

Universidad de las Ciencias Informáticas
Facultad #1

Título: Herramienta de autor para el diseño de tableros de juegos didácticos.



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

Autora: Hanny Valdés Hernández

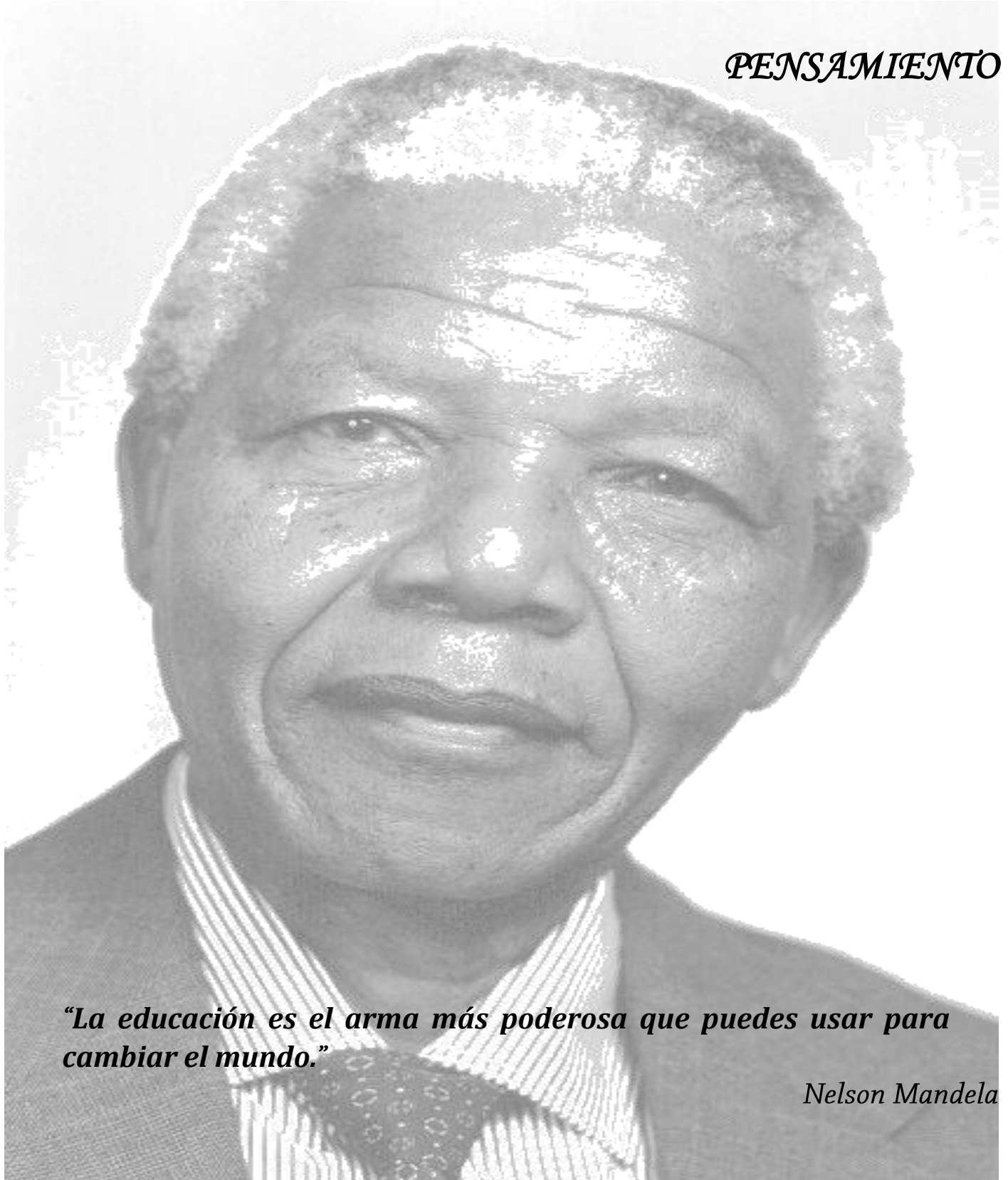
Tutores: Ing. Dayris Espinosa Ronquillo
Ing. Jany Coto García

Cotutor: Ing. Damián Cervantes Rodón

La Habana, de junio de 2013
“Año 55 de la Revolución”



PENSAMIENTO



“La educación es el arma más poderosa que puedes usar para cambiar el mundo.”

Nelson Mandela



AGRADECIMIENTOS

Quiero dejar fe de mis más sentidos agradecimientos a:

Mis padres: por haberme dado todo el apoyo, el amor, el cariño y la fuerza para alcanzar este sueño, incluso si eso significaba estar lejos de ellos.

A mis abuelos: por su dedicación, preocupación, y por mantener siempre su confianza en mí.

A mis hermanos: por ser dos de las personas que más quiero en este mundo y que han estado siempre ahí para apoyarme.

A toda mi familia: por confiar en mí y tenerme presente.

A mi novio: esa persona incondicional que siempre me dio su apoyo y la fuerza para seguir adelante.

A mis profesores: a todos y cada uno de ellos que a lo largo de estos cinco años pusieron su empeño en transmitirnos sus conocimientos, experiencias y que tanto esfuerzo dedicaron para lograr el éxito final.

A mis tutoras: que me guiaron certeramente en la realización y presentación de este trabajo.

A mis compañeros de clase: a todos y cada uno de ellos con los que tuve el placer de compartir clase por todas las cosas que aportaron a mi vida y el lugar que se han ganado en ella.

A mis chicos FEU: esos que noche tras noche no descansan por hacer de esta facultad la mejor cada día y que considero como mi segunda familia por todo el tiempo y las experiencias compartidas junto a ellos

A todos los que directa e indirectamente ayudaron para que el presente trabajo saliera adelante, en especial a Pablo y Manuel.

A todos muchas Gracias...



DEDICATORIA

Quiero dedicar esta tesis a dos personas que son lo más grande que tengo en esta vida, por las que me levanto todos los días, miro hacia el futuro y emprendo un nuevo reto. Porque son mi luz y guía, mi modelo a seguir, en las que siempre encontré apoyo incondicional, amor, cariño y un sabio consejo que agradezco y recuerdo siempre. Con mucho cariño y respeto para mis padres.



RESUMEN

En la Universidad de las Ciencias Informáticas (UCI) se utilizan de manera generalizada las Tecnologías de la Información y las Comunicaciones (TIC) como apoyo al Proceso Enseñanza-Aprendizaje (PEA) de las asignaturas que integran la carrera Ingeniería en Ciencias Informáticas. Habitualmente estas tecnologías se emplean en las clases y auto-preparación de estudiantes y profesores, sin embargo, pudieran ser mejor aprovechadas a partir del diseño de recursos educativos para la disciplina Técnicas de Programación de Computadoras (TPC).

En el marco de un Proyecto de Innovación-Investigación para las asignaturas de Programación se ha desarrollado una herramienta de autor que le permite al profesor el diseño de tableros, lo cual facilita la creación de juegos didácticos para la ejercitación de contenidos. En la presente investigación se exponen los principales artefactos generados durante el desarrollo de la propuesta de solución, así como los resultados obtenidos a partir de las pruebas realizadas al sistema que verifican su correcto funcionamiento.

Palabras claves: diseño, herramienta de autor, juegos didácticos, PEA, recursos educativos.



ABSTRACT

At the University of Information Sciences (UCI) is widely used by the Information Technology and Communications (ICT) to support the Teaching-Learning Process (SAP) of the subjects that make up the race in Computer Science Engineering. Usually these technologies are used in classes and self-grooming of students and teachers, however, could be better exploited through the design of educational resources for the discipline of Computer Programming Techniques (TPC).

As part of Project Innovation-Research for programming courses has developed an authoring tool that allows the teacher to design boards, which facilitates the creation of educational games for the exercise of contents. In the present study presents the main artifacts generated during the development of the proposed solution, and the results obtained from the system tests that verify correct operation.

Keywords: authoring tools, design, educational resources, learning games, PEA.



ÍNDICE DE CONTENIDO

Introducción	1
Capítulo I. Fundamentación teórica de la herramienta de autor para el diseño de tableros de juegos didácticos.	6
1.1 Introducción	6
1.2 Conceptos relacionados con el problema de investigación:	6
1.3 Antecedente de la investigación	8
1.4 Las herramientas de autor, su aplicación en Cuba y el mundo	9
1.5 Marco Teórico-Conceptual de la Investigación	12
1.6 Tendencias actuales a considerar en el desarrollo de software.....	13
1.7 Metodología, lenguaje y herramienta de modelado.....	14
1.8 Herramientas, tecnologías y lenguajes para el desarrollo	17
1.9 Conclusiones parciales	25
Capítulo II: Descripción de la herramienta DEGames Board para el diseño de tableros de juegos didácticos.	26
2.1 Introducción	26
2.2 Características de la herramienta de autor DEGames Board.....	26
2.3 Levantamiento de requisitos	28
2.4 Modelado del sistema	31
2.5 Arquitectura de software	34
2.6 Patrones de diseño	36
2.7 Diagrama de Clases del Diseño.....	37
2.8 Conclusiones parciales	37
Capítulo III: Implementación y pruebas.	38
3.1 Introducción	38
3.2 Modelo de Despliegue	38
3.3 Modelo de Implementación	39
3.4 Almacenamiento de datos	40
3.5 Estándares de codificación	40
3.6 Pruebas	41
3.7 Conclusiones parciales	49
Conclusiones	50



Recomendaciones y trabajo futuro	51
Referencias Bibliográficas	52
Bibliografía Consultada	55
Glosario de términos	63
Anexos.....	69
Anexo #1. Tipos de Juegos Didácticos.	69
Anexo #2. Descripción detallada de los Casos de Uso del sistema.	70
Anexo #3. Diagrama de clases del diseño.....	86
Anexo #4. Estructura del fichero XML.	86
Anexo #5. Ejecución de los casos de prueba.	87
Anexo #6. Validación de la herramienta de autor.....	91



ÍNDICE DE ILUSTRACIONES

Ilustración 1: Fases, iteraciones y flujos correspondientes a la metodología de software RUP.....	15
Ilustración 2: Modelo de dominio de la Herramienta de autor DEGames Board.	27
Ilustración 3: Diagrama CUs diseño de tableros de Juegos.	32
Ilustración 4: Diagrama de despliegue de la herramienta DEGames Board.	38
Ilustración 5: Diagrama de componente.	39
Ilustración 7: Comportamiento por iteraciones de las pruebas funcionales.	49
Ilustración 8: Diagrama de clases de la herramienta DEGames Board.	86



ÍNDICE DE TABLAS

Tabla 1: Listado de requerimientos funcionales de la herramienta de autor DEGames Board.....	29
Tabla 2: Usuarios del sistema y su descripción.....	31
Tabla 3: CUs Gestionar Tablero.....	32
Tabla 4: CUs Gestionar casilla.....	32
Tabla 5: CUs Gestionar ficha.....	33
Tabla 6: CUs Gestionar texto.....	33
Tabla 7: CUs Insertar imagen.....	33
Tabla 8: CUs Eliminar imagen.....	34
Tabla 9: CUs Insertar dado.....	34
Tabla 10: Caso de Prueba Gestionar tablero.....	42
Tabla 11: Caso de Prueba Gestionar casilla.....	44
Tabla 12: Caso de Prueba Gestionar ficha.....	45
Tabla 13: Caso de Prueba Gestionar texto.....	46
Tabla 14: Caso de Prueba Insertar imagen.....	47
Tabla 15: Caso de Prueba Eliminar imagen.....	47
Tabla 16: Caso de Prueba Insertar dado.....	48
Tabla 17: Descripción de variables.....	48
Tabla 19: Descripción detallada del CUs Gestionar tablero.....	70
Tabla 20: Descripción detallada del CUs Gestionar casilla.....	74
Tabla 21: Descripción detallada del CUs Gestionar ficha.....	79
Tabla 22: Descripción detallada del CUs Gestionar texto.....	81
Tabla 23: Descripción detallada del CUs Insertar imagen.....	83
Tabla 24: Descripción detallada del CUs Eliminar imagen.....	84
Tabla 25: Descripción detallada del CUs Insertar dado.....	84
Tabla 26: SC1_Crear_tablero_CU1.....	87
Tabla 27: SC2_Cargar_tablero_CU1.....	87
Tabla 28: SC3_Guardar_tablero_CU1.....	87
Tabla 29: SC4_Modificar_tablero_CU1.....	88
Tabla 30: SC1_Avanzar_CU2.....	88



Tabla 31: SC2_Obstáculo_CU2.	88
Tabla 32: SC3_Didáctica_CU2.....	88
Tabla 33: SC4_Modificar_casilla_CU2.	89
Tabla 34: SC5_Eliminar_casilla_CU2.....	89
Tabla 35: SC1_Insertar_ficha_CU3.	89
Tabla 36: SC2_Modificar_ficha_CU3.....	89
Tabla 37: SC3_Eliminar_ficha_CU3.	90
Tabla 38: SC1_Insertar_texto_CU4.....	90
Tabla 39: SC2_Modificar_texto_CU4.	90
Tabla 40: SC3_Eliminar_texto_CU4.....	90
Tabla 41: SC1_Insertar_imagen_CU5.....	90
Tabla 42: SC1_Eliminar_imagen_CU6.....	91
Tabla 43: SC1_Insertar_dado_CU7.....	91
Tabla 18: Caracterización de los expertos.....	91



Introducción

Los sistemas educativos siempre se han caracterizado por su rigidez en la transmisión del conocimiento como una herencia de la educación tradicional, propuesta que fue válida y para nada condenable en su época. Con el devenir de los años, el avance de las tecnologías y el desarrollo del software educativo como herramienta de enseñanza, ha traído consigo el desarrollo de espacios virtuales de intercambio en tiempo real, y hoy es imprescindible reevaluar y convertir el Proceso Enseñanza- Aprendizaje (PEA) en un encuentro con el saber más dinámico, participativo y menos agresivo con el educando **(1)**.

Encaminando a los sistemas de enseñanza a sumarse a este desarrollo, dado al impacto que estaba obteniendo a nivel mundial el empleo de la Red de Redes (Internet) y el acelerado desarrollo de las Tecnologías de la Información y la Comunicación (TIC) de manera particular, ofreciendo un nuevo espacio de innovación en todos los aspectos de la vida laboral y personal, con una gran presencia en **La Educación**. En la actualidad para la aplicación de esta novedosa tecnología ha tenido lugar el perfeccionamiento del software educativo y la búsqueda continua por parte de los innovadores para la mejora de una comunicación bidireccional entre profesor y alumno.

Los centros educacionales del mundo se han sumado también a la tarea de establecer un PEA que admita desde el punto de vista de la recreación educativa, los juegos como herramienta para favorecer el aprendizaje, incrementando la conexión entre la actividad docente y profesional de los estudiantes y validando su utilización para contrarrestar la falta de interés hacia el estudio consecuente. Es importante resaltar que estudios desarrollados proyectan que, a medida que aumenta la edad de los estudiantes y el nivel de enseñanza que cursan, disminuye la utilización de los juegos didácticos, aún cuando el “(...) *juego puede tener un papel importante como método productivo de enseñanza que contribuye a la activación del aprendizaje*” **(2)**.

Cuba, como el resto de los países se integra también a esta naciente revolución tecnológica y a la búsqueda de transformaciones educacionales que le permitan emplear variantes de aprendizaje con el aprovechamiento pleno de las nuevas tecnologías puestas en función de la calidad del PEA. La Universidad de las Ciencias Informáticas (UCI) no queda exenta de este proceso, por lo que potencia desde sus inicios el uso de las TIC. En el año 2005 se comienza a desarrollar el Entorno Virtual de



Aprendizaje (EVA) como espacio para que los estudiantes y profesores que acceden a este, incorporen habilidades y conocimientos mediante sistemas telemáticos.

Al EVA se integran todas las asignaturas del plan de estudio de la carrera, entre las que se encuentran las que pertenecen al núcleo básico-específico de la especialidad en la universidad, donde se destacan las relativas a la disciplina de Técnicas de Programación de Computadoras (TPC). Estas asignaturas son de difícil comprensión, por lo que los estudiantes deben realizar un estudio sistemático de las mismas. Sin embargo, se evidencia que algunos estudiantes manifiestan desmotivación y desinterés por el estudio y obtienen bajos resultados académicos.

Con vista a mejorar lo antes planteado, la Facultad 1 desarrolló en el 2011 la aplicación de escritorio SMProg¹, para integrar conocimiento y recreación de manera tal que el estudiante, lejos de sentir la necesidad de aprender la asignatura como un método impositivo, lo asumiera como una forma creativa y entretenida potenciando un aprendizaje desarrollador. Hasta ese momento fue concebida para darle un vuelco a los métodos de aprendizaje ya existentes, pues brinda la posibilidad de diseñar tableros y evaluar a través de juegos didácticos los conocimientos adquiridos en clases. Sin embargo, al tener un despliegue específico:

- ♣ Fue considerada cada vez más obsoleta, debido el trabajo engorroso que significaba instalar, configurar y darle mantenimiento para todo aquel que necesitaba su uso.
- ♣ No permite el acceso concurrente de profesores y estudiantes.
- ♣ Los tipos de problemas que genera para la ejercitación de los contenidos no permiten el desarrollo de algoritmos, ya que están basados en sistemas de preguntas de bajo nivel.

Ante tal situación el **Problema de investigación** radica en: ¿Cómo facilitar el diseño de recursos educativos para la ejercitación de contenidos en la disciplina Técnicas de Programación de Computadoras de la Universidad de las Ciencias Informáticas?

¹ Software de Motivación para la Programación.



El **Objeto de estudio** de la investigación lo constituye el proceso de desarrollo de aplicaciones web para la educación, delimitando el **Campo de acción** a las herramientas de autor para crear recursos educativos.

El **Objetivo general** de esta investigación está centrado en: Desarrollar una herramienta de autor que permita la creación de tableros de juegos didácticos para la ejercitación de contenidos en la disciplina Técnicas de Programación de Computadoras de la Universidad de las Ciencias Informáticas.

Y como **Objetivos específicos**:

1. Construir el marco teórico de la investigación relacionado con el desarrollo de sistemas que incluyan creación de tableros de juegos didácticos para la ejercitación de contenidos en la Web.
2. Desarrollar una arquitectura que facilite la creación de tableros de juegos didácticos en la Web.
3. Verificar el correcto funcionamiento de la herramienta de autor para el diseño de tableros de juegos didácticos.

