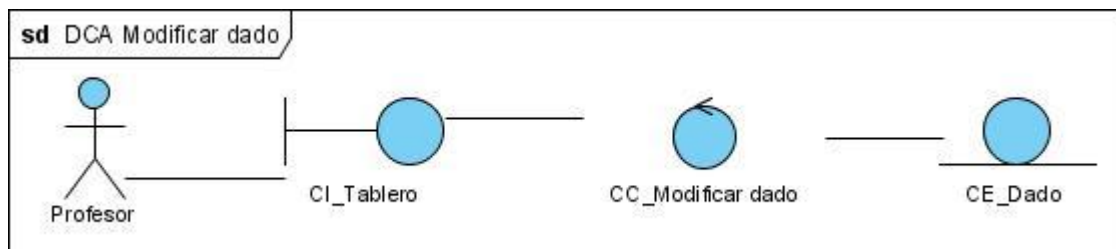


Figura 7: Diagrama de Clases del Análisis del CU Modificar Dado.

Caso de Uso: Gestionar Imágenes.

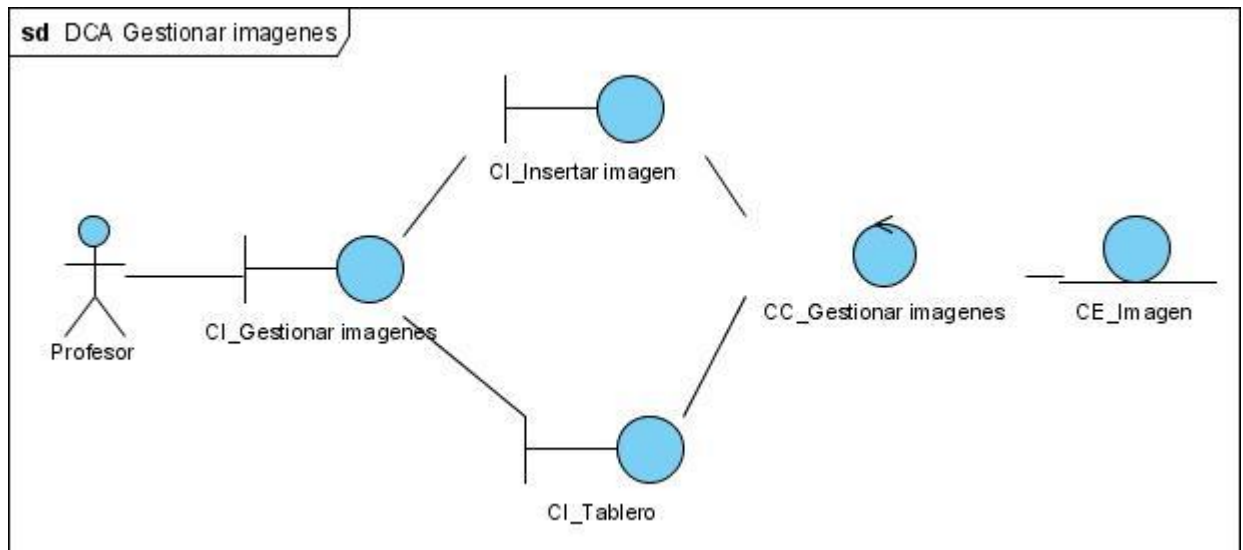


Figura 8: Diagrama de Clases del Análisis del CU Gestionar Imágenes.

3.2 Diseño

El Diseño es un refinamiento del Análisis que tiene en cuenta los requisitos no funcionales, es decir como cumple el sistema sus objetivos. El Diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

3.2.1 Diagrama interacción

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento

Caso de Uso: Gestionar Tablero.

Capítulo 3. Análisis y diseño del sistema.

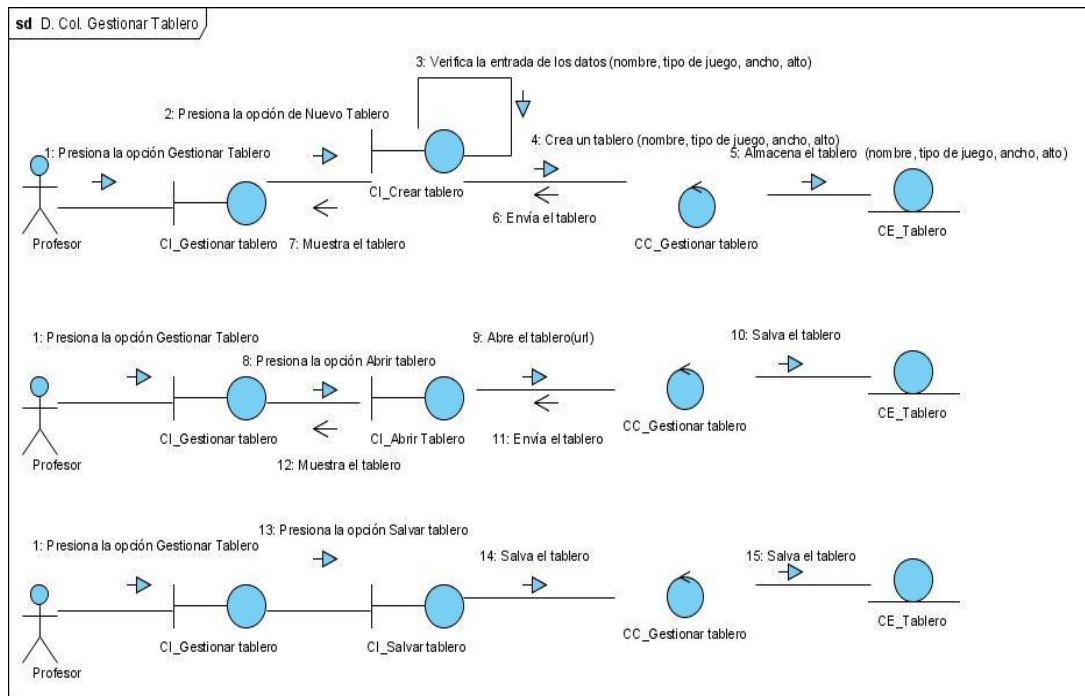


Figura 9: Diagrama de Colaboración del CU Gestionar Tablero.

Caso de Uso: Gestionar Casilla.