Y para darle respuesta a los objetivos específicos es necesaria la realización de las siguientes **Tareas investigativas**:

1. Caracterización del estado del arte acerca de los sistemas que incluyan tableros de juegos didácticos para la ejercitación de contenidos.
2. Fundamentación de las tecnologías a utilizar en el desarrollo de la herramienta.
3. Identificación de los recursos informáticos a desarrollar para la implementación de la herramienta.
4. Definición de la arquitectura de la herramienta.
5. Desarrollo de las funcionalidades requeridas para lograr la creación de tableros de juegos didácticos en la Web.
6. Realización de las pruebas correspondientes a la herramienta.



Los **Métodos científicos** que se utilizaron durante la investigación son:

Métodos Teóricos

✎ **Histórico-lógico:** para realizar un estudio de la trayectoria histórica, evolución y desarrollo sobre los proyectos informáticos de este tipo (Herramientas de autor para juegos didácticos) en el ámbito nacional e internacional y en específico la ya existente (Herramienta educativa SoftMotProg para el EVA MOODLE).

✎ **Analítico-sintético:** para analizar de las características de los diferentes tipos de juegos con tableros, los elementos más importantes que permitan lograr un correcto diseño de la aplicación.

Métodos Empíricos

✎ **Observación:** se realizó un estudio de las características del sistema existente, seleccionando las más útiles para la herramienta a desarrollar.

✎ **Modelación:** para representar el objeto de estudio y la posible solución del problema científico. Con este método se realizó el diseño de la herramienta a implementar.

Justificación de la Investigación

La presente investigación propone como aporte práctico la elaboración de una herramienta de autor que permita a los profesores de la disciplina Técnicas de Programación de Computadoras, la creación de tableros de juegos didácticos en la Web y la asociación de estos de una forma interactiva a preguntas de un tema determinado de dicha disciplina, permitiéndole al estudiante ejercitar los contenidos recibidos en clase. Por tanto, se supone que la utilización de una herramienta como esta contribuya con el PEA de la disciplina de Técnicas de Programación de Computadoras en la universidad, ya que constituye una propuesta novedosa en este nivel de enseñanza.

El trabajo que se presenta está estructurado en tres **Capítulos**, los cuales se despliegan a continuación:



Capítulo I. Fundamentación teórica de la herramienta de autor para el diseño de tableros de juegos

didácticos: Este capítulo presenta algunos elementos teóricos necesarios que ofrecen soporte para la realización de todo el trabajo en general, tales como: estado del arte, la descripción y selección de las herramientas y metodologías a utilizar.

Capítulo II. Descripción de la herramienta DEGames Board para el diseño de tableros de juegos

didácticos: En este capítulo se presenta la propuesta de solución a partir del levantamiento de los requisitos funcionales y no funcionales. Se muestran además cada uno de los artefactos generados como diagrama de casos de uso y diagrama de clases.

Capítulo III. Implementación y pruebas:

En este capítulo se describe cómo está implementada la herramienta, a través de los Diagramas de Componentes y el Diagrama de Despliegue. Se diseña un plan de pruebas con el propósito de definir el alcance de la misma y se verifica el funcionamiento de la solución propuesta.

El presente trabajo contiene además **Conclusiones, Recomendaciones, Referencias Bibliográficas, Bibliografía, Glosario de Términos y Anexos.**



Capítulo I. Fundamentación teórica de la herramienta de autor para el diseño de tableros de juegos didácticos.

1.1 Introducción

La científicidad de la educación presenta gran apoyo en la realización en los juegos con dirección, orientación metodológica y sistematicidad para activar el pensamiento rápido y fuerte, unido a la actividad práctica con vista a desarrollar aún más las capacidades intelectuales de los educandos.

En el presente capítulo se describen los elementos fundamentales que constituyen el soporte teórico de la investigación y se realiza un análisis de los elementos importantes relacionados con la homologación de los sistemas informáticos ya existentes tanto en el ámbito nacional como internacional. Se valoran además las tendencias actuales en el contexto informático y se identifican los elementos clave de las herramientas y tecnologías a utilizar para el desarrollo de la propuesta de solución con el fin de cumplir los requisitos técnicos y funcionales.

1.2 Conceptos relacionados con el problema de investigación:

En este punto se exponen los conceptos fundamentales afines con el problema en cuestión como factor importante para lograr una mejor comprensión del mismo.

Herramienta de autor

Existen incontables tipos de herramientas de apoyo al *e-learning*, entre ellas los repositorios de objetos de aprendizaje y las herramientas de autor. Las segundas pueden clasificarse desde disímiles perspectivas dada la variedad de criterios que existen en cuanto a su definición, como son:

Una herramienta de autor es un programa de ordenador que facilita la creación de material educativo a profesores no especializados en informática (“no-programadores”). En cierta manera, evita la complejidad de la programación tradicional y permite la creación de “lecciones electrónicas” a cualquier instructor interesado y que esté dispuesto a dedicar horas de su tiempo para actualizar las herramientas didácticas con sus conocimientos (3).



Las herramientas de autor son aplicaciones que permiten la creación de diferentes contenidos educativos. Estas posibilitan al usuario realizar un producto con fines educativos y generan un entorno de aprendizaje dinámico. Los objetivos se pueden adecuar según los conocimientos y las habilidades que el autor desee desarrollar. Tienen la posibilidad de cambiar el flujo de la información según las necesidades del alumno, relacionar palabras, incluir cuestionarios y además, marcadores que evalúan el conocimiento que el estudiante vaya adquiriendo (4).

Estas herramientas de autor son responsables de determinadas tareas y están formadas por distintos elementos; donde el número y características de estos varían según la herramienta seleccionada, las que en su mayoría proporcionan las siguientes prestaciones:

- ♣ **Procesadores de texto:** permiten escribir tanto el contenido (los datos e informaciones) como la programación (instrucciones de funcionamiento) de las aplicaciones.
- ♣ **Gestores de bases de datos:** herramienta para organizar la información y permitir su posterior consulta y utilización.
- ♣ **Programas de edición de video:** hacen posible la digitalización de imágenes y su edición (“montaje”), incluyendo la aplicación de efectos de video (“postproducción”).
- ♣ **Programas de edición y generación de sonido.**
- ♣ **Programas de gráficos y dibujo para imagen fija. Ilustración (3D):** modelado y renderizado de objetos.
- ♣ **Programas de animación.**

Las principales ventajas que estas herramientas pueden proporcionar son:

- ♣ Al ser diseñados para un propósito específico, muchas de las necesidades más habituales de los creadores de software educativo han sido previstas de antemano y son fáciles de implantar.
- ♣ Pueden ser utilizados para aplicaciones con gran variedad de contenidos.
- ♣ Mejora la calidad de enseñanza por parte del docente y alumno.



- ♣ Ayuda a la valoración profesional del docente.
- ♣ Mayor posibilidad de trabajo y posibles nuevas fuentes de ingresos.
- ♣ Actualización de los métodos pedagógicos tanto en lo individual como la parte grupal.

Además, muchas de las herramientas de autor disponibles en el mercado actualmente son multiplataforma, lo que las hace capaz de funcionar con distintos tipos de sistemas operativos y ordenadores, facilitando su utilización en prácticamente todas las circunstancias **(3)**.

1.3 Antecedente de la investigación

En la Universidad de las Ciencias Informáticas (UCI) se desarrolló en el 2011 la herramienta educativa SMProg (Software – Motivación - Programación) que le permite 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 equivoque el estudiante, el propio software mostrará un mayor número de preguntas de ese tipo y realizará la evaluación del mismo.

Esta herramienta también permite al profesor realizar el diseño del juego y su vinculación con el sistema de preguntas previamente definido, así como la interacción del estudiante con la aplicación. En la primera versión del software se desarrollan tres módulos diferentes estrechamente vinculados entre sí y con similar ambiente, entre ellos se encuentra:

Módulo diseño de tableros (DesBoard): Herramienta creada para la modelación por parte del profesor del tablero correspondiente de acuerdo al tipo de juego a jugar, ya sea Avanza hasta la Meta, Bingo o Sube y Baja. 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.

El diseño de tableros 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 tiene lugar la Implementación de juegos de tableros que permite a los estudiantes jugar y aprender (5).

1.4 Las herramientas de autor, su aplicación en Cuba y el mundo

Actualmente, existen varias decenas de programas o herramientas de autor de diversos tipos y prestaciones en el mercado y su número sigue incrementándose, por lo que resulta difícil realizar un inventario exhaustivo y actualizado. Durante la investigación se analizaron algunos ejemplos de herramientas de autor, de ellos fueron seleccionados de forma aleatoria los más conocidos en el mundo de la educación.

1.4.1 Ámbito Internacional

JClic: es un proyecto de código abierto formado por un conjunto de aplicaciones informáticas que sirven para realizar actividades educativas. Está desarrollado sobre la plataforma Java por lo que funciona en diversos entornos y sistemas operativos. El formato para almacenar los datos de las actividades es XML (6). Puede integrarse con MOODLE² a partir de la previa instalación del módulo correspondiente y de la activación de la actividad JClic que se desee utilizar. Su alcance es para el nivel primario y secundario ya que las actividades que se pueden diseñar son del tipo de rompecabezas, asociaciones, ejercicios de texto y palabras cruzadas.

Hot Potatoes: herramienta de autor desarrollada por la Universidad de Victoria en Estados Unidos que permite elaborar ejercicios interactivos de cinco tipos:

JBC: ejercicios de elección múltiple.

JCLOZE: ejercicios de rellenar huecos.

JCROSS: crea crucigramas.

JMATCH: crea ejercicios de emparejamiento u ordenación.

JMIX: ejercicios de reconstrucción de párrafos o frases (3).

² Por sus siglas en inglés significa *Modular Object-Oriented Dynamic Learning Environment* (Entorno de Aprendizaje Dinámico Modular Orientado a Objetos)



Ardora: Es una aplicación informática para docentes que permite crear actividades educativas en formato HTML³ para sus alumnos de un modo muy sencillo, permite crear más de 45 tipos distintos de actividades (crucigramas, sopas de letras, completar, paneles gráficos, entre otros), así como más de 10 tipos distintos de páginas multimedia (galerías de imágenes, reproductores, entre otros).

Entre las características de este tipo de aplicación se encuentran:

- ♣ Capacidad de brindarles la posibilidad a los profesores de configurar las actividades como ellos estimen conveniente.
- ♣ La forma de configurar las actividades se realiza mediante formularios muy sencillos.
- ♣ Permite a los profesores centrar su esfuerzo en los elementos de la actividad no en su tratamiento informático.
- ♣ Fácil de usar.
- ♣ Rapidez a la hora de crear actividades docentes educativas.
- ♣ Permite guardar, publicar y hacer uso previo de actividades educativas.
- ♣ Permite guardar la actividad en formato HTML.

Aún cuando Ardora no cuenta con un diseño atractivo, hay que destacar que sus características en función de la creación de actividades docentes educativas la convierten en una herramienta adecuada para su propósito **(7)**.

1.4.2 Ámbito Nacional

Colección Multisaber: Es una colección de software educativo compuesta por varios productos donde cada uno se encuentra integrado por diversos módulos. El módulo Juegos está compuesto por 19 juegos, los cuales fueron diseñados específicamente para apoyar la enseñanza y el aprendizaje en los alumnos de la enseñanza primaria de Cuba. Sin embargo, hay que señalar que aun cuando resultan atractivos para

³ Siglas de Hyper Text Markup Language («lenguaje de marcado de hipertexto»).



aquellos a quienes están dirigidos, los mismos presentan varias deficiencias que le restan dinamismo y flexibilidad.

Actualmente, los juegos no les brindan la posibilidad a los profesores de ejercer una labor más activa sobre el proceso de enseñanza-aprendizaje de los alumnos, ya que su diseño está enfocado específicamente en los estudiantes, sin tener en cuenta las ventajas que puede traer la participación del profesor en todo este ambiente de aprendizaje. Otras de las dificultades, es que los juegos presentan comportamientos comunes para todos los alumnos, por no se tiene implementado una forma que permita flexibilizar los juegos haciendo que estos aumenten o disminuyan su complejidad **(8)**.

HAEduc: Es una herramienta de autor multiplataforma, que surge producto de la necesidad de utilizar herramientas libres para elaborar recursos educativos en los ambientes Windows y Linux, desarrollada por el Centro de Estudios de Software Educativos (CENSOE), de la Universidad de Ciencias Pedagógicas “Rubén Martínez Villena”, de La Habana, en conjunto con un equipo del Ministerio del Poder Popular para la Educación de Venezuela, en el marco del Convenio Colaboración Integral entre Cuba y Venezuela.

La herramienta de autor Haeduc sigue el camino de desarrollo de la Web, sin embargo, está desarrollada solo para colecciones de multimedia y dirigida especialmente a la Educación Venezolana **(9)**.

1.4.3 Las herramientas de autor en la Universidad de las Ciencias Informáticas

En la Universidad de las Ciencias Informáticas se utilizan aplicaciones e-learning como apoyo al proceso docente, entre ellas se encuentran el Repositorio de Objetos de Aprendizaje RHODA, la Herramienta de autor para la Creación de Objetos de Aprendizaje de forma colaborativa CRODA, y la plataforma de Gestión del aprendizaje EVA.

La herramienta de autor CRODA: Esta herramienta de autor Web permite la creación de Objetos de Aprendizaje reutilizables, accesibles, duraderos e interoperables empleando el estándar SCORM⁴. Con vistas a reutilizar los OA⁵, todos sus elementos pueden ser descritos con metadatos siguiendo el estándar

⁴ Modelo de Referencia de Objetos de Contenido Compartible: del inglés Sharable Content Object Reference Model.

⁵ Siglas para denotar el Objeto de Aprendizaje



LOM⁶. En CRODA los creadores tienen la posibilidad de apoyarse en otros docentes y técnicos para la obtención de la información y conocimientos durante la creación de su OA, lo cual constituye la base para el diseño de un entorno colaborativo.

A continuación algunas de sus características y funcionalidades:

- ♣ Presenta una interfaz agradable, de fácil y cómodo uso.
- ♣ Permite el acceso a diferentes servicios manteniendo una simplicidad en su arquitectura.
- ♣ Brinda la posibilidad de crear plantillas: estas son solamente una estructura inicial para la creación de algún OA. Siendo posible además, luego de haber creado dichas plantillas, hacerlas públicas para que otras personas que interactúen con el sitio, luego de haber sido analizadas y revisadas por un especialista.
- ♣ Cuando se crea algún OA, dicha herramienta brinda la posibilidad de modificar la estructura inicial de las plantillas, permitiendo incluir nueva información y contenido.
- ♣ A los OA que son creados, se les incorporan metadatos que describen algunos de sus elementos facilitando así la reusabilidad e identificación en otras herramientas.
- ♣ La incorporación de estos metadatos se implementa con el uso del estándar LOM.

Esta herramienta de autor presenta comunicación con un Repositorio de Objetos de Aprendizaje (ROA), brindándole la posibilidad de guardar los recursos educativos una vez creados. En dicho repositorio se pueden realizar búsquedas con diferentes criterios, con el fin de pasar al área de creación aquel OA al cual se le desee modificar su contenido **(10)**.

1.5 Marco Teórico-Conceptual de la Investigación

Después de realizar un estudio de las herramientas de autor utilizadas en los ámbitos analizados, se pudo apreciar que, a pesar de las amplias posibilidades que brindan en cuanto al desarrollo de recursos educativos, en su generalidad fueron diseñadas según las particularidades del nivel de enseñanza donde son aplicados.

⁶ Metadatos de los objetos de aprendizaje: del inglés Learning Objects Metadata.



En el caso de las herramientas JClic, Hot Potatoes, Colección Multisaber y Ardora, se puede señalar que son de código abierto, que están desarrolladas en la Web y que permiten la interacción con el profesor. También están destinadas a crear tipos de juegos específicos que en su generalidad incluyen la creación de multimedias para los niveles de enseñanza primaria y secundaria, lo que no se adecúa a las necesidades de la herramienta que se desea crear pues, aunque están bien diseñadas para estos niveles de enseñanza, para utilizarlas en el nivel universitario se requiere de un alcance mayor.

Otras como HAEduc se limitan a la creación de colecciones de multimedia y están destinadas a la enseñanza venezolana. Y CRODA que es la única que va dirigida al nivel superior se especializa solo en la creación de objetos de aprendizaje reutilizables, lo que trae consigo señalar que estas herramientas de autor no son capaces de cumplir con la necesidad de obtener una herramienta libre para desarrollar juegos didácticos tipo tableros en la Web, que sean adaptables a cualquier asignatura de la carrera y a la incorporación de nuevos tipos de juegos.

1.6 Tendencias actuales a considerar en el desarrollo de software

La presente investigación se propone como alcance garantizar, a través de la implementación de la herramienta de autor para el diseño de tableros, la ejercitación de los contenidos o conocimientos a través de un juego didáctico en la Web.

Para el desarrollo de dicha aplicación se realizó un estudio sobre las principales herramientas a utilizar, así como de las metodologías a ser empleadas durante su elaboración. Analizando las tendencias actuales y las novedades existentes en el campo del desarrollo, se llega a las siguientes propuestas sobre las metodologías, lenguajes de programación y otras herramientas que serán empleadas durante su desarrollo.

“El software libre, es un movimiento tecnológico que ha revolucionado la sociedad. Presenta características especiales que han permitido la experimentación de nuevas formas de desarrollo y mantenimiento de programas, nuevos modelos económicos, y nuevas normas legales. Es un asunto de libertad, no de precio. Para entender el concepto, se debe pensar en “libre” como en “libertad de expresión” (11).



1.7 Metodología, lenguaje y herramienta de modelado

1.7.1 Metodología de Desarrollo de Software

Una metodología de desarrollo de software es un conjunto de pasos y procedimientos que deben seguirse para desarrollar un software con calidad. Define las etapas para el desarrollo de un software, las tareas que se deben llevar a cabo en cada etapa, las restricciones que deben aplicarse, las técnicas y herramientas que deben emplearse, y la forma de controlar y gestionar un proyecto.

RUP

El Proceso Unificado de Desarrollo de Software (en inglés: *Rational Unified Process*) es una metodología de desarrollo de software robusta, la cual constituye entre todas las metodologías estándares la más utilizada para el diseño, implementación y documentación de sistemas orientados a objetos. Se encuentra preparada para el desarrollo de grandes y complejos proyectos, representa un ideal de referencia para todo el equipo de desarrollo.

Es además, un proceso que contiene un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software, se puede especializar para gran variedad de sistemas, distintas áreas de aplicación, tipos de organizaciones, niveles de actitud y tamaños de proyecto. Cuenta con un ciclo de vida que se caracteriza por ser **(12)**: dirigido por casos de uso, centrado en la arquitectura e iterativo – incremental.

No posee pasos que estén firmemente establecidos, sino que cuenta con un conjunto de flujos de trabajo adaptables al contexto y necesidades de cada organización, lo que la hace perfecta para emplear como guía de desarrollo de la solución propuesta, por las peculiaridades que presenta la misma. De ahí que solo se generen los artefactos necesarios como: levantamiento de requisitos, los diagramas y descripciones de casos de uso, la descripción de la arquitectura, los diagramas de clases del diseño, el diagrama de componentes, de despliegue y los casos de prueba.

La siguiente ilustración muestra cómo las fases de RUP concluyen con un hito bien definido y en cada uno de estos intervienen los flujos de trabajo correspondientes, garantizando el cumplimiento de los objetivos y metas antes de la transición a la nueva fase.

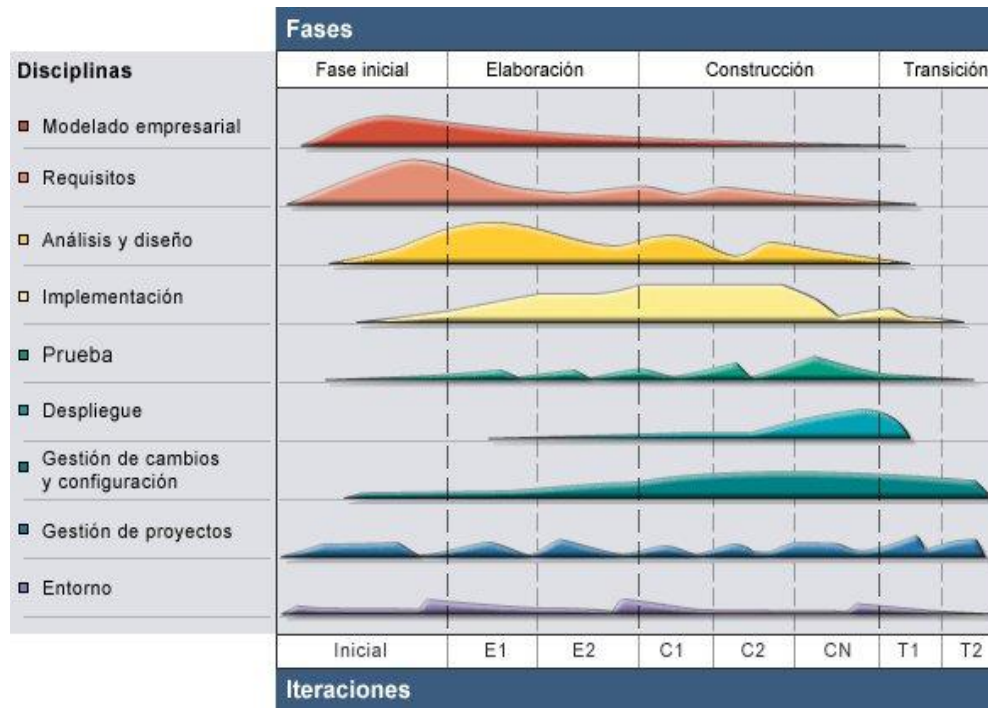


Ilustración 1: Fases, iteraciones y flujos correspondientes a la metodología de software RUP.

Fuente: <http://bannysolano.wordpress.com/2010/02/09/rup-en-espanol/>

1.7.2 Herramienta CASE y lenguaje para el modelado

Las herramientas CASE (**C**omputer **A**ided **S**oftware **E**ngineering, por sus siglas en inglés, o Ingeniería de Software Asistida por Ordenador) son aplicaciones informáticas con el objetivo de aumentar la productividad y la eficiencia en el desarrollo de software, minimizando de esta forma el costo en términos de tiempo y dinero.

La misión de cualquier herramienta CASE que utiliza el Lenguaje Unificado de Modelado (UML) como notación para elaborar los modelos, es comunicar, de la manera más eficiente posible a todos los agentes del proyecto, todas aquellas decisiones que se toman con respecto a la arquitectura del sistema en discusión y que son determinantes para cumplir con los objetivos del proyecto. Establecer una arquitectura racional y eficiente, acorde a las necesidades de los actores, es la razón fundamental para invertir tiempo y esfuerzos en una herramienta CASE que utiliza la notación UML.



Visual Paradigm

Herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas que sigue el estándar de UML (Diagramas de clase, Casos de uso, Comunicación, Secuencia, Estado, Actividad, Componentes). También proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML. Presenta licencia gratuita y comercial, es fácil de instalar y actualizar, compatible entre ediciones y admite ingeniería de ida y vuelta. Además, permite generación de código, documentación y de bases de datos **(13)**.

UML

El Lenguaje Unificado de Modelado más conocido por todos como UML (por sus siglas en inglés *Unified Modeling Language*), es un lenguaje utilizado para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software **(14)**, es el más usado ya que ofrece un estándar para describir un plano o modelo del sistema en el cual se incluyen los conceptos de proceso de negocio así como funciones del sistema.

UML en su funcionamiento emplea una serie de diagramas **(15)** que son fundamentales para el correcto uso del mismo, los cuáles son: de estructura estática (clases, objetos y casos de uso), de comportamiento (interacción, estado, actividad) y de implementación (componentes, despliegue).

1.7.3 Fundamentación de la herramienta y lenguaje de modelado a utilizar

Para el modelado fue seleccionado Visual Paradigm, que tiene características gráficas muy cómodas, que facilitan la realización de los diagramas de modelado que siguen el estándar de UML.

También posee cualidades como la generación automática de diagramas a partir de descripciones de casos de usos, permitiendo la agilidad en el trabajo del analista. Además usa un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación. Como lenguaje a utilizar para el modelado se escogió UML, ya que este permite visualizar, construir y documentar los artefactos de un proyecto de software. Además es el lenguaje que utiliza la herramienta escogida para realizar el modelado.



1.8 Herramientas, tecnologías y lenguajes para el desarrollo

1.8.1 Marco de trabajo

Un marco de trabajo no es más que una estructura de soporte definida mediante la cual otro proyecto de software puede ser desarrollado y organizado. Puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

Son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional. Es el esqueleto sobre el cual varios objetos son integrados para una solución dada. No es más que una base de programación que atiende a sus descendientes (manejado de una forma estructural y/o en cascada) posibilitando cualquier respuesta ante las necesidades de sus miembros, o secciones de una aplicación Web **(16)**.

Spring

Los principales valores de Spring, según Rod Johnson, se pueden resumir en: No es agresivo, provee un modelo consistente de programación, ayuda a promover la reusabilidad de código, facilita el diseño Orientado a Objetos en aplicaciones JEE, permite la extracción de valores de configuración desde el código java a archivos XML o archivos de propiedades, está diseñado a fin de que las aplicaciones lo usen para que las pruebas sean lo más fácil posible, Spring hace de soluciones existentes un uso más fácil, dentro de una arquitectura consistente.

Además de esto, Spring provee en su paquete Spring Web MVC una implementación del patrón MVC(Modelo Vista Controlador), el cual brinda una limpia y clara separación entre las 3 capas arquitectónicas: Acceso a Datos, Negocio y Lógica de Presentación. Esta implementación facilita una serie de clases controladoras, que manejan la lógica de la navegación e interactúan con la capa de Negocio de la aplicación Web **(17)**.



1.8.2 Lenguaje y tecnologías para el desarrollo Web

Un lenguaje de programación es un lenguaje artificial que permite establecer una comunicación con una máquina, especialmente una computadora. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes.

Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. 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 **(18)**.

El lenguaje para la programación en Java es de una plataforma independiente. Los programas en Java generalmente son compilados y luego interpretados por una máquina virtual. Esta última se utiliza como una plataforma de abstracción entre la máquina y el lenguaje permitiendo que se pueda "escribir el programa una vez, y correrlo en cualquier lado". También existen compiladores nativos de Java, tanto comerciales como libres.

JavaScript

La utilización de JavaScript como aplicación del lado del cliente, se debe a que es un lenguaje basado en objetos, utilizado para acceder a objetos en aplicaciones. Este lenguaje se utiliza principalmente integrado en un navegador Web, permitiendo el desarrollo de interfaces de usuario mejoradas y páginas Web dinámicas. Se caracteriza por ser un lenguaje basado en prototipos, con entrada dinámica y con funciones de primera clase. Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas Web. Para interactuar con una página Web se provee al lenguaje JavaScript de una implementación del Modelo de Objetos del Documento (DOM) **(19)**.



HTML

HTML, es el lenguaje de marcado predominante para la construcción de páginas Web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>), también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML **(20)**.

CSS

Son las siglas de *Cascading Style Sheets*, en español Hojas de estilo en Cascada. Es una tecnología que permite crear páginas Web de una manera más exacta. Gracias a este lenguaje se pueden hacer mucho más cosas que antes no se podían hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores, definir estilos en un archivo externo a nuestras páginas **(21)**.

AJAX

El término AJAX se presentó por primera vez en el artículo "Ajax: *A New Approach to Web Applications*" publicado por Jesse James Garrett el 18 de Febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación Web que estaba apareciendo.

"Ajax no es una tecnología en sí misma. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes" **(22)**.

Permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano. Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

JQuery

Es una librería de JavaScript, que facilita la forma de interactuar con los documentos HTML, el acceso a los objetos del DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX



a páginas web. Ofrece una serie de funcionalidades basadas en JavaScript, que posibilita grandes resultados en la menor cantidad de tiempo y espacio posible. La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX **(23)**.

Raphaël

Es una pequeña librería JavaScript para simplificar el trabajo con gráficos vectoriales en la Web. Raphaël utiliza cada objeto gráfico que se crea también como un objeto DOM, por lo que se puede asociar controladores de eventos JavaScript o modificar más adelante. El objetivo de Raphaël es proporcionar un adaptador que va a hacer el dibujo vectorial compatible con varios navegadores de una manera fácil**(24)**.

XML

Extensible Markup Language (XML) es un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium*. Es una simplificación y adaptación del Lenguaje de Marcado Generalizado (SGML) y permite definir la gramática de lenguajes específicos. Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Esta tecnología propone un estándar de datos que puede codificar el contenido, la semántica y los esquemas de una gran variedad de casos, desde los más simples a los más complejos. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

Aquí se busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible. Que la información sea estructurada quiere decir que se compone de partes bien definidas, y que esas partes se componen a su vez de otras partes. Es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil **(25)**.

1.8.3 Sistema Gestor de Bases de Datos

Un Sistema Gestor de Bases de Datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base de Datos, asegurando su integridad, confidencialidad y seguridad. Existen diferentes



tipos de SGBD, pero los más conocidos y utilizados actualmente son PostgreSQL, Oracle, MySQL, entre otros.

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS), debido a que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos. Utiliza un modelo cliente/servidor y el lenguaje SQL para llevar a cabo las búsquedas de información, para garantizar la estabilidad del sistema usa multiprocesos, en vez de multi-hilos. Este sistema de gestión de bases de datos objeto-relacional es de código abierto(12).

Se caracteriza por ser un sistema estable, de alto rendimiento y gran flexibilidad, ya que funciona en la mayoría de los sistemas Unix, puede ser integrado al ambiente Windows, permitiendo de esta manera a los desarrolladores, generar nuevas aplicaciones o mantener las ya existentes (26).

1.8.4 Servidor Web

Un servidor Web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP (*Hyper Text Transfer Protocol*) o el protocolo HTTPS (versión cifrada y autenticada). Por medio de la especificación de la búsqueda el servidor Web buscará una página específica o ejecutará un programa, pero necesariamente, enviará algún resultado sobre la búsqueda. Entre los servidores Web más utilizados se encuentran el Internet Information Server (IIS), el Xitami y el Apache.

Apache TomCat

Como servidor Web de aplicaciones fue escrito en Java que es el lenguaje de programación propuesto para implementar la aplicación. Funciona en cualquier sistema operativo que tenga instalado la Máquina Virtual de Java, es usado como servidor Web autónomo en entornos con alto nivel de tráfico y alta disponibilidad, es un software de código abierto, funciona como un contenedor de *servlets* (objetos que corren dentro del contexto de un servidor de aplicaciones y extienden su funcionalidad), además implementa las especificaciones de los *servlets* y de Java Server Pages (JSP) de Sun Microsystem (27).



1.8.5 Administrador de Base de Datos

El **pgAdmin** es conocido como el máximo exponente de cliente gráfico para PostgreSQL, aparte de que es una herramienta libre y multiplataforma, posee una serie de características que a continuación se mencionan:

1. Puede trabajar con todos los objetos de la Base de Datos, examinar sus propiedades así como realizar tareas administrativas en la misma, (entiéndase por objetos de la Base de Datos a columnas, dominios, funciones, grupos, índices etc.).
2. Cada vez que se realiza una modificación en un objeto escribe las sentencias SQL correspondientes a dicha modificación, lo que lo convierte aparte de una herramienta útil en una herramienta didáctica.
3. Incorpora funcionalidades para realizar consultas, examinar su ejecución y trabajar con los datos. Todas estas características hacen que pgAdmin sea la herramienta que se necesite para el trabajo con PostgreSQL, ya que tanto desde el punto de vista del administrador como de los usuarios, las acciones que se llevarán a cabo a través de este cliente serán de acuerdo a los privilegios definidos en la base de datos para el usuario que ha establecido conexión con ella**(12)**.

1.8.6 Entorno de Desarrollo Integrado

Es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDE pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes.

NetBeans

El Entorno de Desarrollo Integrado **(28)** NetBeans es una herramienta libre y gratuita sin restricciones de uso, licenciada bajo la CDDL(*Common Development and Distribution License*), que en español sería Desarrollo Común y Licencia de Distribución, la herramienta es utilizada por los programadores para escribir, compilar, depurar y ejecutar programas, permite el desarrollo de aplicaciones web. Está escrito en Java pero puede soportar cualquier otro lenguaje de programación, esta herramienta es modular debido a que todas las funciones que posee están provistas de módulos como el de soporte y edición.



Este entorno de desarrollo integrado se caracteriza por:

- ♣ Editor de código sensible al contenido, con soporte para autocompletar el código, coloreado de etiquetas, auto tabulación y uso de abreviaturas para varios lenguajes de programación.
- ♣ Posibilidad de utilizar otras versiones de compiladores depuradores.
- ♣ Creación visual de componentes gráficos.
- ♣ Soporte para JSP, XML, RMI, CORBA, JINI, JDBC y tecnologías *Servlet*.
- ♣ Soporte para Java, XML y lenguajes HTML.
- ♣ Herramientas con asistentes para facilitar la escritura de código **(29)**.

1.8.7 Fundamentación de las herramientas, tecnologías y lenguajes a utilizar para el desarrollo de la herramienta

Entre los proyectos de Innovación – Investigación que lleva a cabo la dirección del Departamento de Programación de la Facultad 1 se encuentra DEGames Board (herramienta de autor para el diseño de tableros juegos didácticos), dicha dirección tiene como objetivo fundamental, estandarizar el uso de las herramientas, tecnologías y lenguajes en los diferentes proyecto que desarrolla.

Para el proceso se propone la utilización como lenguaje de programación Java por dos razones fundamentales:

1. Portabilidad de su plataforma.

La portabilidad del sistema está basada principalmente en que el compilador de Java genera un código binario conocido como bytecode el cual es interpretado por la Máquina Virtual de Java (JVM por sus siglas en inglés: Java Virtual Machine), y esta a su vez se encarga de ejecutar dicho código. Esto permite que un programa escrito en Java pueda ser ejecutado en cualquier ordenador independientemente de la plataforma que este use, ya sea Windows o Linux, ya que la máquina virtual es independiente de la plataforma.



2. Se puede incrustar en otro programa.

Otra de las razones por las cuáles java es uno de los lenguajes más usados entre los desarrolladores es por la condición de poder incrustarse en cualquier otro programa en forma de applets. Un applets es un componente de Java que se ejecuta en el contexto de otro programa, el más común de los ejemplos es la incrustación de estos en los navegadores Web, diversidad de sitios Web trabajan con applets para representar algún contenido dentro del mismo, además el archivo .jar creado luego de la compilación de la aplicación puede ser ejecutado en cualquier contexto ya que también es independiente de la plataforma. Mediante estas funcionalidades o característica de Java es que la aplicación que se deriva de esta investigación será ejecutada.

Se propone la utilización de Spring Framework debido a que es un marco de trabajo de código abierto para el desarrollo de aplicaciones en la plataforma Java. Fue desarrollado por Rod Johnson, quien lo describió por primera vez en su libro “*Expert One-on-One Java EE Design and Development*”. Es el único marco de trabajo que interviene en todas las capas arquitectónicas de una aplicación JEE, además está diseñado para facilitar una flexibilidad arquitectónica.

Como sistema gestor de base de datos se propone la utilización de PostgreSQL, ya que es uno de los gestores de bases de datos relacional orientado a objetos de software libre capaz de manejar una enorme cantidad de datos, permitiendo el acceso simultáneo de un conjunto de usuarios, además brinda seguridad y estabilidad a los mismos y facilita el trabajo con procedimientos almacenados y consultas.

Se propone la utilización de ApacheTomCat como servidor Web de aplicaciones, debido a que el mismo fue escrito en Java y es el lenguaje de programación propuesto para implementar la aplicación, funciona en cualquier sistema operativo que tenga instalado la Máquina Virtual de Java.

Se propone utilizar NetBeans porque ofrece gran flexibilidad entre plataformas, el cumplimiento de UML y la capacidad de administrar la complejidad, ayudan a garantizar que las aplicaciones cumplan con los requerimientos definidos. Además este IDE garantiza que los desarrolladores puedan confiar en una plataforma de desarrollo que integra todas las piezas críticas necesarias para llevar a cabo el desarrollo de aplicaciones.