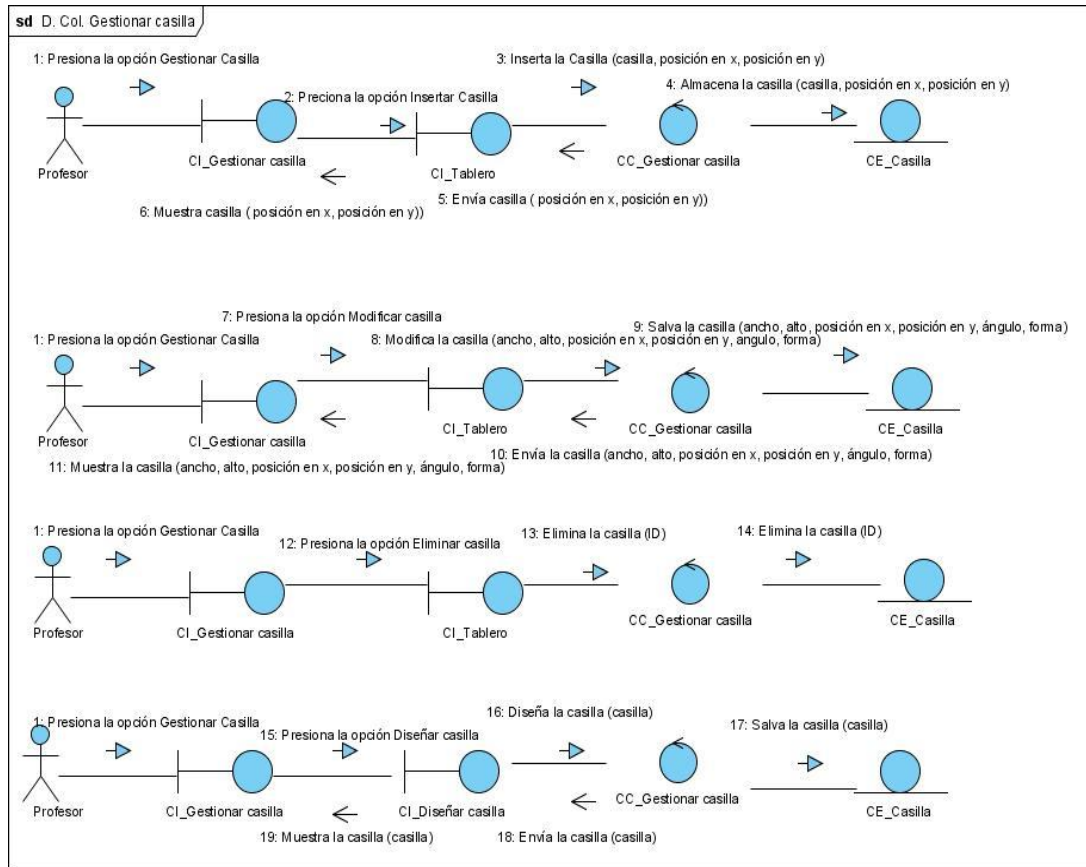


Figura 10: Diagrama de Colaboración del CU Gestionar Casilla.

3.2.2 Diagrama de Clases.

El siguiente diagrama muestra las clases que dan respuesta al problema, destacándose entre las mismas las clases: *Tablero*, *Texto*, *Dado*, *Ficha*, *Imágenes*, *Casilla* y *Tabla*; las cuales forman la columna vertebral del programa por lo que se hace necesario que estén relacionadas entre sí, y con otras clases que le dan terminación al programa añadiéndole valor agregado al mismo, mediante relaciones de tipo “agregación”, “composición” y “generalización/especialización”.

En el diagrama se representa la clase *Tablero* la cual está compuesta por objetos de las clases *Texto*, *Dado*, *Ficha*, *Imágenes*, *Casilla* (o sea, puede existir un tablero sin ninguna instancia de estas clases, que en este caso sería un tablero vacío), la misma clase *Texto* posee relación de agregación con las clases *TextoPanel* y *Seleccionado* (debido a que la clase *Texto* es la abstracción de una caja de texto que da la posibilidad de escribir en ella y alterar sus dimensiones. Para que esta exista necesita de los contenedores conceptualizados en las clases *TextoPanel* y *Seleccionado*).

Se observan también en el diagrama varias relaciones de generalización/especialización principalmente entre las clases de los eventos, tales como la clase *Eventos* (la cual es una clase genérica) de la cual heredan las clases *EventosTableros*, *EventosJIFrame*, *EventosCasilla*, *EventosCasTabla*, entre otras que se pueden observar en la Ilustración 11.

Mediante el uso de estas clases, se define la abstracción de la dependencia entre los componentes de un tablero físico y el creado con la herramienta, así como las acciones que se pueden realizar sobre el mismo.

Capítulo 3. Análisis y diseño del sistema.