1.9 Conclusiones parciales

Luego de realizar una minuciosa investigación de las principales características, ventajas y desventajas de cada una de las herramientas e-learning que permiten la creación de recursos educativos en los ámbitos analizados, se pudo concluir que estas herramientas no elaboran tableros de juegos didácticos como recurso educativo. Por lo tanto, no es posible aplicar ninguna de ellas como propuesta de solución, ya que a pesar de las amplias posibilidades que brindan en cuanto al desarrollo de recursos educativos, en su generalidad no son adaptables para el nivel de enseñanza superior.

Se puede concluir también que en la UCI no se emplean herramientas educativas que permitan la creación de tableros de juegos didácticos específicos que faciliten, la ejercitación de contenidos en las asignaturas de Programación. Además los lenguajes y tecnologías definidas para el desarrollo de la herramienta permiten elaborar un producto con calidad y brindan la garantía de contar con las libertades del software libre, una amplia documentación y robusta comunidad de desarrollo.



Capítulo II: Descripción de la herramienta DEGames Board para el diseño de tableros de juegos didácticos.

2.1 Introducción

El presente capítulo cuenta con la característica principal de identificar y describir un diagnóstico asociado al campo de acción del trabajo de diploma, donde se describen elementos del proceso de creación de tableros de juegos didácticos como recurso educativo. Durante su elaboración se definen los requerimientos funcionales y no funcionales, el diagrama de casos de uso con la descripción de cada uno de estos y los roles que intervienen en él. También se describe el diagrama de clases del diseño, que de una manera más clara refleja el funcionamiento de la herramienta.

2.2 Características de la herramienta de autor DEGames Board

Al realizar un análisis sobre el engorroso trabajo que representa el despliegue específico del módulo Diseñador de Tableros de la herramienta SMProg anterior, se propone el desarrollo de una herramienta de autor como parte de un sistema más complejo. El mismo agrupa una serie de herramientas desarrolladas con tecnología Web que permiten la creación de varios tipos de juegos (tablero y 3D), la generación de problemas para la ejercitación de contenidos, realizar reportes y permitir el acceso de profesores y estudiantes de manera concurrente.

Debido a esto, la herramienta de autor DEGames Board propuesta se debe integrar al sistema SMProg en la Web como solución a la limitante planteada. Es una herramienta que le permite a los profesores el diseño del tablero correspondiente de acuerdo al tipo de juego a jugar, ya sea Avanzar a la Meta, Bingo o Sube y Baja (**Ver Anexo #1**), sin la necesidad de tener profundos conocimientos sobre cómo funciona. Esta herramienta además de diseñar nuevos tableros, permite abrir tableros diseñados con anterioridad, a los cuales se les puede modificar el diseño inicial en cuanto a la apariencia, cambiar el color o la forma de las casillas, insertar una imagen nueva, un texto, cambiar el tipo de ficha a utilizar e insertar un dado.



2.2.1 Modelo de Dominio

Debido a que en la metodología del proceso de desarrollo del Software no existe un negocio bien definido para este proyecto, ya que pertenece a un Proyecto-Innovación-Investigación orientado al desarrollo de una herramienta Web para el diseño de tableros de juegos didácticos, se realiza un modelo de dominio:

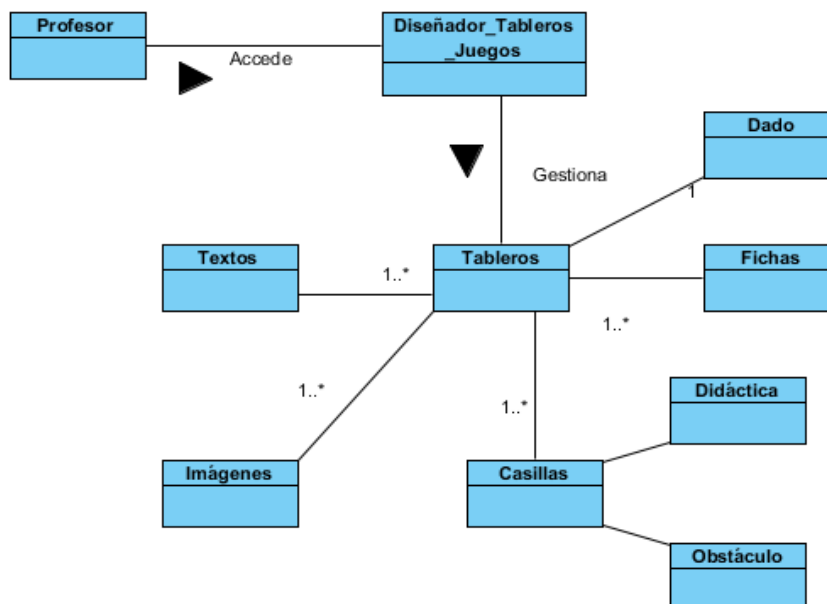


Ilustración 2: Modelo de dominio de la Herramienta de autor DEGames Board.

En la Ilustración 2 se muestra la relación existente entre las entidades o clases, es posible apreciar como el profesor es quien realiza el diseño de los tableros de juego, para esto accede a la clase “Diseñador_Tableros_Juegos”, la cual cuenta con todas las funcionalidades necesarias para diseñar y configurar los tableros de juegos didácticos mediante los distintos elementos que lo componen.

Para un mejor entendimiento del diagrama es necesario identificar todos los elementos que se utilizarán en él, mediante un **glosario de términos**:

- El **profesor** es aquella persona que posee los privilegios para acceder a esta herramienta.
- El **Diseñador_Tableros_Juegos** es la herramienta capacitada para diseñar y configurar los tableros de juegos didácticos.



- c. El **tablero** es un fichero que contiene la información necesaria para ser utilizada en el desarrollo del juego didáctico seleccionado.
- d. Las **fichas** son unos de los elementos que componen el diseño del tablero, es el elemento con el cual el jugador recorre el tablero creado.
- e. Las **imágenes** son elementos que componen el diseño del tablero para un mejor embellecimiento del entorno.
- f. El **dado** es el elemento que compone el tablero para decidir de qué manera se mueven las fichas dentro del tablero.
- g. Las **casillas** son los elementos que componen la estructura del tablero, conectadas entre sí según la manera en la que se diseña.
- h. Las **casillas didácticas** son los elementos del diseño del tablero que contendrán el componente educativo del juego.
- i. Los **obstáculos** son los elementos que componen el diseño del tablero para darle la complejidad al juego.

2.3 Levantamiento de requisitos

En la primera fase del Proceso Unificado de Desarrollo de Software (RUP), se encuentra el flujo de la Captura de Requisitos, la cual es imprescindible para la realización certera de todo proyecto de software ya que los mismos son la base para otras fases de desarrollo. En el levantamiento de requisitos se define qué es lo que el sistema debe realizar, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen. Estos requisitos pueden ser de dos tipos: funcionales y no funcionales.

2.3.1 Requerimientos funcionales

Un requisito funcional no es más que la capacidad o condición que el sistema debe cumplir. Definen qué debe hacer el software. Se mantienen invariables sin importar con qué propiedades o cualidades se relacionen.



Herrera en 2003, deja plasmado que para completar el proceso que se lleva a cabo dentro del flujo de Captura de Requisitos existe una actividad básica conocida como extracción, nombre común dado al descubrimiento de los requerimientos del sistema de cualquier fuente de información disponible. Debido a la complejidad que este proceso implica, se han trazado técnicas que permiten realizarlo de una forma más eficiente y precisa. Dentro del grupo de técnicas que de forma clásica se han utilizado en el proceso de desarrollo de todo tipo de software, para el levantamiento y validación de los requisitos funcionales de la herramienta de autor DEGames Board se utilizaron como técnicas:

♣ **Entrevista:** se realizó una entrevista con el cliente, permitiendo tomar conocimiento del problema y comprender los objetivos de la solución buscada.

♣ **Sistemas existentes:** a través de esta técnica se realizó un análisis del sistema ya desarrollado, lo que permitió observar el tipo de información que maneja y de qué manera lo hace, las distintas salidas que produce (archivos, etc.) y las interfaces de usuario correspondiente, generando una variedad de nuevas ideas sobre la base de esta.

♣ **Lluvia de ideas:** se realizó a través de reuniones con un personal con experiencia en el tema, con el objetivo de generar la mayor cantidad de ideas acertadas para el desarrollo de la herramienta propuesta.

A continuación se muestra un listado de los requisitos funcionales que identifican a la herramienta a partir del manejo de las técnicas mencionadas:

Tabla 1: Listado de requerimientos funcionales de la herramienta de autor DEGames Board.

RF1 Crear tablero	RF8 Mostrar casilla	RF15 Eliminar texto del tablero
RF2 Modificar tablero	RF9 Crear ficha	RF16 Mostrar texto en el tablero
RF3 Eliminar tablero	RF10 Modificar ficha	RF17 Insertar imagen en el tablero
RF4 Cargar tablero	RF11 Eliminar ficha	RF18 Eliminar imagen del tablero
RF5 Crear casilla	RF12 Mostrar ficha	RF19 Insertar dado
RF6 Modificar casilla	RF13 Insertar texto en el tablero	
RF7 Eliminar casilla	RF14 Modificar texto seleccionado	

A partir del levantamiento de los requisitos se hace necesario aplicar un patrón de caso de uso para definir un modelo mantenido, reusable y entendible. El patrón que se utilizó para la correcta estructuración del



comportamiento de la herramienta educativa DEGames Board fue el patrón CRUD Completo, ya que permite administrar la información del mismo y modelar las diferentes operaciones que administran una entidad de información, tales como crear, leer, modificar y eliminar.

2.3.2 Requerimientos no funcionales

Son propiedades o cualidades que el producto debe tener. Una vez que se conozca qué debe hacer el sistema puede determinarse como ha de hacerlo. En muchos casos son fundamentales en el éxito del producto.

Normalmente están vinculados a requerimientos funcionales. Los Requisitos No Funcionales que identifican a esta herramienta son:

1. Apariencia o Interfaz externa:

1.1 La aplicación Web debe contar con una interfaz profesional, siguiendo una arquitectura de información, permitiendo que los usuarios finales de la misma sean capaces de interactuar con esta, aún, cuando solo posean conocimientos básicos en el manejo de las computadoras.

1.2 La herramienta estará optimizada para una resolución de 1024 x 768.

1.3 Empleo de imágenes y colores planos.

2. Software:

2.1 Máquina virtual de Java instalada.

3. Hardware:

3.1 Para el servidor:

- ♣ Tarjeta de red.
- ♣ El Servidor de Aplicaciones debe tener 1GB de RAM y 60 GB de disco duro como mínimo.

4. Restricciones en el diseño y la implementación:

Emplear las siguientes herramientas:

- ♣ Servidor de bases de datos PostgreSQL 8.4.
- ♣ Máquina virtual de Java: JDK 7.0.
- ♣ Servidor Web Apache Tomcat 7.0.22.0.



- ♣ Visual Paradigm 8.0 for UML 2.1.
- ♣ NetBeans IDE 7.1.
- ♣ Spring 3.0.6.

5. Seguridad:

5.1 Para lograr la confidencialidad, integridad y disponibilidad de la aplicación se necesita que:

- ♣ El servidor donde esté instalada la herramienta debe estar situado en un local protegido, donde no esté expuesto a desastres naturales o robo.

6. Portabilidad:

6.1 La aplicación debe funcionar en varias plataformas, siendo posible su acceso a través de cualquier navegador Web.

2.4 Modelado del sistema

2.4.1 Actores del sistema

Los actores del sistema son entidades que guardan relación con este, pueden ser personas, otros sistemas que interactúen con el mismo o entidades abstractas como puede ser el tiempo.

A continuación se realiza una breve descripción de los actores que intervienen en el sistema DEGames Board que permite la creación de Juegos Didácticos estilo tableros para la ejercitación y evaluación de contenidos.

Tabla 2: Usuarios del sistema y su descripción.

Actor	Descripción
Profesor	Usuario autenticado que puede diseñar tableros de Juegos Didácticos.

2.4.2 Diagrama de Casos de Uso del sistema

Un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción



con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que representa gráficamente a los procesos y su interacción con los actores.

A continuación se muestran los casos de uso del sistema identificados:

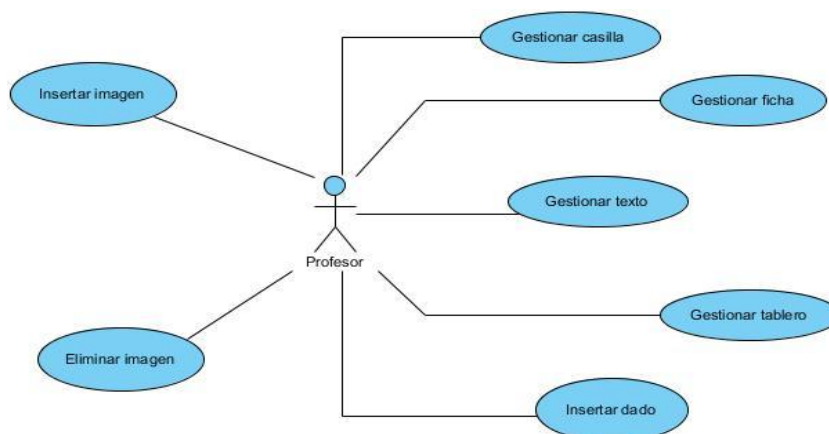


Ilustración 3: Diagrama CUs diseño de tableros de Juegos.

2.4.3 Descripción de los Casos de Uso del sistema

A continuación se muestran las descripciones abreviadas de los Casos de Uso del Sistema, las descripciones detalladas se encuentran en el **Anexo #2**.

Tabla 3: CUs Gestionar Tablero.

Caso de Uso 1	Gestionar Tablero
Actores	Profesor
Resumen	El usuario accede a las opciones que le permiten crear un tipo de tablero de juego didáctico, modificar el tablero, eliminarlo o cargar uno que ya este previamente guardado. El tablero puede ser salvado o actualizado una vez creado o abierto previamente.
Precondicione	El usuario debe estar autenticado en el Sistema como Profesor.
Referencias	RF1, RF2, RF3, RF4
Prioridad	Crítico

Tabla 4: CUs Gestionar casilla.

Caso de Uso 2	Gestionar casilla
Actores	Profesor



Resumen	El usuario puede: seleccionar el tipo de casilla que desea insertar, ya sea de avanzar, de obstáculo o didáctica y seleccionar el tipo de casilla que desea modificar, estas pueden ser transformadas, eliminadas o duplicadas.
Precondiciones	El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe haber creado o cargado un tablero antes de comenzar el caso de uso.
Referencias	RF5, RF6, RF7, RF8
Prioridad	Crítico

Tabla 5: CUs Gestionar ficha.

Caso de Uso 3	Gestionar ficha
Actores	Profesor
Resumen	El usuario 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.
Precondiciones	El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe haber creado o cargado un tablero antes de comenzar el caso de uso.
Referencias	RF9, RF10, RF11, RF12
Prioridad	Secundario

Tabla 6: CUs Gestionar texto.

Caso de Uso 4	Gestionar texto
Actores	Profesor
Resumen	El usuario introduce un texto en el tablero. Los textos pueden ser modificados o eliminados una vez insertados.
Precondiciones	El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe haber creado o cargado un tablero antes de comenzar el caso de uso.
Referencias	RF13, RF14, RF15, RF16
Prioridad	Secundario

Tabla 7: CUs Insertar imagen.

Caso de Uso 5	Insertar imagen
Actores	Profesor
Resumen	El usuario puede introducir cualquier imagen dentro del tablero.



Precondiciones	El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe haber creado o cargado un tablero antes de comenzar el caso de uso.
Referencias	RF17
Prioridad	Auxiliar

Tabla 8: CUs Eliminar imagen.

Caso de Uso 6	Eliminar imagen
Actores	Profesor
Resumen	El usuario puede eliminar cualquier imagen que se encuentre dentro del tablero en caso de no ser requerida en el diseño de tableros de juegos didácticos.
Precondiciones	El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe haber creado o cargado un tablero antes de comenzar el caso de uso. El usuario tiene que haber insertado una imagen previamente en el tablero.
Referencias	RF18
Prioridad	Auxiliar

Tabla 9: CUs Insertar dado.

Caso de Uso 7	Insertar dado
Actores	Profesor
Resumen	El usuario puede introducir un dado en el caso de que el tipo de tablero creado sea Sube y Baja o Avanzar hacia la meta.
Precondiciones	El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe crear un tablero antes de comenzar el caso de uso.
Referencias	RF19
Prioridad	Auxiliar

2.5 Arquitectura de software

La arquitectura de software de un programa o sistema de computación es la (s) estructura (s) del sistema que comprende los componentes del software, las propiedades visibles de esos componentes y las relaciones entre ellos. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer los requerimientos funcionales de un sistema, así como los requerimientos no funcionales tales como la confiabilidad, escalabilidad, portabilidad y disponibilidad.



Cuando se diseña la arquitectura de software de un sistema se parte de los requerimientos funcionales, no funcionales y los casos de usos del sistema, especificando las diferentes vistas de la arquitectura (Casos de Uso, Lógica, Procesos, Despliegue, Implementación, Datos), así como el patrón arquitectónico a seguir.

La herramienta de autor DEGames Board se diseña sobre una arquitectura por capas, específicamente una de tres capas, a partir de la utilización de un patrón arquitectónico.

2.5.1 Patrón de arquitectura Modelo Vista Controlador (MVC)

El patrón Modelo Vista Controlador (por sus siglas en inglés: *Model View Controller*) es un modelo tres capas que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos:

- ♣ El modelo: representa la información con que trabaja la aplicación.
- ♣ La vista: transforma el modelo en una interfaz a través de la cual el usuario puede interactuar con los datos.
- ♣ El controlador: se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o la vista.

Fue diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados que se nutren de los mismos datos. Su característica principal consiste en que el Modelo, las Vistas y los Controladores se tratan como entidades separadas, lo que posibilita que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas, permitiendo además la implementación de módulos por separado.

2.5.2 Fundamentación de la arquitectura a utilizar

La herramienta de autor para el diseño de tableros de juegos didácticos utiliza el patrón MVC con el objetivo de mantener la independencia de los datos y permitir la inclusión de nuevos Juegos Didácticos, así como la sencilla modificación de funcionalidades o interfaces visuales. Teniendo en cuenta además el empleo del Framework Spring que trabaja sobre este patrón arquitectónico MVC.

A continuación se muestran los artefactos del diseño correspondientes a los casos de uso más representativos dentro del sistema, lo cual constituye la finalidad de la herramienta.



2.6 Patrones de diseño

Con el objetivo de lograr un sistema robusto y reutilizable el diseño se realizó utilizando patrones de diseño, ya que estos son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software.

Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe cumplir con ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas, debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Los patrones de diseño utilizados son los de asignación de responsabilidades (GRASP).

Controlador: este patrón tiene en cuenta para realizar las asignaciones en cuanto al manejo de los eventos del sistema y definir sus operaciones. Spring Framework contribuye a la utilización de este patrón ya que facilita una serie de clases controladoras, que manejan la lógica de la navegación e interactúan con la capa de Negocio de la aplicación Web.

Experto: este patrón propone como solución asignar la responsabilidad a la clase que cuenta con la información necesaria para cumplir la responsabilidad.

Creador: este patrón se tiene en cuenta para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto solo pueda ser creada por el objeto que contiene la información necesaria para ello. El uso de este patrón permite crear las dependencias mínimas necesarias entre las clases, favoreciendo al mantenimiento del sistema.

Bajo acoplamiento: este patrón brinda como solución asignar responsabilidades de manera que las clases no dependan fuertemente de otras. Ofreciendo como beneficio que son fáciles de entender por separadas, fáciles de reutilizar y no se afectan por cambios de otros componentes. Dicho patrón se tiene en cuenta debido a la importancia de realizar un diseño de clases utilizando el patrón arquitectónico MVC que soporta los cambios.

Alta cohesión: este patrón propone asignar la responsabilidad de manera que la complejidad se mantenga dentro de límites manejables asumiendo solamente las responsabilidades que deben manejar, evadiendo



un trabajo excesivo. Su utilización mejora la claridad y facilidad con que se entiende el diseño, simplifica el mantenimiento y las mejoras de funcionalidad, generan un bajo acoplamiento, soporta mayor capacidad de reutilización.

2.7 Diagrama de Clases del Diseño

Un diagrama de clases del diseño es un diagrama que describe gráficamente las especificaciones de las clases existentes en un producto de software así como de las interfaces involucradas. Estos diagramas contienen información útil para el usuario como por ejemplo clases, asociaciones y atributos, interfaces con sus operaciones y constantes, métodos, información sobre los tipos de atributos, navegabilidad y dependencia existente. A diferencia de un modelo conceptual, un diagrama de clases del diseño contiene las definiciones de las entidades del software en vez de conceptos del mundo real **(15)**.

El diagrama con las clases que conforman el sistema, especificando los atributos y las operaciones que las mismas realizan se muestra en los anexos (**Ver Anexo #3**).

2.8 Conclusiones parciales

Con la definición de la propuesta de solución, el análisis de los requerimientos funcionales y la identificación de los casos de usos necesarios con que debe contar la herramienta, se pudo concluir que es posible obtener el diseño de la herramienta de autor para la creación de tableros de juegos didáctico DEGames Board en la Web. Además con la descripción de la arquitectura se logró un mejor entendimiento del código de la herramienta y quedó clara la posibilidad de modificar funcionalidades o interfaces visuales de forma sencilla, manteniendo la independencia entre los datos



Capítulo III: Implementación y pruebas.

3.1 Introducción

Según el Proceso Unificado de Desarrollo de Software (RUP), la Implementación es el centro de la Fase de Construcción, aunque también se lleva a cabo durante la Fase de Elaboración para crear la línea base ejecutable de la arquitectura y durante la de Transición para tratar defectos que se hayan encontrado en ese momento, en caso de que existan.

El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes, es decir, ficheros de código fuente o binario, scripts, ejecutables o similares. Aquí se implementan las clases elaboradas durante el Diseño y se muestra su organización.

3.2 Modelo de Despliegue

El modelo de despliegue como propósito principal captura la configuración de los elementos de procesamiento, así como las conexiones entre estos elementos en el sistema. El modelo consiste en uno o más nodos, dispositivos y conectores, entre otros. Este modelo también permite mapear procesos dentro de estos elementos de procesamiento, permitiendo la distribución del comportamiento a través de los nodos que son representados.

A continuación se muestra el diagrama de despliegue modelado para la herramienta a desarrollar:

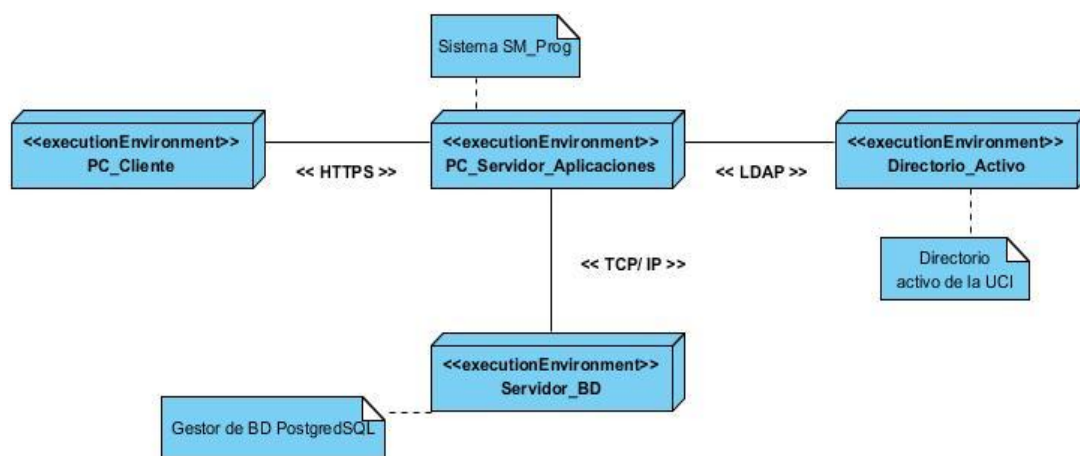


Ilustración 4: Diagrama de despliegue de la herramienta DEGames Board.



En la ilustración 4 se representa el diagrama de despliegue de la herramienta DEGames Board en el que se ilustran las relaciones software-hardware necesarias para el correcto funcionamiento de la aplicación en un ambiente de trabajo real y óptimo, para lo cual es necesario mantener el servidor de Base de Datos separados de la PC Cliente o máquina donde se ejecute dicha aplicación, con el objetivo de evitar sobrecarga en los servidores, ya que una vez que cierta cantidad de usuarios interactúen con el sistema es necesario evitar demoras en las transacciones que se realicen.

El usuario (desde la PC Cliente) tendrá acceso a la herramienta DEGames Board, a partir del ejecutable de la misma, la cual posibilita diseñar un tablero según el rol del usuario autenticado contra el directorio activo (LDAP) de la UCI. Existe además una conexión mediante el protocolo TCP/IP con la base de datos donde se almacenan los ficheros XML que contienen toda la información relacionada con el tablero diseñado.

3.3 Modelo de Implementación

El diagrama de componentes siguiendo la arquitectura definida de acuerdo con lo que RUP plantea, es el encargado de mostrar la estructura de alto nivel del Modelo de Implementación, específicamente los Subsistemas de Implementación y sus dependencias de importación. También se usan para mostrar ficheros de código fuente y sus dependencias de compilación; ficheros de aplicación y sus dependencias en tiempo de ejecución; relaciones derivadas entre ficheros de código fuente y ficheros resultantes de la compilación.

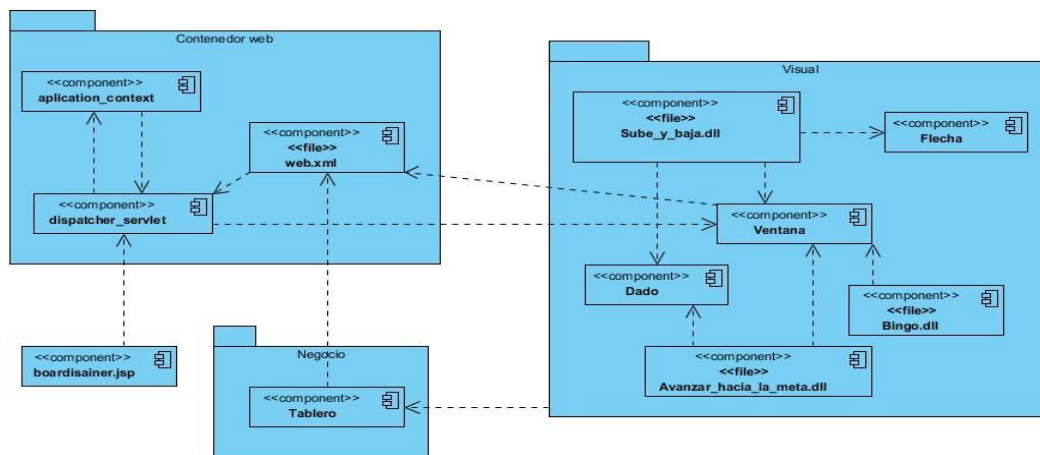


Ilustración 5: Diagrama de componente.



En el diagrama presentado aparecen los componentes y paquetes imprescindibles que intervienen en todo el proceso, donde el componente boardisainer.jsp es quien se relaciona con el cliente directamente y el contenedor Web es quien domina y enlaza el resto de los paquetes y clases.

Los paquetes de Negocio y Visual son los contenedores de las clases que intervienen en los tipos de tableros. En el primer paquete se percibe el componente Tablero que contiene todas las clases relacionadas con él y en el segundo paquete se ven como los 3 tipos de tableros utilizan el componente ventana, también se evidencia la presencia del dado que es utilizado por los tipos de tableros Sube y Baja y Avanza hasta la Meta, así como la flecha es solamente la usa el tipo de tablero Suba y Baja

3.4 Almacenamiento de datos

3.4.1 Fichero XML para el tablero de juego

Aún, cuando el sistema central utilice una base de datos para almacenar los datos relacionados con el sistema de preguntas, la herramienta de autor DEGames Board almacenará la información referente a los tableros de juegos didácticos en un fichero XML. Esto permite cargar y guardar la información que se almacena de manera eficiente y ocupando poco espacio en disco.

El fichero contará con un formato propio donde se encuentren las características de cada tipo de casillas, de los textos, de las imágenes, así como la ubicación y relación de cada uno de estos elementos dentro del tablero. La estructura del documento XML puede ser consultada en los anexos (**Ver Anexo #4**).

3.5 Estándares de codificación

Los Estándares de Codificación son reglas específicas a un lenguaje que reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores. Durante el desarrollo de un software estos estándares ayudan a los ingenieros a producir un código de alta calidad y a entender y a utilizar el código de sus colegas. Pero también realzan considerablemente la capacidad de mantenimiento y reusabilidad a largo plazo del producto final **(30)**. Con esto se reduce la posibilidad de introducir errores en el código haciéndolo más legible y fácil de leer ayudando así al posterior mantenimiento del software.



CamelCase

Es un estilo de escritura que se aplica a frases o palabras compuestas. Se caracteriza porque las palabras van unidas entre sí sin espacios; con la peculiaridad de que la primera letra de cada término se encuentra en mayúscula para hacer más legible el conjunto. Tradicionalmente, se había utilizado para la formulación química; pasando a ser empleado, en la actualidad, como un lenguaje de catalogación o clasificación, utilizado exclusivamente en la Web como lenguaje de programación.

El CamelCase admite dos posibles combinaciones entre mayúsculas y minúsculas: la primera palabra en mayúscula y el resto en minúscula (UpperCamelCase) o, por el contrario, cuando la primera está en minúscula y las demás están en mayúscula (LowerCamelCase).

3.6 Pruebas

Las pruebas son actividades a través de las cuales un sistema o componente es ejecutado bajo ciertas condiciones o requisitos que permiten determinar de forma parcial o completa si el funcionamiento es el esperado o si existen errores que atenten contra el buen desempeño de la aplicación.

3.6.1 Pruebas funcionales

Las pruebas funcionales son pruebas de software que tienen por objetivo probar que los sistemas desarrollados cumplan con los requisitos funcionales. A este tipo de pruebas se les denomina también pruebas de comportamiento y para realizarlas se emplea el método de caja negra, donde los probadores o analistas de pruebas no enfocan su atención en cómo se generan las respuestas del sistema, sino en el funcionamiento de la interfaz del sistema.

El principal objetivo de este tipo de pruebas es demostrar que las funciones del software son operativas, que la entrada y la salida se producen de forma correcta y que además se mantiene la integridad de la información.

Método de caja negra

Las pruebas de caja negra o funcionales son las que se realizan sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar. También conocidas como Pruebas de



Comportamiento, estas pruebas se basan en la especificación del programa o componente a ser probado para elaborar los casos de prueba. Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas: basado en grafos, partición equivalente, análisis de valores límites y prueba de talla ortogonal.

A la herramienta educativa DEGames Board se le realizaron pruebas basadas en la técnica de partición de equivalencia. Este es un método que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba, dirigiendo su trabajo a descubrir clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. El diseño de los casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una condición de entrada (31).

Casos de Prueba

A partir de la técnica seleccionada la prueba se realizó basada en los casos de uso, diseñando un caso de prueba por cada uno de estos.

♣ Caso de Prueba: Gestionar Tablero.

Descripción general: El usuario accede a las opciones que le permiten crear un tipo de tablero de juego didáctico, modificar el tablero, eliminarlo o cargar uno que ya este previamente guardado. El tablero puede ser salvado o actualizado una vez creado o abierto previamente.

Condiciones de ejecución: El usuario debe estar registrado como profesor y tener los permisos necesarios para realizar las acciones. No debe existir ningún tablero abierto en la aplicación.

Secciones

Tabla 10: Caso de Prueba Gestionar tablero.

Nombre de la Sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1 Crear tablero.	EC 1.1 Crear nuevo tablero.	Se selecciona la opción Nuevo Tablero y se muestra el formulario correspondiente con los datos que son necesarios.
	EC 1.2 Ingresar datos de un tablero.	Se ingresan los datos necesarios y se presiona el botón "Aceptar". Se almacenan los datos y se crea el tablero.



	EC 1.3 Cerrar ventana "Tablero".	Se presiona el botón "Cerrar", no se guardan los cambios y se regresa al EC1.1.
SC 2 Cargar tablero	EC 2.1 Abrir tablero	Se selecciona la opción "Abrir tablero" y se muestra un formulario para escoger el tablero que se desea abrir.
	EC 2.2 Seleccionar	Se selecciona el tablero a cargar.
	EC 2.3 Cerrar ventana "Cargar"	Se presiona el botón "Cerrar", no se selecciona nada y se regresa al EC 2.1.
SC 3 Guardar	EC 3.1 Guardar tablero	Se selecciona la opción "Guardar" y se muestra el formulario correspondiente.
	EC 3.2 Insertar datos para guardar un tablero	Se ingresan los datos necesarios, se presiona el botón "Aceptar" y se guardan los datos.
SC 4 Modificar tablero	EC 4.1 Modificar alto del tablero.	Se activa la paleta de propiedades, se selecciona en la opción "Alto" y se modifica el tamaño del tablero.
	EC 4.2 Modificar ancho del tablero.	Se activa la paleta de propiedades, se selecciona en la opción "Ancho" y se modifica el tamaño del tablero.
	EC 4.3 Modificar fondo del tablero.	Se activa la paleta de propiedades. Se presiona la opción "Fondo", se muestra el formulario correspondiente.
	EC 4.4 Insertar datos necesarios.	Se introducen los datos necesarios, se presiona el botón "Aceptar" y se realizan las modificaciones.

♣ **Caso de Prueba: Gestionar casilla.**

Descripción general: El usuario puede: seleccionar el tipo de casilla que desea insertar, ya sea de avanzar, de obstáculo o didáctica y seleccionar el tipo de casilla que desea modificar, estas pueden ser transformadas, eliminadas o duplicadas.

Condiciones de ejecución: El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe haber creado o cargado un tablero en la aplicación. El tablero debe estar creado y debe existir al menos una casilla insertada.



Secciones

Tabla 11: Caso de Prueba Gestionar casilla.

Nombre de la Sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1 Avanzar	EC 1.1 Insertar casilla avanzar.	Se selecciona el componente “casilla avanzar” y se arrastra hacia el tablero.
	EC1.2 Mostrar casilla avanzar.	Se muestra la casilla con un pie en el tablero.
SC 2 Obstáculo	EC 2.1 Insertar casilla obstáculo.	Se selecciona el componente “casilla obstáculo” y se arrastra hacia el tablero.
	EC 2.2 Mostrar casilla obstáculo.	Se muestra la casilla con un cono en el tablero.
SC 3 Didáctica	EC 3.1 Insertar casilla didáctica.	Se selecciona el componente “casilla didáctica” y se arrastra hacia el tablero.
	EC 3.2 Mostrar casilla didáctica.	Se muestra la casilla con un “?” en el tablero.
	EC 3.3 Insertar casilla didáctica (exclamación)	Se selecciona el componente “casilla didáctica (exclamación)” y se arrastra hacia el tablero.
	EC 3.4 Mostrar casilla didáctica (exclamación).	Se muestra la casilla con un “!” en el tablero.
	EC 3.5 Insertar casilla didáctica (asterisco)	Se selecciona el componente “casilla didáctica (asterisco)” y se arrastra hacia el tablero.
	EC 3.6 Mostrar casilla didáctica (asterisco).	Se muestra la casilla con un “*” en el tablero.
SC 4 Modificar casilla.	EC 4.1 Modificar fondo de las casillas.	Se activa la paleta de propiedades, se selecciona en la opción “Imagen” y se muestra el formulario correspondiente.
	EC 4.2 Insertar datos necesarios.	Se introducen los datos necesarios, se presiona el botón “Aceptar” y se realizan las modificaciones.
	EC 4.3 Modificar ángulo de las casillas.	Se activa la paleta de propiedades, se selecciona en la opción “Rotación”, se cambia el ángulo correspondiente y



		se realiza el cambio.
	EC 4.4 Modificar rondas para la casilla obstáculo.	Se activa la paleta de propiedades, se selecciona en la opción "Ronda" y se cambia el número de la ronda.
	EC 4.5 Modificar castigo para la casilla obstáculo.	Se activa la paleta de propiedades, se selecciona en la opción "Castigo" y se muestra el formulario correspondiente.
	EC 4.6 Insertar datos necesarios.	Se introducen los datos necesarios, se presiona el botón "Aceptar" y se realizan las modificaciones.
SC 5 Eliminar casillas.	EC 5.1 Eliminar casilla.	Se selecciona la casilla, seguidamente se presiona la opción "Eliminar" y se elimina la casilla seleccionada.