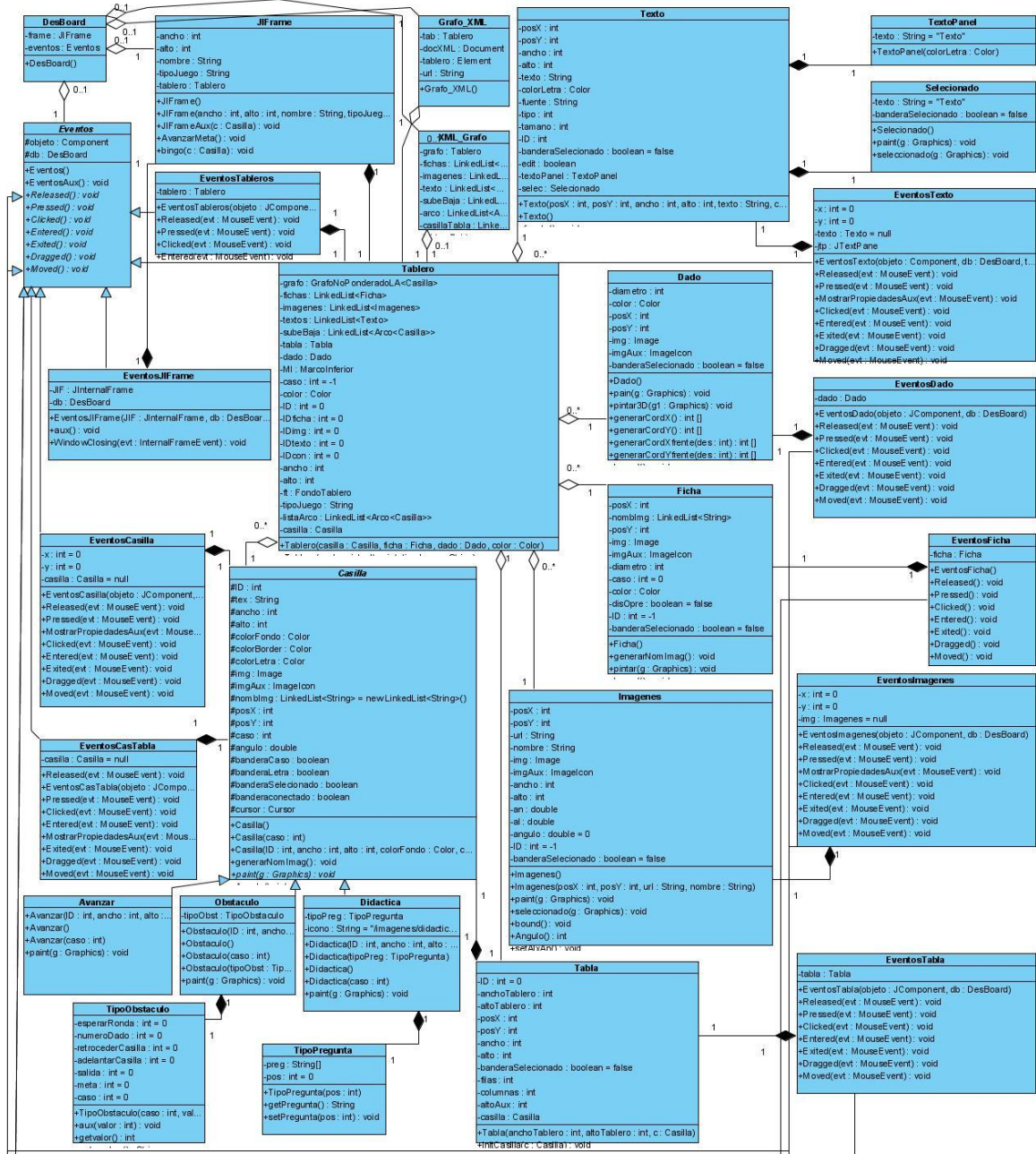


Figura 11: Diagrama de Clases.

3.2.3 Descripción de clases

A continuación se realiza la descripción detallada de las principales clases que conforman el sistema, especificando los atributos y las operaciones que las mismas realizan.

Tabla 16: Descripción de la clase JFrame

Nombre: JFrame	
Tipo de clase: Interface	
Atributo	Tipo
ancho	int
alto	int
nombre	String
tipoJuego	String
tablero	Tablero
Para cada responsabilidad:	
Nombre:	JFrame()
Descripción:	Inicializa los atributos de la clase
Nombre:	JFrame(int ancho, int alto, String nombre, String tipoJuego, Casilla c)
Descripción:	Inicializa los atributos de este objeto con los parámetros recibidos.
Nombre:	void AvanzarMeta()

Descripción:	Crea el tipo de juego “Avanzar hasta la Meta”.
Nombre:	void bingo(Casilla c)
Descripción:	Crea el tipo de juego “Bingo”.

Tabla 17: Descripción de la clase Tablero

Nombre: Tablero	
Tipo de clase: Controladora	
Atributo	Tipo
grafo	GrafoNoPonderadoLA<Casilla>
fichas	LinkedList<Fichas>
imágenes	LinkedList<Imágenes>
textos	LinkedList<Textos>
tabla	Tabla
dado	Dado
MI	MarcoInferior
caso	Int
color	Color
ID	Int

IDimg	Int
IDtex	Int
IDficha	Int
IDcon	Int
ancho	Int
alto	int
ft	FondoTabler
tipoJuego	String
listaArcos	LinkedList<Casilla>
casilla	Casilla
subeBaja	LinkedList<Arco<Casilla>>
Para cada responsabilidad:	
Nombre:	Tablero(int ancho, int alto, String tipoJuego)
Descripción:	Inicializa los atributos de este objeto con los parámetros recibidos.
Nombre:	void setCasilla(Casilla casilla)
Descripción:	Cambia el valor de la casilla antes de ser insertada
Nombre:	void adicionaCasilla(int x, int y)
Descripción:	Inserta la casilla en las coordenadas recibidas por parámetro y en el grafo
Nombre:	void eliminarCasilla(int ID)

Capítulo 3. Análisis y diseño del sistema.

Descripción:	Elimina la casilla correspondiente al ID recibido por parámetro
Nombre:	void adicionarImágenes(Imágenes img)
Descripción:	Inserta la imagen recibidas por parámetro en el tablero y en la lista de imágenes
Nombre:	eliminarImágenes(int ID)
Descripción:	Elimina la imagen correspondiente al ID recibido por parámetro
Nombre:	adicionarSubeBaja(Conexion con)
Descripción:	Inserta la conexión recibidas por parámetro en el tablero y en la lista de subeBaja
Nombre:	void eliminarArcoSB(Arco<Casilla> a)
Descripción:	Elimina el arco subeBaja correspondiente al arco recibido por parámetro
Nombre:	void adicionarTabla(Casilla c)
Descripción:	Inicializa la tabla del tipo de juego Bingo con la casilla recibida por parámetros
Nombre:	void desmarcarComponentes()
Descripción:	Desmarca todos lo componentes insertados en el tablero.
Nombre:	void pasarFrenteCasilla(int IDaux, Casilla cas)
Descripción:	Trae al frente la casilla recibida por parámetro
Nombre:	void pasarFrenteTexto(int IDaux, Texto tex)

Capítulo 3. Análisis y diseño del sistema.

Descripción:	Trae al frente el texto recibido por parámetro.
Nombre:	void pasarFrentelImagen(int IDaux, Imagenes img)
Descripción:	Trae al frente la imagen recibida por parámetro
Nombre:	void adicionarDado(Dado dado)
Descripción:	Inserta el dado recibido por parámetro en el tablero.
Nombre:	void adicionarFicha(Ficha ficha)
Descripción:	Inserta la ficha recibida por parámetro en el tablero y en la lista de fichas
Nombre:	void eliminarFicha(int IDfich)
Descripción:	Elimina la ficha correspondiente al ID recibido por parámetro
Nombre:	void adicionarTexto(Texto texto)
Descripción:	Inserta el texto recibido por parámetro en el tablero y en la lista de textos.
Nombre:	void eliminarTexto(int ID)
Descripción:	Elimina el texto correspondiente al ID recibido por parámetro

Tabla 18: Descripción de la clase Casilla

Nombre: Casilla	
Tipo de clase: Controladora	
Atributo	Tipo
ID	int
tex	String
alto	int
ancho	int
posX	Int
posY	Int
colorFondo	Color
colorLetra	Color
colorBorder	Color
caso	int
angulo	Double
banderaCaso	boolean
banderaLetra	boolean
banderaSeleccionado	boolean
banderaConectado	boolean
nomImg	LinkedList<String>
Para cada responsabilidad:	

Nombre:	Casilla()
Descripción:	Inicializa los atributos de la clase
Nombre:	Casilla(int ID, int ancho, int alto, Color colorFondo, Color colorBorder, Color colorLetra, int posX, int posY, int caso, double angulo, boolean banderaCaso, boolean banderaLetra, boolean banderaSeleccionado, boolean banderaconectado, String tex)
Descripción:	Inicializa los atributos con los parámetros recibidos.
Nombre:	void generarNomlmg()
Descripción:	Agrega a lista nomlmg el nombre de las imágenes prediseñadas que se utilizan en el diseño de las casillas
Nombre:	abstract void paint(Graphics g)
Descripción:	Será implementado en las clases que heredan de esta.
Nombre:	int Angulo()
Descripción:	Garantiza que el atributo ángulo siempre tome el valor de 0, 90, 180 y 270 grados
Nombre:	void pintarPre(Graphics g)
Descripción:	Pinta la casilla con la imagen seleccionada
Nombre:	void pintarDis(Graphics g)
Descripción:	Pinta la casilla con el color seleccionado

Nombre:	void seleccionado(Graphics g)
Descripción:	Pinta en la casilla la vista de seleccionado
Nombre:	void Conectando(Graphics g)
Descripción:	Pinta en la casilla la vista de conectado

Tabla 19: Descripción de la clase Obstaculo

Nombre: Obstaculo: Casilla	
Tipo de clase: Controladora	
Atributo	Tipo
tipoObst	TipoObstaculo
Para cada responsabilidad:	
Nombre:	Obstaculo()
Descripción:	Inicializa los atributos de la clase
Nombre:	Obstaculo(int ID, int ancho, int alto, Color colorFondo, Color colorBorder, Color colorLetra, int posX, int posY, int caso, double angulo, boolean banderaCaso, boolean banderaLetra, boolean banderaSeleccionado, boolean banderaconectado, String tex, TipoObstaculo tipoObst)
Descripción:	Inicializa los atributos de la clase con los parámetros recibidos

Nombre:	void paint(Graphics g)
Descripción:	Pinta la casilla

Tabla 20: Descripción de la clase Avanzar

Nombre: Avanzar: Casilla	
Tipo de clase: Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	Avanzar()
Descripción:	Inicializa los atributos de la clase
Nombre:	Avanzar(int ID, int ancho, int alto, Color colorFondo, Color colorBorder, Color colorLetra, int posX, int posY, int caso, double angulo, boolean banderaCaso, boolean banderaLetra, boolean banderaSeleccionado, boolean banderaconectado, String tex)
Descripción:	Inicializa los atributos de la clase con los parámetros recibidos
Nombre:	void paint(Graphics g)
Descripción:	Pinta la casilla

Tabla 21: Descripción de la clase Didactica

Nombre: Didactica: Casilla	
Tipo de clase: Controladora	
Atributo	Tipo
tipoPreg;	TipoPregunta
Para cada responsabilidad:	
Nombre:	Didactica ()
Descripción:	Inicializa los atributos de la clase
Nombre:	Didactica (int ID, int ancho, int alto, Color colorFondo, Color colorBorder, Color colorLetra, int posX, int posY, int caso, double angulo, boolean banderaCaso, boolean banderaLetra, boolean banderaSeleccionado, boolean banderaconectado,String tex, TipoPregunta tipoPreg)
Descripción:	Inicializa los atributos de la clase con los parámetros recibidos
Nombre:	void paint(Graphics g)
Descripción:	Pinta la casilla

Nombre: Imagenes
Tipo de clase: Controladora

Atributo	Tipo
ID	int
nombre	String
alto	int
ancho	int
posX	Int
posY	Int
url	String
img	Image
imgAux	ImageIcon
caso	int
ángulo	Double
an	int
al	int
banderaSeleccionado	boolean
Para cada responsabilidad:	
Nombre:	Imagenes()
Descripción:	Inicializa los atributos de la clase
Nombre:	Imagenes(int posX, int posY, String url, String nombre)
Descripción:	Inicializa los atributos con los parámetros recibidos.

Nombre:	void paint(Graphics g)
Descripción:	Pinta la imagen en el tablero
Nombre:	int Angulo()
Descripción:	Garantiza que el atributo ángulo siempre tome el valor de 0, 90, 180 y 270 grados
Nombre:	void seleccionado(Graphics g)
Descripción:	Pinta la imagen con la vista de seleccionado

Tabla 22: Descripción de la clase Texto

Nombre: Texto	
Tipo de clase: Controladora	
Atributo	Tipo
ID	int
texto	String
alto	int
ancho	int
posX	Int
posY	Int

fuente	String
tipo	int
tamano	int
textoPanel	TextoPanel
banderaSeleccionado	boolean
colorLetra	Color
selec	Seleccionado
Para cada responsabilidad:	
Nombre:	Texto()
Descripción:	Inicializa los atributos de la clase
Nombre:	Texto(int posX, int posY, int ancho, int alto, String texto, Color colorLetra, String fuente, int tipo, int tamano, int ID, boolean edit)
Descripción:	Inicializa los atributos con los parámetros recibidos.
Nombre:	void fuente()
Descripción:	Modifica el tipo y el tamaño de letra del atributo texto

Tabla 23: Descripción de la clase TextoPanel

Nombre: TextoPanel
Tipo de clase: Controladora

Atributo	Tipo
texto	String
Para cada responsabilidad:	
Nombre:	TextoPanel(Color colorLetra)
Descripción:	Inicializa los atributos con los parámetros recibidos.

Tabla 24: Descripción de la clase Seleccionado

Nombre: Seleccionado	
Tipo de clase: Controladora	
Atributo	Tipo
texto	String
banderaSeleccionado	boolean
Para cada responsabilidad:	
Nombre:	Seleccionado()
Descripción:	Inicializa los atributos de la clase
Nombre:	void paint(Graphics g)
Descripción:	Pinta la imagen en el tablero
Nombre:	void seleccionado(Graphics g)
Descripción:	Pinta el texto con la vista de seleccionado

Tabla 25: Descripción de la clase Ficha

Nombre: Ficha	
Tipo de clase: Controladora	
Atributo	Tipo
ID	int
nombre	LinkedList<String>
diametro	int
posX	Int
posY	Int
img	Image
imgAux	ImageIcon
caso	int
color	Color
banderaSeleccionado	boolean
Para cada responsabilidad:	
Nombre:	Ficha()
Descripción:	Inicializa los atributos de la clase
Nombre:	void pintar(Graphics g)
Descripción:	Pinta la ficha
Nombre:	void seleccionado(Graphics g)

Descripción:	Pinta la imagen con la vista de seleccionado
Nombre:	void generarNomlMag()
Descripción:	Agrega a lista nomlMag el nombre de las imágenes prediseñadas que se utilizan en el diseño de las fichas