♣ **Caso de Prueba: Gestionar ficha.**

Descripción general: El usuario 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.

Condiciones de ejecución: El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe haber creado o cargado un tablero antes de comenzar el caso de uso.

Secciones

Tabla 12: Caso de Prueba Gestionar ficha.

Nombre de la Sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1 Ficha	EC 1.1 Insertar ficha.	Se selecciona el componente "ficha" y se arrastra hacia el tablero.
	EC 1.2 Mostrar ficha	Se muestra la ficha en el tablero.
SC 2 Modificar ficha.	EC 2.1 Modificar ficha.	Se activa la paleta de propiedades, se selecciona en la opción "Imagen" y se muestra el formulario correspondiente.
	EC 2.2 Insertar datos necesarios.	Se introducen los datos necesarios, se presiona el botón "Aceptar" y se realizan las modificaciones.



SC 3 Eliminar fichas.	EC 3.1 Eliminar ficha.	Se selecciona la ficha, seguidamente se presiona la opción "Eliminar" y se elimina la ficha seleccionada.
-----------------------	------------------------	---

♣ **Caso de Prueba: Gestionar texto.**

Descripción general: El usuario introduce un texto en el tablero. Los textos pueden ser modificados o eliminados una vez insertados.

Condiciones de ejecución: El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe haber creado o abierto un tablero (fichero) en la aplicación.

Secciones

Tabla 13: Caso de Prueba Gestionar texto.

Nombre de la Sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1 Texto	EC 1.1 Insertar texto.	Se selecciona el componente "texto" y se arrastra hacia el tablero.
	EC 1.2 Mostrar texto	Se muestra un cuadro de texto en el tablero.
SC 2 Modificar texto.	EC 2.1 Modificar texto.	Se activa la paleta de propiedades, se selecciona en la opción "Propiedades" y se muestra el formulario correspondiente.
	EC 2.2 Insertar datos necesarios.	Se introducen los datos necesarios, se presiona el botón "Aceptar" y se realizan las modificaciones.
SC 3 Eliminar texto.	EC 3.1 Eliminar texto.	Se selecciona el texto, seguidamente se presiona la opción "Eliminar" y se elimina el cuadro de texto seleccionado.

♣ **Caso de Prueba: Insertar imagen.**

Descripción general: El usuario puede introducir cualquier imagen dentro del tablero.

Condiciones de ejecución: El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe haber creado o cargado un tablero antes de comenzar el caso de uso.



Secciones

Tabla 14: Caso de Prueba Insertar imagen.

Nombre de la Sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1 Insertar imagen.	EC 1.1 Insertar imagen.	Se selecciona el componente "imagen", se arrastra hacia el tablero y se muestra el formulario correspondiente.
	EC 1.2 Ingresar datos de una imagen.	Se ingresan los datos necesarios, se presiona el botón "Aceptar" y se almacenan los datos.
	EC 1.3 Mostrar imagen	Se muestra la imagen en el tablero.

♣ Caso de Prueba: Eliminar imagen.

Descripción general: El usuario puede eliminar cualquier imagen que se encuentre dentro del tablero en caso de no ser requerida en el diseño de tableros de juegos didácticos.

Condiciones de ejecución: El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe haber creado o cargado un tablero antes de comenzar el caso de uso. El usuario tiene que haber insertado una imagen previamente en el tablero.

Secciones

Tabla 15: Caso de Prueba Eliminar imagen.

Nombre de la Sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1 Eliminar imagen.	EC 1.1 Eliminar imagen.	Se selecciona la imagen, seguidamente se presiona la opción "Eliminar" y se elimina la imagen seleccionada.

♣ Caso de Prueba: Insertar dado.

Descripción general: El usuario puede introducir un dado dentro del tablero, siempre y cuando este sea de tipo Avanzar hacia la meta o Sube y Baja.

Condiciones de ejecución: El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe crear un tablero antes de comenzar el caso de uso.



Secciones

Tabla 16: Caso de Prueba Insertar dado.

Nombre de la Sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1 Insertar imagen.	EC 1.1 Insertar imagen.	Se selecciona el componente “dado” y se arrastra hacia el tablero.
	EC 1.2 Mostrar dado	Se muestra el dado en el tablero.

Tabla 17: Descripción de variables.

No.	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Nombre	Campo de texto	si	Textos, números y símbolos.
2	Tablero	Lista desplegable	no	Texto
3	Alto	Campo de texto	si	Números
4	Ancho	Campo de texto	si	Números
5	Fondo	Lista desplegable	no	Texto
6	Ronda	Campo de texto	si	Números
7	Castigo	Lista desplegable	no	Texto
8	Tipo	Lista desplegable	no	Texto
9	Ángulo	Campo de texto	si	Números

Ejecución de los casos de prueba

La ejecución de las pruebas se realizó en tres iteraciones por cada escenario según el comportamiento de la herramienta a partir de la entrada de los datos recogidos en la tabla de variables (**ver Anexo #5**).

3.6.2 Resultados obtenidos

Al aplicar las pruebas funcionales se alcanzaron los siguientes resultados en cada una de las iteraciones aplicadas a la herramienta:

- ♣ En la primera iteración de 45 eventos probados, 17 de ellos arrojaron resultados satisfactorios y 28 no satisfactorios, estos últimos están dados por la detección de errores tales como: funcionalidades ausentes, errores de interfaz y funcionalidades incorrectas.



♣ Al efectuarse la segunda iteración, se mostró que se habían corregido los errores anteriores, detectándose 29 eventos con resultados satisfactorios y 16 no satisfactorios. Manteniendo errores como: funcionalidades ausentes y errores de interfaz que fueron rectificadas.

♣ En la tercera iteración no se encontraron eventos no satisfactorios, demostrando que la herramienta cumple con los requisitos funcionales expuestos anteriormente.

La siguiente ilustración muestra gráficamente el resultado obtenido en las pruebas, por cada una de las iteraciones mencionadas.

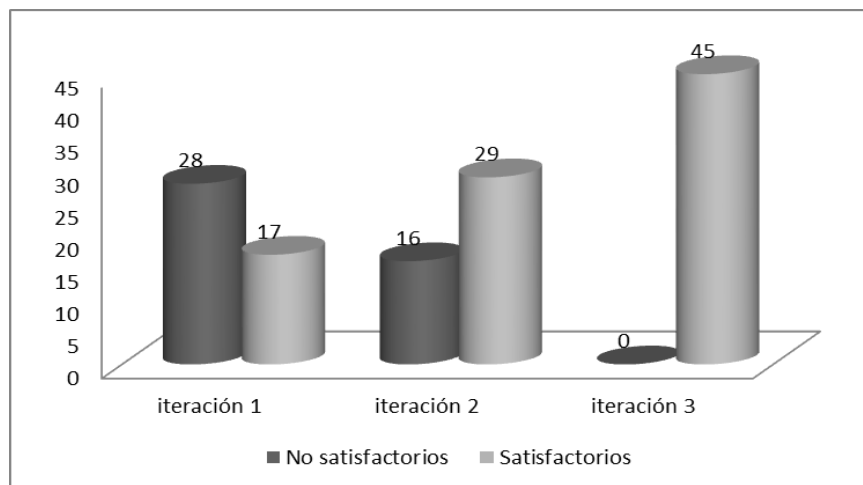


Ilustración 6: Comportamiento por iteraciones de las pruebas funcionales.

3.7 Conclusiones parciales

La aplicación de las pruebas funcionales a través del método de caja negra, permitieron detectar errores y problemas a los que era necesario darle solución, tales como: funcionalidades ausentes, funcionalidades incorrectas y errores de interfaz. Al finalizar la ejecución de los casos de prueba, se obtuvieron resultados satisfactorios, por lo que se puede concluir que la herramienta de autor se encuentra correctamente diseñada y se presupone será efectiva al aplicarla para la ejercitación de contenidos, dando cumplimiento a los objetivos trazados durante la investigación.



Conclusiones

Con el desarrollo de esta investigación se puede afirmar que los objetivos trazados fueron cumplidos satisfactoriamente, alcanzando los siguientes resultados:

1. Las herramientas de autor analizadas, a pesar de las amplias posibilidades que brindan en cuanto al desarrollo de recursos educativos, no permiten la creación de tableros de juegos didácticos en la Web para la ejercitación de contenidos en disciplinas de Programación.
2. La herramienta educativa DEGames Board tiene un diseño adecuado para su propósito, con todos sus componentes visuales y todas las funcionalidades en correspondencia con las características diseñadas y los requisitos trazados, de acuerdo a la revisión de la arquitectura y el funcionamiento de la herramienta.
3. La aplicación de pruebas a la herramienta de autor DEGames Board, posibilitó solucionar los errores encontrados durante la ejecución de los casos de pruebas, logrando así, obtener un producto funcional y con calidad.
4. La herramienta de autor DEGames Board contribuye con el PEA de la disciplina Técnicas de Programación de Computadoras, al facilitar el diseño de tableros de juegos didácticos para la ejercitación de contenidos.



Recomendaciones y trabajo futuro

A partir del estudio realizado en la presente investigación y teniendo en cuenta las ideas que surgieron durante el desarrollo del diseño de la herramienta de autor DEGames Board, se realizan las siguientes recomendaciones:

- ✍ Validar la herramienta de autor DEGames Board en el colectivo de profesores de Programación de la Facultad 1 y del Departamento Central de Programación, utilizando el método definido en los anexos. **(Ver Anexo #6)**
- ✍ Integrar la herramienta de autor DEGames Board al sistema SMProg (Software-Motivación-Programación) luego de ser validada por los expertos.



Referencias Bibliográficas

1. PERS, H. Guía de Metodologías Participativas y Juego Educativo. En *Programa de Formación Técnica Laboral para Jóvenes Bachilleres. Bolivia. 2009.* p. 70.
2. DE ZAYAS, C. M. A. *La escuela en la vida.* Editorial Félix Varela, 1992. 186 p. Colección Educación y desarrollo. ISBN 9789590700156.
3. *Herramientas de autor - EcuRed.* Última actualización: 2013/01/18/20:35:53. Disponible en: http://www.ecured.cu/index.php/Herramientas_de_autor/files/9/Herramientas_de_autor.html.
4. GUERRA, Y. A. y GUERRA, Y. A. Conceptualización de una red social educativa que integre de forma colaborativa las aplicaciones e-learning de la Universidad de las Ciencias Informáticas. 2011 2011, vol. 5, nº 3, [Consultado el: 2013/01/28/21:50:52]. Disponible en: <http://rcci.uci.cu/index.php/rcci/article/view/144/files/53/144.html>.
5. BARRIOS, L. C. *Módulo para la construcción de tableros de juegos didácticos de la herramienta educativa SoftMotProg para el Entorno Virtual de Aprendizaje Moodle.* Tutor: Ferreiro, L. D. S. y Hernández, I. Y. V. Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas, FACULTAD 10. Universidad de las Ciencias Informáticas, 2008.
6. *zonaClic - Jclic.* Última actualización: 2013/05/03/12:51:25. Disponible en: <http://clic.xtec.cat/es/jclic/files/105/jclic.html>.
7. BOUZÁN MATANZA, J. M. .: *Ardora :: Creación de actividades escolares* [Web de ayuda de Ardora-Creación de contenidos para la web]. Última actualización: 2013/02/11/08:56:24. (webArdora.net). Disponible en: http://webardora.net/index_cas.html/files/155/index_cas.html.
8. HECHAVARRÍA, R. A. y RIO, D. D. D. *Análisis, diseño e implementación de la versión 2.0 de los juegos del primer ciclo de la colección multisaber.* Tutor: Ing. Ismael Armando Nodarse Mora, I. a. M. R. R. Trabajo de Diploma, Facultad 3. Universidad de las Ciencias Informáticas, 2010.
9. EQUIPO DE DESARROLLO DE, H. *HAEduc* Última actualización: 2013/02/08/22:20:27. (HAEduc). Disponible en: <http://www.haeduc.rimed.cu/files/151/www.haeduc.rimed.cu.html>.
10. FALCÓN GARCÍA, M. M. *Implementación del nivel A de IMS-LD para la herramienta de autor web CRODA.* Facultad 4. Universidad de las Ciencias Informáticas, 2011.



11. LEÓN, C. M. R. *Las esferas socioculturales del Software Libre* Última actualización: 2004 2013/05/07/15:40:48. (Derecho de Internet). Disponible en: <http://www.derecho-internet.org/node/293files/242/293.html>.
12. *Sobre PostgreSQL*. [portal web]. Última actualización: 2010/10/02/de de 2013/05/07/15:02:16. Disponible en: http://www.postgresql.org/es/sobre_postgresqlfiles/227/sobre_postgresql.html.
13. PARADIGM, V. *Visual Paradigm for UML 10.1 Modeler Edition* Disponible en: <http://www.visual-paradigm.com/product/vpuml/editions/modeler.jsp>.
14. JAMES, R.; GARY, B., et al. *El proceso unificado de desarrollo de software*. ADDISON WESLEY ed. Madrid: 2000. 438 p. ISBN 8478290362.
15. LARMAN, C. *UML y patrones: introducción al análisis y diseño orientado a objetos*. Félix Varela, 2004.
16. GUTIÉRREZ, J. J. ¿ Qué es un framework web? 2006 2006, nº
17. JOHNSON, R. y HOELLER, J. *Expert one-on-one J2EE development without EJB*. ilustrada ed. Wrox, 2004. 552 p. Expert One-On-One Programmer to programmer. ISBN 0764558315.
18. JAVA, E. D. *Conozca más sobre la tecnología Java* Última actualización: 2013/05/07/15:09:41. Disponible en: <http://www.java.com/es/about/files/230/about.html>.
19. *Joomla! y sus lenguajes de programación: PHP, HTML, XHTML, SQL, XML, JavaScript, AJAX, CSS - Joomla-Chile.cl*. Última actualización: 2013/05/07/15:12:00. (Desarrollo de tecnología web avanzada). Disponible en: <http://joomla-chile.cl/unidad-i/135-joomla-y-sus-lenguajes-de-programacion-php-html-xhtml-sql-xml-javascript-ajax-css.htmlfiles/232/135-joomla-y-sus-lenguajes-de-programacion-php-html-xhtml-sql-xml-javascript-ajax-css.html>.
20. MUSCIANO, C. y KEMEDY, B. *HTML. La guía completa*. Traducido por: Licenciada En Computacih Yazmh Ju5rez Parra, I. E. E. a. M. 2da. ed. Mexico, D.F: McCraw-Hill Companies, Inc, 1999. 546 p. Ayuda para autores de web. ISBN 1-56592-235.
21. PÉREZ, J. E. *Introducción a CSS*. 2007. 272 p.



-
22. JAVIER EGUÍLUZ, P. *Introducción a AJAX*. 1era ed. España: Creative Commons, 2009.
 23. *jQuery*. Última actualización: 2013/05/22/21:35:11. Disponible en: <http://jquery.com/files/251/jquery.com.html>.
 24. *Raphaël—JavaScript Library*. Última actualización: 2013/05/22/19:07:53. Disponible en: [http://raphaeljs.com/index.html](http://dmitrybaranovskiy.github.io/raphael/index.html)files/249/index.html.
 25. DEVELOPERWORKS, I. *Introducción a XML*.
 26. DENVER, P. *PostgreSQL*. [Power Point]. publicado el: 2002/11/08/de de de 2002, última actualización: 2002/11/08/de de. Disponible en: http://www.slidefinder.net/g/qu%C3%A9_postgresql/32057312.
 27. EAVES, J. y GODFREY, R. J. W. *Apache Tomcat Bible*. Wiley India Pvt. Limited, 2003. 817 p. ISBN 9788126504305.
 28. WALLS, C. y BREIDENBACH, R. *Spring in Action*. Second ed. Greenwich,: Manning Publications Co., 2008. 765 p. ISBN 1-933988-13-4.
 29. SÁNCHEZ, J. *NetBeans 3.51 guía rápida*. publicado el: 2004 de 2004, última actualización: 2004. Disponible en: <http://es.scribd.com/doc/6615294/NetBeans>.
 30. *Notación y estilo en programación*. Última actualización: 2013/05/02/21:46:46. Disponible en: <http://www.bitriding.com/articulos/notacion-estilo-programacion.html>files/202/notacion-estilo-programacion.html.
 31. PRESSMAN, R. S. A. y MURRIETA, J. E. M. *Ingeniería del software: un enfoque práctico*. 6 ed. Mcgraw Hill/Interamericana Editores, 2006. 958 p. ISBN 9789701054734.



Bibliografía Consultada

- ATutor Learning Management System: Information.* Última actualización: 2013/05/08/09:03:14. Disponible en: <http://atutor.ca/files/247/atutor.ca.html>.
- Conferencia # 1.* [Entorno virtual de aprendizaje]. Última actualización: 2011/03/04/de de. (Ingeniería de software 2). Disponible en: <http://eva.uci.cu>.
- Herramientas de autor - EcuRed.* Última actualización: 2013/01/18/20:35:53. Disponible en: http://www.ecured.cu/index.php/Herramientas_de_autor/files/9/Herramientas_de_autor.html.
- Java Help Center.* Última actualización: 2013/05/07/14:33:38. Disponible en: <http://www.java.com/en/download/help/files/216/help.html>.
- Joomla! y sus lenguajes de programación: PHP, HTML, XHTML, SQL, XML, JavaScript, AJAX, CSS - Joomla-Chile.cl.* Última actualización: 2013/05/07/15:12:00. (Desarrollo de tecnología web avanzada). Disponible en: <http://joomla-chile.cl/unidad-i/135-joomla-y-sus-lenguajes-de-programacion-php-html-xhtml-sql-xml-javascript-ajax-css.htmlfiles/232/135-joomla-y-sus-lenguajes-de-programacion-php-html-xhtml-sql-xml-javascript-ajax-css.html>.
- jQuery.* Última actualización: 2013/05/22/21:35:11. Disponible en: <http://jquery.com/files/251/jquery.com.html>.
- Notación y estilo en programación.* Última actualización: 2013/05/02/21:46:46. Disponible en: <http://www.bitriding.com/articulos/notacion-estilo-programacion.htmlfiles/202/notacion-estilo-programacion.html>.
- Portal Educativo Cubaeduca.* [portal]. Última actualización: 2013/02/08/22:30:08. (CubaEduca). Disponible en: http://www.cubaeduca.cu/index.php?option=com_content&view=article&id=3823&Itemid=153files/153/index.html.
- Raphaël—JavaScript Library.* Última actualización: 2013/05/22/19:07:53. Disponible en: <http://raphaeljs.com/index.htmlfiles/249/index.html>.
- Sobre PostgreSQL.* [portal web]. Última actualización: 2010/10/02/de de 2013/05/07/15:02:16. Disponible en: http://www.postgresql.org.es/sobre_postgresqlfiles/227/sobre_postgresql.html.
- SpringSource.org.* [Comunidad]. Última actualización: 2013/02/04/20:56:00. Disponible en: <http://www.springsource.org/?cid=7018000000wle4AAEfiles/91/www.springsource.org.html>.
- UML, BPMN and Enterprise Architecture Tool for Software Development.* Última actualización: 2013/05/07/14:58:47. Disponible en: <http://www.visual-paradigm.com/files/225/www.visual-paradigm.com.html>.