Tabla 26: Descripción de la clase Dado

Nombre: Dado	
Tipo de clase: Controladora	
Atributo	Tipo
diametro	int
posX	Int
posY	Int
img	Image
imgAux	Imagelcon
color	Color
banderaSeleccionado	boolean
Para cada responsabilidad:	
Nombre:	Dado()
Descripción:	Inicializa los atributos de la clase
Nombre:	void pain(Graphics g)

Descripción:	Pinta el dado
Nombre:	void seleccionado(Graphics g)
Descripción:	Pinta la imagen con la vista de seleccionado
Nombre:	void pintar3D(Graphics g1)
Descripción:	Le añade un efecto 3D al dado

Tabla 27: Descripción de la clase Tabla

Nombre: Tabla	
Tipo de clase: Controladora	
Atributo	Tipo
ID	int
texto	String
alto	int
ancho	int
posX	int
posY	int
altoAux	int
altoTablero	int

anchoTablero	int
casillas	LinkedList<Casilla>
banderaSeleccionado	boolean
filas	int
columnas	int
Para cada responsabilidad:	
Nombre:	Tabla(int anchoTablero, int altoTablero, Casilla c)
Descripción:	Inicializa los atributos con los parámetros recibidos.
Nombre:	void insertarFilas(LinkedList<Casilla> aux)
Descripción:	Inserta una fila a la tabla.
Nombre:	void insertarColumnas(LinkedList<Casilla> aux)
Descripción:	Inserta una columna a la tabla.
Nombre:	void eliminarrColumnas()
Descripción:	Elimina una columna a la tabla.
Nombre:	void eliminarFilas()
Descripción:	Elimina una fila a la tabla.
Nombre:	void generarCordenadas()
Descripción:	Genera las coordenadas que tendrán las casillas de la tabla.
Nombre:	void generarID()

Descripción:	Genera el ID que tendrán las casillas de la tabla.
---------------------	--

Tabla 28: Descripción de la clase Grafo_XML

Nombre: Grafo_XML	
Tipo de clase: Controladora	
Atributo	Tipo
docXML	Document
tablero	Element
url	String
tab	Tablero
Para cada responsabilidad:	
Nombre:	Grafo_XML(Tablero tablero, String url)
Descripción:	Crea un fichero en la dirección recibida por parámetros salvando en el los atributos de tablero.

Tabla 29: Descripción de la clase XML_Grafo

Nombre: XML_Grafo

Tipo de clase: Controladora	
Atributo	Tipo
grafo	GrafoNoPonderadoLA<Casilla>
fichas	LinkedList<Fichas>
imágenes	LinkedList<Imágenes>
textos	LinkedList<Textos>
tabla	Tabla
dado	Dado
IDcon	Int
ancho	Int
alto	int
url	String
casillaTabla	LinkedList<Casilla>
arco	LinkedList<Arco<Casilla>>
subeBaja	LinkedList<Arco<Casilla>>
Para cada responsabilidad:	
Nombre:	XML_Grafo(String url)
Descripción:	Parsea el fichero XML.

3.2.4 Tratamiento de errores

La confiabilidad del sistema constituye un punto fundamental en la interacción con el usuario, para ello se lleva a cabo el tratamiento de errores para los posibles fallos que se pueden encontrar en el sistema durante el trabajo con la aplicación. Siempre que ocurra un error o se lance una excepción el sistema notificará inmediatamente al usuario. La notificación de los errores se realiza mediante ventanas de error que genera el sistema, de tal modo que el usuario entienda la naturaleza del error y pueda corregirlo. Otras de las alternativas del sistema para tratar los errores es la validación de la entrada de datos, verificando que los mismos sean del tipo de dato correcto.

3.2.5 Interfaz.

La interfaz del sistema es sencilla y fácil de entender. Cuenta con un menú Archivo que permite crear, abrir y guardar un tablero de juego, una vez seleccionada una de estas dos últimas opciones la interfaz muestra un ventana que nos permite elegir la dirección en que se desea salvar o abrir el fichero, si el usuario opta por la opción de crear un nuevo tablero la aplicación le permite escoger el tipo de juego (Avanzar hasta la Meta, Sube y baja o Bingo) que se desea diseñar, además de poder redimensionar el mismo.

Una vez creado el lienzo, el sistema activa las opciones de diseño en el panel de componentes según el tipo de juego seleccionado. Estos componentes permiten realizar el diseño de un tablero de juegos, incorporando los elementos necesarios para el mismo, dígame insertar los diferentes tipos de casillas (Avanzar, Obstáculo y Didácticas), imágenes, textos, las fichas que representarán a los jugadores; además permite establecer la conexión entre las casillas definiendo así el camino por el que se desplazará el jugador.

El tablero cuenta con un contenedor desplegable que muestra las fichas escogidas por el diseñador, permitiéndole de esta manera poderlas eliminar o modificar según lo deseado. El contenedor también visualiza el dado que se utilizará en el juego, el mismo es insertado automáticamente una vez creado el tablero.

Conclusiones

A partir de los objetivos planteados y el trabajo realizado en esta investigación se arribó a los siguientes resultados:

- Se realizó un estudio detallado los sistemas y herramientas similares existentes, así como sus características y funcionalidades.
- Se diseñó una herramienta que permite el diseño eficiente de Tableros de Juegos Didácticos.
- Se implementó la herramienta DesBoard que permite el diseño de tableros de juegos didácticos e incluye todas las funcionalidades requeridas.

Por todo lo anteriormente expuesto es posible concluir que los objetivos trazados para el presente trabajo han sido cumplidos satisfactoriamente.

Recomendaciones

1. Confeccionar un documento Manual de Usuario que sirva de apoyo y guía técnica a los usuarios.
2. Investigar e implementar acerca de las nuevas funcionalidades que pueden adicionarse a la herramienta.
3. Perfeccionar la forma de salvar los tableros al fichero xml.

Referencias Bibliográficas

1. **Suárez Ferreiro, Dunia.** Herramienta Educativa sobre Software Libre para apoyar las asignaturas de Programación en la Universidad de las Ciencias Informáticas. C. Habana : s.n., 2009.
2. **Mendoza Sanchez, María A.** informatizate. [En línea] 7 de 6 de 2004. [Citado el: 18 de 12 de 2008.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
3. **Martinez Hidalgo, Hector Hugo.** lycos.es. *IDEs Open Source Java*. [En línea] lycos.es, 17 de 10 de 2007. [Citado el: 20 de 10 de 2008.] <http://usuarios.lycos.es/neossoftware/articles/ides/ides-java.php>.
4. **Ferreiro Cordero, Ana María.** Propuesta de Juegos Didácticos para la motivación hacia las Matemáticas de los estudiantes del Curso de Superación Integral para Jóvenes de la Sede Fulgencio Oroz. C. Habana : s.n., 2008.
5. **Casas Rodríguez, Dra. C. Ludmila, y otros.** La didáctica de la Educación Médica Superior utilizando software educativos. [En línea] 21 de 3 de 2008. [Citado el: 7 de 11 de 2008.] <http://www.amc.sld.cu/amc/2008/v12n3/amc15308.htm>.
6. **signando.com.** Signando.com. [En línea] 15 de 7 de 2007. [Citado el: 12 de 11 de 2008.] http://www.signando.com/index2.php?option=com_content&do_pdf=1&id=12.
7. neoparaiso. [En línea] 6 de 10 de 2008. [Citado el: 21 de 1 de 2009.] <http://neoparaiso.com/logo/software-educativo.html>.
8. **ConocimientosWeb.net.** ConocimientosWeb. [En línea] [Citado el: 12 de 5 de 2008.] <http://www.conocimientosweb.net/zip/article108.html>.
9. **Pages, Jaume.** Profesores Universia. [En línea] universia.es, 9 de 6 de 2006. [Citado el: 4 de 12 de 2008.] <http://profesores.universia.es/seccionEspecial.jsp?idEspecial=12&idSeccion=4230&title=CARACTERISTICAS-MOODLE>.
10. **Mondeja González, Diana, y otros.** JUEGOS DIDÁCTICOS: ¿ÚTILES EN LA EDUCACIÓN SUPERIOR? . C. Habana : s.n., 2003.
11. **Teleformación.** Entorno Virtual de Aprendizaje:. [En línea] 2005. [Citado el: 5 de 3 de 2009.] <http://eva.uci.cu/>.
12. **Moodle.** Moodle.org. [En línea] Moodle, 8 de 2002. [Citado el: 14 de 12 de 2008.] <http://moodle.org>.

Bibliografía

1. **Netbeans.org.** Netbeans.org. [En línea] [Citado el: 19 de 1 de 2009.] http://www.netbeans.org/index_es.html.
2. **Sun Microsystems.** Sun Microsystems. [En línea] 1994. [Citado el: 8 de 2 de 2009.] <http://java.sun.com/docs/books/tutorial/uiswing/>.
3. Visual Paradigm. [En línea] [Citado el: 14 de 3 de 2009.] <http://www.visual-paradigm.com/>.
4. **IBM.** IBM - Rational Unified Process (RUP). [En línea] IBM. [Citado el: 26 de 4 de 2009.] <http://www-01.ibm.com/software/awdtools/rup/>.
5. Java en castellano. [En línea] 1999. <http://www.programacion.com/java/tutoriales/>.

Anexos

Caso de Uso: Gestionar Ficha.

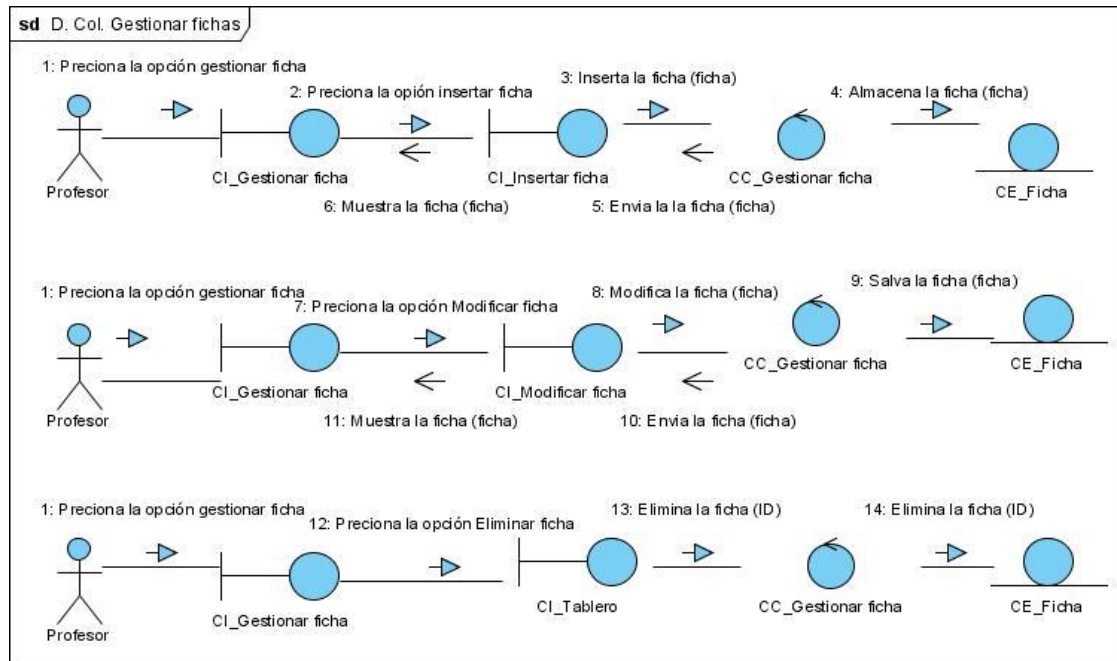


Figura 12: Diagrama de Colaboración del CU Gestionar Ficha.

Caso de Uso: Gestionar Imágenes.

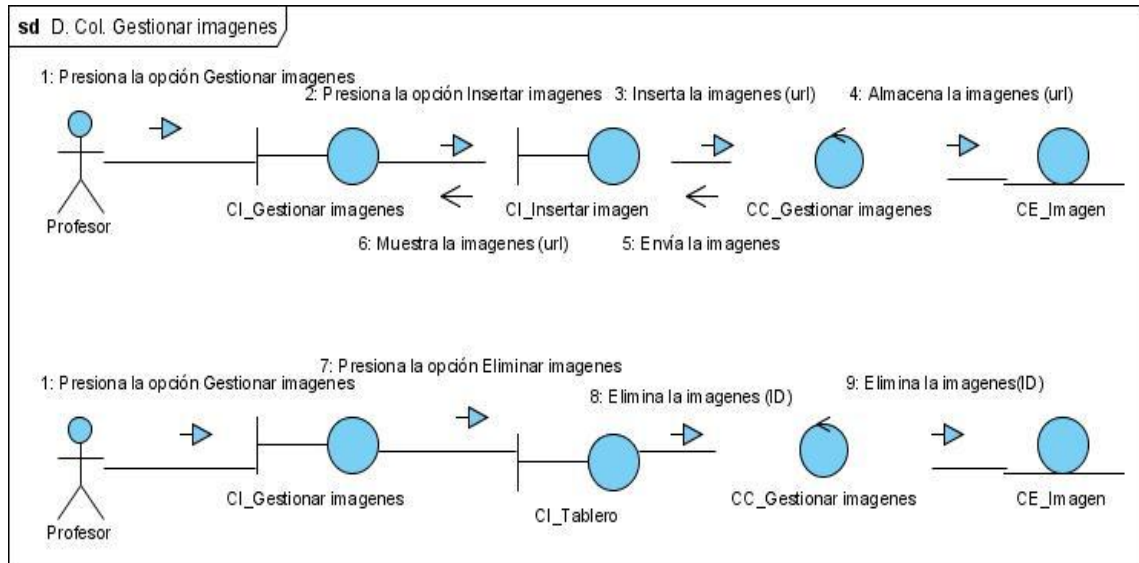


Figura 13: Diagrama de Colaboración del CU Gestionar Imágenes.