- Visual Paradigm for UML (ME) - (Paradigma Visual para UML (ME)) (Visual Paradigm for UML (ME))* por Visual Paradigm International Ltd. - reporte y descarga. [Sitio de descarga libre de software]. Última actualización: 2007/03/05/de de 2013/05/07/14:55:25. Disponible en: [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/files/223/Paradigma_Visual_para_UML_\(MÍ\)_14720_p.html](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/files/223/Paradigma_Visual_para_UML_(MÍ)_14720_p.html).
- zonaClic - ¿Qué es JClic?* Última actualización: 2013/05/03/12:51:31. Disponible en: <http://clic.xtec.cat/es/jclic/howto.htmfiles/204/howto.html>.
- ADELL, J. *Tendencias en educación en la sociedad de las tecnologías de la información*. publicado el: 1997/11//de de 1997, última actualización: 1997/11//de. [Consultado el: 2013/01/18/20:51:13]. Disponible en: <http://nti.uji.es/~jordi>GrupdeNovesTecnologiesaplicadesal'Educaciófiles/17/revelec7.html>. ISBN 1135-9250.
- AFILIATE, C. *¿Qué es SCORM?* [Desarrollo de Componentes, contenidos, mantenimiento, soporte y formación]. Última actualización: 2013/02/08/22:07:49. (SAMOO). Disponible en: <http://samoo.es/index.php/es/component/content/article/17-roknewsrotator/1-more-versatility-more-of-everythingfiles/141/1-more-versatility-more-of-everything.html>.
- AMO, F. A.; NORMAND, L. A. M., et al. *Introducción a la ingeniería del software*. Delta Publicaciones, 2005. 542 p. ISBN 9788496477001.
- ANA, A. *SCORM* Última actualización: 2013/02/08/22:07:26. (Área de Tecnologías para la Docencia). Disponible en: http://add.unizar.es/add/area/index.php?option=com_content&view=article&id=72:scorm&catid=6&Itemid=43files/139/index.html.
- ARGOTE, J. C. y LABAÑINO, H. R. *Análisis y Diseño de una herramienta de autor Web interoperable*. Tutor: Leyva, M. D. L.; Avila, M. D. T. et al. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas, Facultad 10. Universidad de las Ciencias Informáticas, 2008.
- BARRIOS, L. C. *Módulo para la construcción de tableros de juegos didácticos de la herramienta educativa SoftMotProg para el Entorno Virtual de Aprendizaje Moodle*. Tutor: Ferreiro, L. D. S. y Hernández, I. Y. V. Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas, FACULTAD 10. Universidad de las Ciencias Informáticas, 2008.
- BERZAL, F.; CORTIJO, F. J., et al. *Desarrollo Profesional de Aplicaciones Web con ASP.NET*. iKor Consulting, 179 p. ISBN 8460942457.
- BOOCH, G. A.; RUMBAUGH, J., et al. *UML: el lenguaje unificado de modelado : guía del usuario. 2*, reimpresión ed. Pearson Educación, 2006. 527 p. Addison-Wesley Object Technology Series. ISBN



9788478290765.

- BOUZÁN MATANZA, J. M. .: *Ardora :: Creación de actividades escolares* [Web de ayuda de Ardora- Creación de contenidos para la web]. Última actualización: 2013/02/11/08:56:24. (webArdora.net). Disponible en: http://webardora.net/index_cas.html/files/155/index_cas.html.
- BRADENBAUGH, J. *Aplicaciones JavaScript*. Traducido por: O'reilly & Associates, I. ANAYA MULTIMEDIA (GRUPO ANAYA, S.A.), 2000 ed. Madrid: O'Reilly & Associates, Inc, 2000. 532 p. ISBN 84-41 5-1 070-9.
- CASTRO, M.; LOSADA, P., *et al. Comparación de Técnicas y Herramientas de Autor para la Generación de Aplicaciones Educativas*. DIEEC/UNED. 1997/09//de de 1997
- COLOMÉ CEDEÑO, I. D. M.; ESTRADA SANTÍ, D. V., *et al. Ambiente tecnológico para la creación de objetos de aprendizaje en apoyo al proceso docente de las universidades cubanas*. *ACIMED*, 2012/03/21/de de 2012, vol. 23, nº p. 116-129. [Consultado el: 2013/01/20/17:40:33]. Disponible en: <http://new.medigraphic.com/cgi-bin/resumenMain.cgi?IDARTICULO=35871> files/23/resumenMain.html. ISSN 1024-9435.
- CONNOLLY, T. M.; BOYLE, E. A., *et al. A systematic literature review of empirical evidence on computer games and serious games*. *Computers & Education*, 2012/09// 2012, vol. 59, nº 2, p. 661-686. Disponible en: <http://www.sciencedirect.com/science/article/pii/S0360131512000619>. ISSN 0360-1315.
- CHAVES, M. A. La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *Revista InterSedes*, 26 de julio de 2007 2006, vol. VI, nº p. 13. ISSN 1409-4746.
- DE ZAYAS, C. M. A. *La escuela en la vida*. Editorial Félix Varela, 1992. 186 p. Colección Educación y desarrollo. ISBN 9789590700156.
- DENVER, P. *PostgreSQL*. [Power Point]. publicado el: 2002/11/08/de de de 2002, última actualización: 2002/11/08/de de. Disponible en: http://www.slidefinder.net/q/qu%C3%A9_postgresql/32057312.
- DEVELOPERWORKS, I. *Introducción a XML*.
- E, A. B. C. *¿Qué es el estándar SCORM?* Última actualización: 2013/02/08/22:06:28. (e-learning sin límites). Disponible en: <http://www.e-abclearning.com/queesscormfiles/137/queesscorm.html>.
- EAVES, J. y GODFREY, R. J. W. *Apache Tomcat Bible*. Wiley India Pvt. Limited, 2003. 817 p. ISBN 9788126504305.
- EQUIPO DE DESARROLLO DE, H. *HAEduc* Última actualización: 2013/02/08/22:20:27. (HAEduc). Disponible en: <http://www.haeduc.rimed.cu/files/151/www.haeduc.rimed.cu.html>.



- FALCÓN GARCÍA, M. M. *Implementación del nivel A de IMS-LD para la herramienta de autor web CRODA*. Facultad 4. Universidad de las Ciencias Informáticas, 2011.
- FERNÁNDEZ, E. F. y HERNÁNDEZ, A. G. *Módulo de gestión de asignaturas de la herramienta educativa SMProg para el EVA - UCI*. Tutor: Ferreiro., L. D. S. y Chapman, I. a. H. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas, Facultad 10. Universidad de las Ciencias Informáticas, 2010.
- FERNÁNDEZ, J. M. S.; DEZA, M. C., et al. *Un enfoque arquitectónico para el desarrollo de software de gestión basado en Servicios Web*. En 28 de febrero de 2006.2006.
- FERNÁNDEZ, O. *“Haeduc, primera plataforma cubana de software libre para hacer multimedias” 2da Parte* « Última actualización: 2012/12/09/de de 2013/02/08/22:19:36. (Grupo de Usuarios de Tecnologías Libres). Disponible en: <http://gutl.jovencub.com/qaeduc-primer-plataforma-cubana-de-software-libre-para-hacer-multimediasq-2da-parte/files/149/qaeduc-primer-plataforma-cubana-de-software-libre-para-hacer-multimediasq-2da-parte.html>.
- FERREIRO, D. S. A. y TRUJILLO, R. T. *Representación tridimensional de edificaciones*. Tutor: Enríquez, D. a. M. Tesis de Diploma, Facultad de Matemática y Computación. Universidad de La Habana, 2005.
- FOUNDATION, A. S. *Apache Tomcat 7 User Guide*. Fultus Corporation, 2011. 254 p. Linbrary Advertising Club. ISBN 9781596822719.
- GÓMEZ ÁLVAREZ, I. M. C. *Definición de un método para el diseño de juegos orientados al desarrollo de habilidades gerenciales como estrategia de entrenamiento empresarial*. Tesis de Maestría, Universidad Nacional de Colombia, 2010.
- GOODMAN'S, D. *JavaScript® Bible, Gold Edition*. New York, NY: Hungry Minds, Inc., 2001. 2177 p. ISBN 0-7645-4718-6.
- GUERRA, Y. A. y GUERRA, Y. A. *Conceptualización de una red social educativa que integre de forma colaborativa las aplicaciones e-learning de la Universidad de las Ciencias Informáticas*. 2011 2011, vol. 5, nº 3, [Consultado el: 2013/01/28/21:50:52]. Disponible en: <http://rcci.uci.cu/index.php/rcci/article/view/144/files/53/144.html>.
- GUTIÉRREZ, J. J. ¿ Qué es un framework web? 2006 2006, nº
- GUZMAN, H. C. *Configurar NetBeans para usar Spring Framework con Maven* Última actualización: 2013/02/04/21:11:24. (Web personal de Hector Costa Guzman). Disponible en: <http://www.hcosta.info/wp/2011/09/configurar-netbeans-para-usar-spring-framework-con-maven/files/107/configurar-netbeans-para-usar-spring-framework-con-maven.html>.



- HECHAVARRÍA, R. A. y RIO, D. D. D. *Análisis, diseño e implementación de la versión 2.0 de los juegos del primer ciclo de la colección multisaber*. Tutor: Ing. Ismael Armando Nodarse Mora, I. a. M. R. R. Trabajo de Diploma, Facultad 3. Universidad de las Ciencias Informáticas, 2010.
- HOGAN, B. *HTML5 and CSS3*. Editado por: Pragmatic Programmers, L. Raleigh, North Carolina Dallas, Texas: The Pragmatic Bookshelf, 2010. 265 p. Develop with Tomorrow's Standards Today. ISBN 1-934356-68-9/ 978-1-934356-68-5.
- JACOBSON, I.; BOOCH, G., et al. *El proceso unificado de desarrollo de software*. La Habana: Félix Varela, 2004. The Addison-Wesley Object Technology Series.
- JAMES, R.; GARY, B., et al. *El proceso unificado de desarrollo de software*. ADDISON WESLEY ed. Madrid: 2000. 438 p. ISBN 8478290362.
- JAMES, R.; IVAR, J., et al. *El lenguaje unificado de modelado manual de referencia*. ADDISON WESLEY ed. Madrid: 2000. 526 p. ISBN 8478290370.
- JAVA, E. D. *Conozca más sobre la tecnología Java* Última actualización: 2013/05/07/15:09:41. Disponible en: <http://www.java.com/es/about/files/230/about.html>.
- JAVIER EGUÍLUZ, P. *Introducción a AJAX*. 1era ed. España: Creative Commons, 2009.
- JOAQUIN, G. *Patrones de diseño. Análisis y Diseño. Ingeniería del Software* Última actualización: 2005/05/27/de de 2013/05/07/15:31:30. Disponible en: <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.phpfiles/239/patrones-diseno.html>.
- JOHNSON, R. y HOELLER, J. *Expert one-on-one J2EE development without EJB*. ilustrada ed. Wrox, 2004. 552 p. Expert One-On-One Programmer to programmer. ISBN 0764558315.
- KHARE, T. *Apache Tomcat 7 Essentials*. Packt Publishing, 2012. 294 p. Community experience distilled. ISBN 9781849516631.
- LARMAN, C. *UML y patrones: introducción al análisis y diseño orientado a objetos*. Félix Varela, 2004.
- LEÓN, C. M. R. *Las esferas socioculturales del Software Libre* Última actualización: 2004 2013/05/07/15:40:48. (Derecho de Internet). Disponible en: <http://www.derecho-internet.org/node/293files/242/293.html>.
- LEONARDO, D. *Spring Framework - Dos Ideas* [wiki]. Última actualización: 2013/02/04/21:00:34. (Dos Ideas. personas y software). Disponible en: http://www.dosideas.com/wiki/Spring_Frameworkfiles/99/Spring_Framework.html.



- LÓPEZ ARROYO, D. M. *Herramientas de autor para el profesorado*. publicado el: 2001 2002 de 2001, última actualización: 2001 2002. 53-58 p. Disponible en: <http://dialnet.unirioja.es/servlet/articulo?codigo=205649>. ISBN 1136-7733.
- LLOYD, I. *The Ultimate HTML Reference*. United States of America: SitePoint Pty Ltd, 2008. 545 p. ISBN 978-0-9802858-8-8.
- MÁRQUEZ DENIS, Y. *Análisis y diseño del modulo IMS-Learning Design para CRODA*. Tutor: Ing. Dunia M. Colomé Cedeño, M. D. L. L. Trabajo de Diploma, Facultad 8. Universidad de las Ciencias Informáticas, 2010.
- MARTÍNEZ TABASCO, M. M. *Desarrollo de un módulo para la construcción colaborativa de objetos de aprendizaje en la herramienta de autor CRODA 2.0*. Tutor: Ing. Dunia María Colomé Cedeño, I. J. I. P., Ing. Osvaldo Ernesto Stable Vilches. Trabajo de Diploma, Universidad de las Ciencias Informáticas, 2012.
- MIRIAM, G.; MAYRA, Y., et al. *La herramienta informática Jclic vinculada en el proceso de enseñanza - aprendizaje del idioma inglés*. publicado el: 2009 de 2009, última actualización: 2009. vol. 6, Disponible en: <http://laboratorios.fi.uba.ar/lie/Revista/Articulos/060612/A1jul2009.pdf>. ISBN 1667-8338.
- MONCAYO, G. Curso Fundamentos de Programación en Java. nº
- MUSCIANO, C. y KEMEDY, B. *HTML. La guía completa*. Traducido por: Licenciada En Computacih Yazmh Ju5rez Parra, I. E. E. a. M. 2da. ed. Mexico, D.F: McCraw-Hill Companies, Inc, 1999. 546 p. Ayuda para autores de web. ISBN 1-56592-235.
- NARANJO, I. M. *Introducción a la Arquitectura de Software*. En Sep. 2003.2003.
- O'FARRILL, M. S. J. L. M. y DRA. ELSA HERRERO, T. *Las Herramientas de Autor en el proceso de producción de materiales educativos en formato digital*.
- OLSSON, T. y O'BRIEN, P. *The Ultimate CSS Reference*. United States of America: SitePoint Pty Ltd, 2008. 436 p. ISBN 978-0-9802858-5-7.
- ORACLE. *Introduction to Spring Web MVC - NetBeans IDE Tutorial* Última actualización: 2013/02/04/21:13:41. (NetBeans). Disponible en: <http://netbeans.org/kb/docs/web/quickstart-webapps-spring.html#files/111/quickstart-webapps-spring.html>.
- PARA, G. D. E. E. D. UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS GESTIÓN DE ESTRUCTURAS DIDÁCTICAS PARA EL DISEÑO DE OBJETOS DE APRENDIZAJE. nº



- PARADIGM, V. *Visual Paradigm for UML 10.1 Modeler Edition* Disponible en: <http://www.visual-paradigm.com/product/vpuml/editions/modeler.jsp>.
- PÉREZ, J. E. *Introducción a CSS*. 2007. 272 p.
- . *Introducción a JavaScript*. Incompleta ed. 2007.
- PERS, H. Guía de Metodologías Participativas y Juego Educativo. En *Programa de Formación Técnica Laboral para Jóvenes Bachilleres*. Bolivia. 2009. p. 70.
- PHILLIPS, P. B. A. *Sistemas de Informação*. McGraw Hill Brasil, 284 p. A. ISBN 9780073376837.
- PRESSMAN, R. S. A. y MURRIETA, J. E. M. *Ingeniería del software: un enfoque práctico*. 6 ed. Mcgraw Hill/Interamericana Editores, 2006. 958 p. ISBN 9789701054734.
- PROCESAMIENTO DEL LENGUAJE NATURA, S. E. *Procesamiento del Lenguaje Natura*. publicado el: 2008/03//de de 2008, última actualización: 2008/03//de. [Consultado el: 2012/11/21/]. vol. 40, 9-160 p. Disponible en: <http://www.sepln.org/revistaSEPLN/revista/40/todo.pdf>. ISBN 1135-5948.
- RAIBLE, M. *Spring Live*. Highlands Ranch, Colorado: SourceBeat, LLC, 2004. 353 p. ISBN 0974884340.
- RAZQUIN ZAZPE, P. Los sistemas de autor multimedia. *Servicio de Publicaciones de la Universidad Complutense de Madrid*, 1998/01/01/ 1998, vol. 8, nº 2, p. 127-139. Disponible en: <http://revistas.ucm.es/index.php/RGID/article/view/RGID9898220127A>. ISSN 1132-1873.
- ROD JOHNSON, J. H. K. D. C. S. R. H. T. R. A. A. D. D. D. K. M. P. T. T. E. V. P. T. B. H. A. C. J. *Spring Framework Reference Documentation* Última actualización: 2013/02/04/20:54:10. Disponible en: <http://static.springsource.org/spring/docs/current/spring-framework-reference/html/files/87/html.html>.
- ROLANDO ALFARO, T. *Educación*. Ministerio de Educación., 1994. 30-33 p. ISBN 0138-8029.
- SÁNCHEZ, J. *NetBeans 3.51 guía rápida*. publicado el: 2004 de 2004, última actualización: 2004. Disponible en: <http://es.scribd.com/doc/6615294/NetBeans>.
- SÁNCHEZ, J. y OLIVARES, R. Problem solving and collaboration using mobile serious games. *Computers & Education*, 2011/11// 2011, vol. 57, nº 3, p. 1943-1952. Disponible en: <http://www.sciencedirect.com/science/article/pii/S0360131511000935>. ISSN 0360-1315.
- SOMMERVILLE, I. y GALIPIENSO, M. I. A. *Ingeniería del software 7/e*. 7, ilustrada ed. Pearson, Addison Wesley, 2005. 687 p. Fuera de colección Out of series. ISBN 9788478290741.
- SPRINGSOURCE. *Part I. Overview of Spring Framework* Última actualización: 2013/02/04/20:55:57. (Springsource community). Disponible en: <http://static.springsource.org/spring/docs/current/spring->



framework-reference/html/spring-introduction.htmlfiles/89/spring-introduction.html.