Caso de Uso: Gestionar Dado.

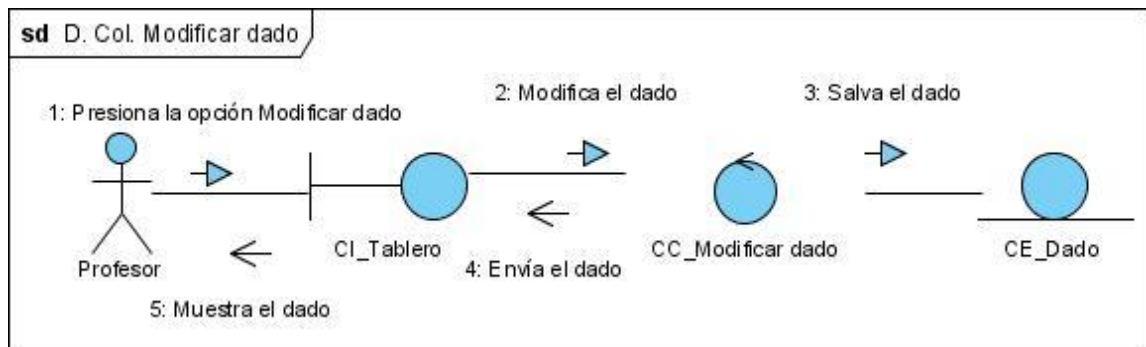


Figura 14: Diagrama de Colaboración del CU Gestionar Dado.

Caso de Uso: Gestionar Textos.

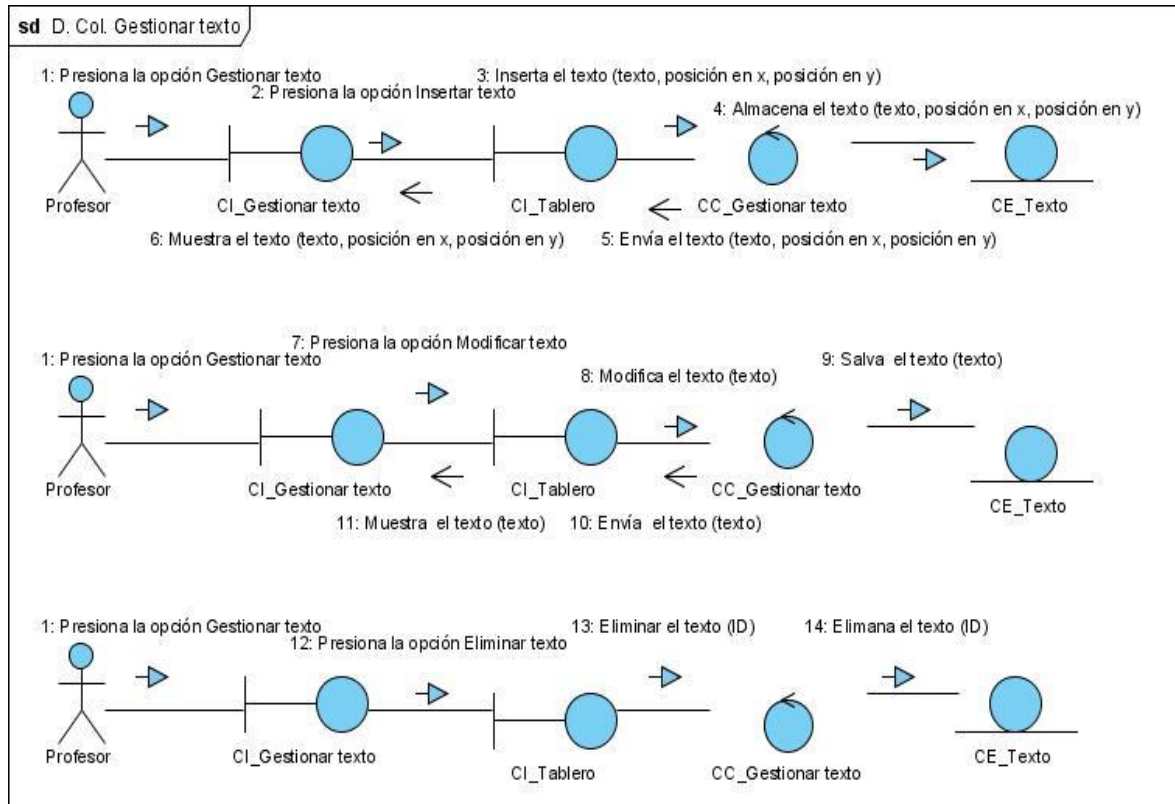


Figura 15: Diagrama de Colaboración del CU Gestionar Texto.

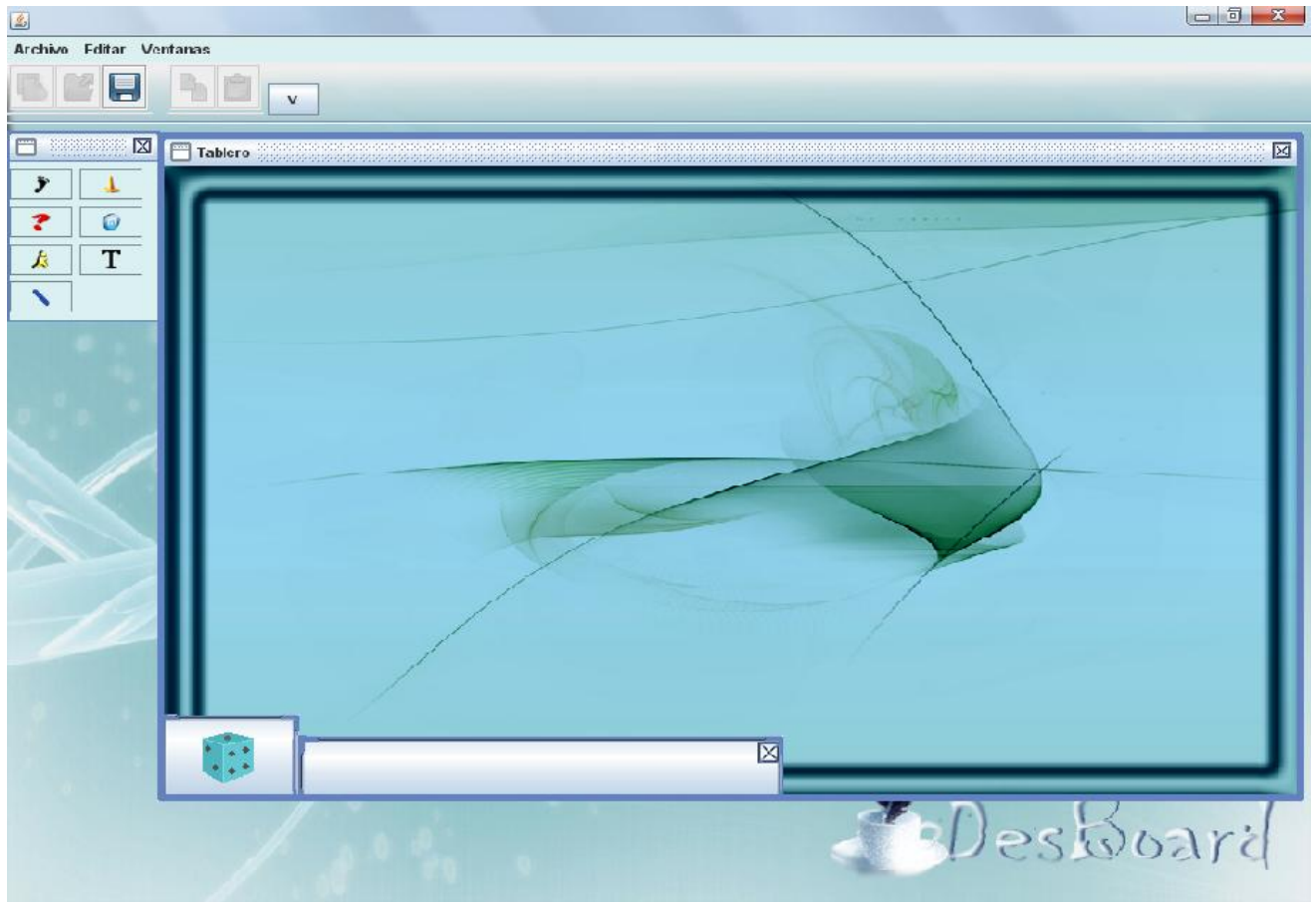


Figura 16: Pantalla de inicio

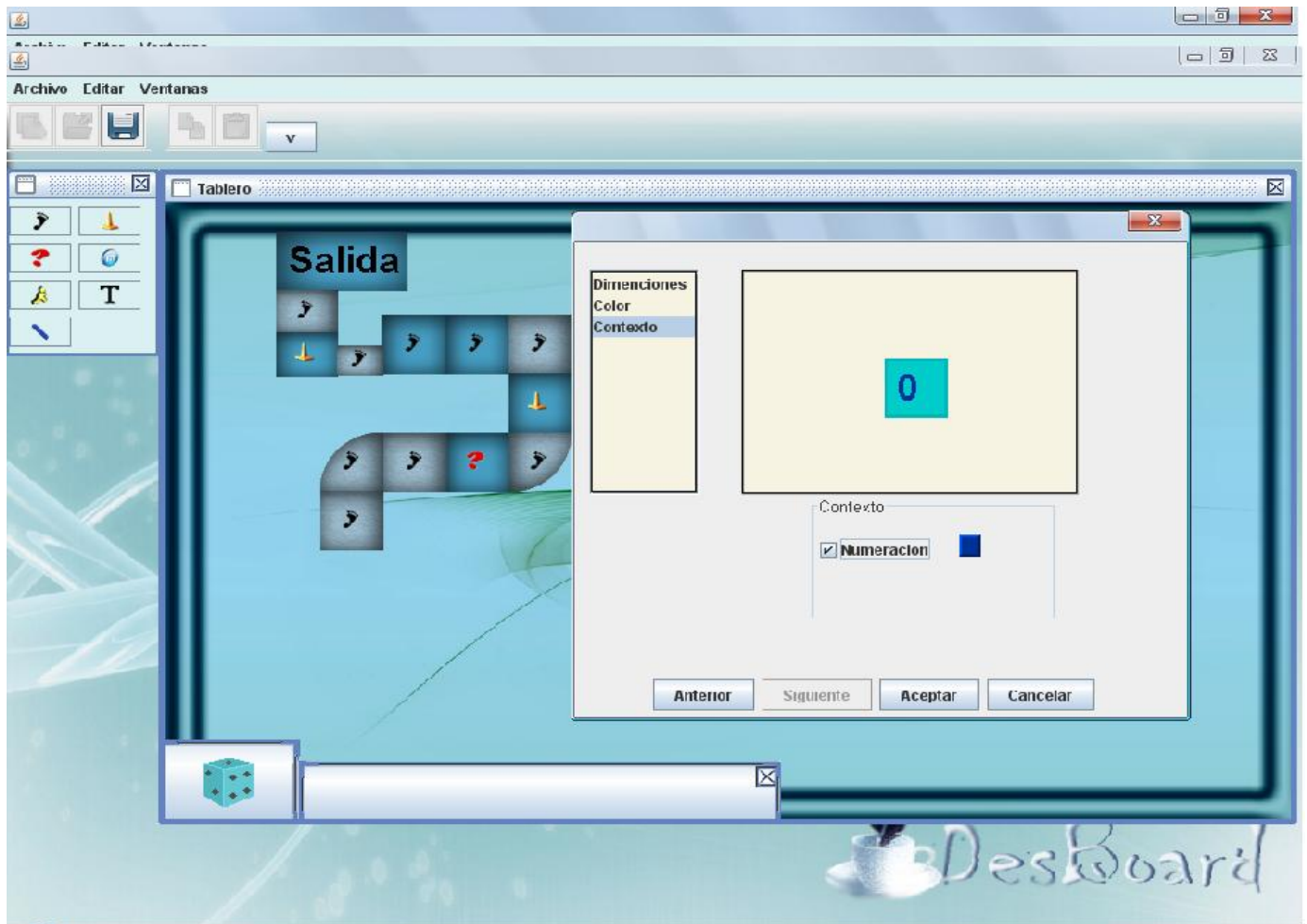


Figura 17: Pantalla de Edición de Casilla