SUÁREZ FERREIRO, D. *Herramienta para la creación y uso de juegos didácticos en la enseñanza de la Programación*. Tesis de Maestría, Departamento Técnicas de Programación Facultad 1. Universidad de las Ciencias Informáticas, 2011.

THE APACHE SOFTWARE, F. *Apache Tomcat* Última actualización: 1999 2013/05/02/19:13:21. (The Apache Software Foundation). Disponible en: <http://tomcat.apache.org/files/194/tomcat.apache.org.html>.

TRIANFAYLLAKOS, G.; PALAIGEORGIU, G., *et al.* Designing educational software with students through collaborative design games: The We!Design&Play framework. *Serious Games*, 2011/01// 2011, vol. 56, n^o 1, p. 227-242. Disponible en: <http://www.sciencedirect.com/science/article/pii/S036013151000223X>. ISSN 0360-1315.

WALLS, C. y BREIDENBACH, R. *Spring in Action*. Second ed. Greenwich,: Manning Publications Co., 2008. 765 p. ISBN 1-933988-13-4.

WHITE, A. *JavaScript® Programmer's Reference*. Canada: Wiley Publishing, Inc., 2009. 1035 p. ISBN 978-0-470-34472-9.

WOUTERS, P. y VAN OOSTENDORP, H. A meta-analytic review of the role of instructional support in game-based learning. *Computers & Education*, 2013/01// 2013, vol. 60, n^o 1, p. 412-425. Disponible en: <http://www.sciencedirect.com/science/article/pii/S0360131512001984>. ISSN 0360-1315.

WWW.LINUX.CIBERAULA.COM. *Una Introducción a APACHE* [Cursos para empresas]. Última actualización: 1997. (Ciberaula Linux). Disponible en: http://linux.ciberaula.com/articulo/linux_apache_intro/.

YUEN, P. K. y LAU, V. *Practical Web Technologies*. La Habana: Félix Varela, 2004. vol. I, 903 p.

ZAPATA, C. M.; PALACIO, C., *et al.* UNC-ANALISTA: HACIA LA CAPTURA DE UN CORPUS DE REQUISITOS A PARTIR DE LA APLICACIÓN DEL EXPERIMENTO MAGO DE OZ. *Revista EIA*, Junio 2007 2007, n^o p. 16. ISSN 1794-1237.

ZERQUERA, B. M. Z. y MILANÉS, Y. M. *Módulo SMProg para el Entorno Virtual de Aprendizaje de la Universidad de las Ciencias Informáticas*. Tutor: Ferreiro, L. D. S. y Lacal, L. D. M. Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas., Facultad 10. Universidad de las Ciencias Informáticas, 2010.



Glosario de términos

A continuación, en orden alfabético, se muestra el significado de algunos términos usados en este documento cuyo uso no es común y que pueden dificultar la comprensión del mismo:

A

AJAX: Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo Web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla.

Aplicación: programa informático que proporciona servicios de alto nivel al usuario, generalmente utilizando otros programas más básicos que se sitúan por debajo.

Applets: Es otra manera de incluir código a ejecutar en los clientes que visualizan una página Web. Se trata de pequeños programas hechos en Java, que se transfieren con las páginas Web y que el navegador ejecuta en el espacio de la página. Los applets de Java están programados en Java y pre-compilados, es por ello que la manera de trabajar de éstos varía un poco con respecto a los lenguajes de script como JavaScript. Los applets son más difíciles de programar que los scripts en JavaScript y requerirán unos conocimientos básicos o medios del lenguaje Java.

La principal ventaja de utilizar applets consiste en que son mucho menos dependientes del navegador que los scripts en JavaScript, incluso independientes del sistema operativo del ordenador donde se ejecutan. Además, Java es más potente que JavaScript, por lo que el número de aplicaciones de los applets podrá ser mayor.

B

Bases de datos: Es un conjunto integrado de datos junto con una serie de aplicaciones para su manejo, accesibles simultáneamente por diferentes usuarios y programas.



C

CASE (*ComputerAided Software Engineering*): son aplicaciones informáticas para aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

CU: Caso de Uso.

CUs: Casos de uso del sistema.

D

DOM (*Document Object Model*): es esencialmente una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.

E

E-Learning: es una manera flexible y poderosa mediante la cual individuos y grupos adquieren nuevos conocimientos y destrezas con apoyo de tecnología de redes de computadoras. Ésta permite diseminar y tener acceso a información multimedia, hacer uso de simuladores, al tiempo que permite interacción y colaboración con aprendices que pueden estar dispersos alrededor del mundo.

F

Framework: representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

H

Hardware: corresponde a todas las partes físicas y tangibles de una computadora.

HTML (*Hypertext Markup Language*): lenguaje de marcado para hipertextos, utilizado para la confección de páginas Web.

HTTPs (Protocolo seguro de transferencia de hipertexto): es un protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto.



I

IDE (*Integrated Development Environment*): 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.

Interfaz: En informática, una interfaz es la parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario. En software también se habla de interfaz gráfica de usuario, que es un método para facilitar la interacción del usuario con el ordenador o la computadora a través de la utilización de un conjunto de imágenes y objetos pictóricos (iconos, ventanas, formularios, páginas Web...).

J

JavaScript: es un lenguaje basado en objetos, utilizado para acceder a objetos en aplicaciones, se utiliza integrado en un navegador Web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas Web dinámicas.

JQuery: consiste en un conjunto de librerías en un único archivo de varios kilobytes que permite enriquecer estéticamente una página web. Proporciona funciones listas para usar, que permite facilitar y estandarizar la creación de aplicaciones, además posibilita obtener información de los navegadores. Es compatible con los navegadores más usados a nivel internacional como Firefox desde su versión 3.6, Internet Explorer desde la versión 6.0, Safari desde la versión 5.0, Opera y Google Chrome

JSP: es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es, pues, una tecnología orientada a crear páginas Web con programación en Java. Con JSP se pueden crear aplicaciones web que se ejecuten en variados servidores Web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Por tanto, las JSP pueden ser escritas con el editor HTML/XML habitual.

L

LDAP: Protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.



Lenguaje: En informática, cuando se habla de lenguaje se refiere generalmente al de programación, conjunto de instrucciones que las aplicaciones necesitan para que el ordenador ejecute determinadas operaciones. Hay lenguaje de alto y bajo nivel, de tercera y cuarta generación, lenguaje natural y lenguaje máquina.

Logs: registro oficial de eventos durante un rango de tiempo en particular. Para los profesionales en seguridad informática es usado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué un evento ocurre para un dispositivo en particular o aplicación.

LOM: es un modelo de datos, usualmente codificado en XML, usado para describir un objeto de aprendizaje y otros recursos digitales similares usados para el apoyo al aprendizaje.

M

MOODLE: 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*). 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.

Multiplataforma: es un término usado para referirse a los programas, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

P

Precondición: Restricción que ha de ser cierta cuando una operación es invocada.

Poscondición: Restricción que ha de ser cierta al completarse una operación.

R

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



S

SCROM: es un conjunto de estándares y especificaciones que permite crear objetos pedagógicos estructurados.

Servidor: Sistema que proporciona recursos (por ejemplo, servidores de ficheros, servidores de nombres). En Internet este término se utiliza muy a menudo para designar a aquellos sistemas que proporcionan información a los usuarios de la red.

Servlets: son componentes de la parte del servidor de Java EE, encargados de generar respuestas a las peticiones recibidas de los clientes.

Software: se refiere al equipamiento lógico o soporte lógico de una computadora digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de tareas específicas.

Software libre: Es la denominación del software que brinda libertad a los usuarios sobre un producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

SQL (*Structured Query Language*): es un lenguaje declarativo de acceso a bases de datos relacionales.

U

UCI: Universidad de las Ciencias Informáticas.

UML: El Lenguaje Unificado de Modelado (*Unified Modeling Language* en inglés) prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

W

Web: sistema para presentar información en internet basado en hipertexto.



X

XML (*Extensible Markup Language*): es un metalenguaje que nos permite definir lenguajes de marcado adecuados a usos determinados. Es un estándar internacionalmente reconocido y no pertenece a ninguna compañía por lo que su utilización es libre.



Anexos

Anexo #1. Tipos de Juegos Didácticos.

Se han diseñado los siguientes tipos de tableros, los cuales definen el comportamiento (propiedades) que pueden tener las casillas ubicadas en el mismo.

- ♣ Sube y Baja: el tablero consta de casillas en las que se permanece, avanza o retrocede hacia otra casilla ubicada en el mismo.
- ♣ Avanzar hacia la meta: en este tipo de tablero las casillas no tienen asociado ningún comportamiento especial. Simplemente se avanza hacia la meta (hacia adelante) según el número obtenido en el lanzamiento de los dados.
- ♣ Bingo: las casillas de este tablero carecen de un comportamiento particular, para poder concluir el juego es necesario que se complete una cantidad prefijada de casillas contiguas horizontal, vertical o diagonalmente.

En el tablero se pueden incluir casillas con comportamiento de obstáculos, que pueden ser los comunes en los juegos estilo tablero, como son: “el equipo puede avanzar o retroceder cierto número de espacios”, “ir hacia la meta o la salida”, “no jugar en varias rondas”, “esperar para continuar el juego una cantidad de turnos determinada por los dados”, etc.

Existirán además las casillas Didácticas, las cuales estarán señalizadas (*, ¡! , ¿?) y responden a la siguiente clasificación:

*: Preguntas relacionadas con la actualidad nacional e internacional, efemérides u otra asignatura (permite interdisciplinariedad).

¡! : Preguntas del sistema de clases que se está ejercitando.

¿?: Preguntas de materias regresivas.



Anexo #2. Descripción detallada de los Casos de Uso del sistema.

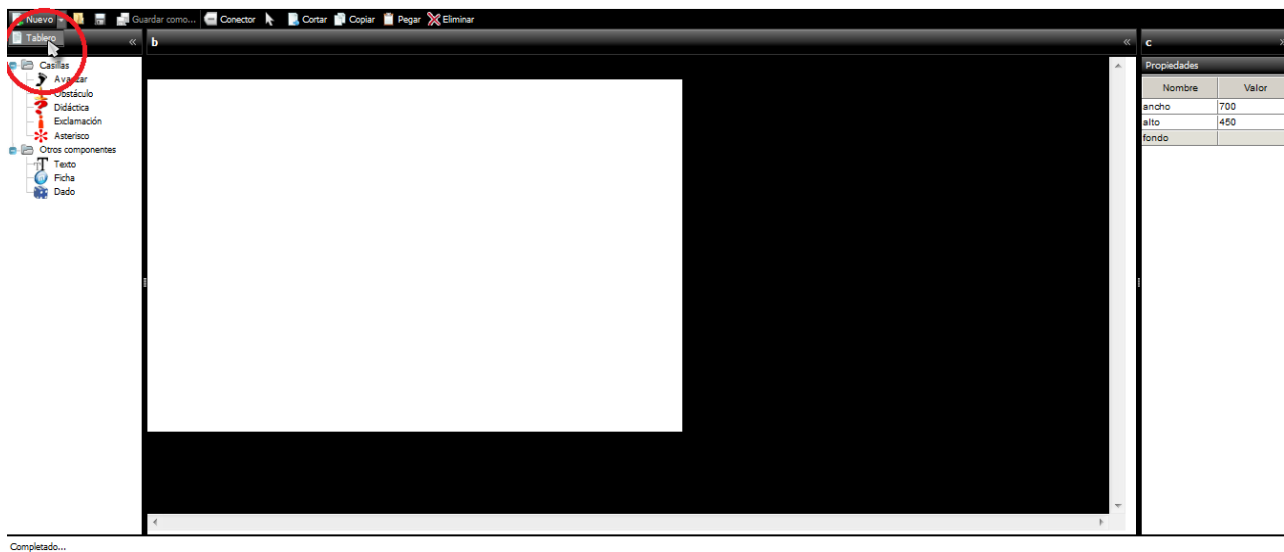
Tabla 18: Descripción detallada del CUs Gestionar tablero.

CUS-1	Gestionar tablero	
Propósito	Permitir crear, eliminar, modificar y mostrar tablero.	
Actores	Profesor	
Resumen:	El usuario accede a las opciones que le permiten crear un tipo de tablero de juego didáctico, modificar el tablero, eliminarlo o cargar uno que ya este previamente guardado. El tablero puede ser salvado o actualizado una vez creado o abierto previamente.	
Precondiciones	El usuario debe estar registrado como profesor y tener los permisos necesarios para realizar las acciones. No debe existir ningún tablero abierto en la aplicación.	
Referencias	RF 1, RF 2, RF 3, RF 4	
Prioridad	Crítico	
Poscondiciones	El tablero diseñado es guardado en un fichero con una estructura determinada.	
Sección: "General"		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El profesor selecciona en el menú superior "Archivo".	2. La herramienta muestra varias opciones: a. Nuevo tablero b. Abrir tablero	
3. El profesor selecciona la opción deseada.	Según la selección realizada la herramienta muestra: 4. Si el profesor seleccionó la opción Nuevo tablero, ver sección: "Crear nuevo tablero". 5. Si el profesor seleccionó la opción Abrir tablero, ver sección: "Cargar tablero".	
Sección: "Crear nuevo tablero"		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	



<p>4. El profesor selecciona la opción Nuevo tablero.</p>	<p>4.1 La herramienta muestra:</p> <ol style="list-style-type: none"> Nombre Tipo de tablero a crear (Sube y Baja, Avanzar hacia la meta y Bingo) Dimensiones (alto y ancho) Descripción
<p>4.2 El profesor introduce un nombre, una descripción para el tablero, selecciona el tipo de tablero y las dimensiones que va a tener.</p>	<p>4.3 La herramienta crea un nuevo tablero y almacena los datos del mismo.</p> <p>4.4 La herramienta muestra las opciones relacionadas con insertar las casillas, el texto, las imágenes, las fichas y el dado dentro del tablero.</p> <p>4.5 La herramienta activa la opción salvar como, ver sección “Guardar como”.</p> <p>4.6 La herramienta activa la opción salvar, ver sección “Guardar”.</p> <p>4.7 La herramienta activa el panel de propiedades, ver sección: “Modificar”</p>

Prototipo de Interfaz de Usuario



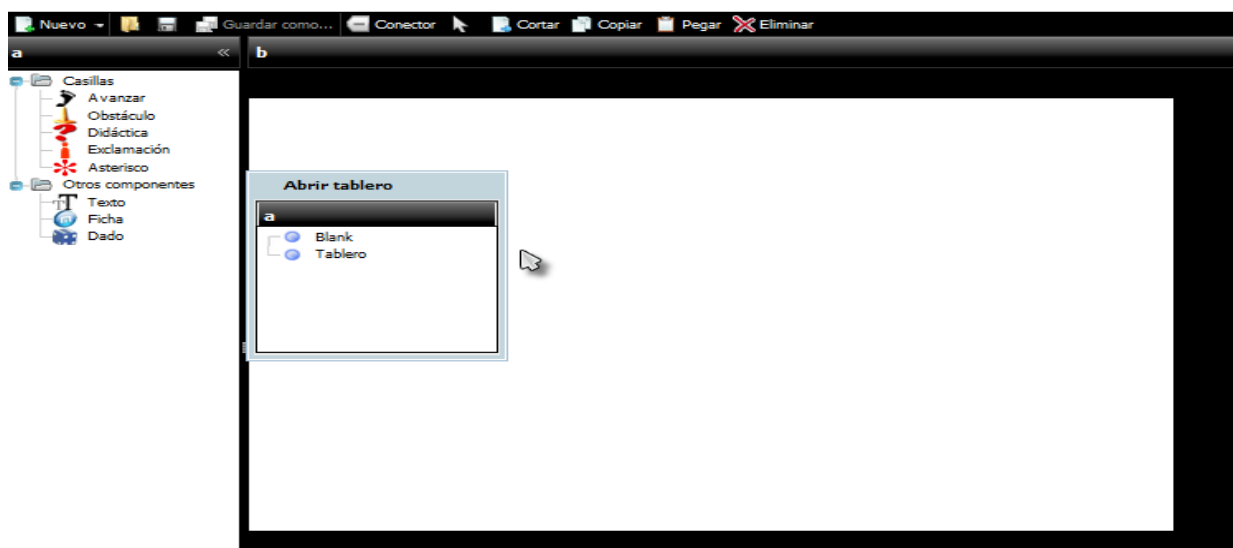
Sección: “Cargar tablero”

Flujo Normal de Eventos



Acción del Actor	Respuesta del Sistema
5. El profesor selecciona la opción Abrir tablero.	5.1 La herramienta permite seleccionar la ruta en la que se encuentra el tablero deseado.
5.2 El profesor selecciona el tablero.	5.3 La herramienta abre el tablero y muestra las opciones relacionadas con modificar y eliminar las casillas, el texto, las imágenes, las fichas y el dado dentro del tablero.

Prototipo de Interfaz de Usuario



Sección: "Guardar como"

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
6. El profesor selecciona la opción Guardar como.	6.1 La herramienta permite seleccionar el directorio y definir el nombre del tablero (fichero).
6.2 El profesor escoge la dirección y define un nombre.	6.3 La herramienta guarda el tablero en la ruta especificada.

Flujos Alternativos

Acción del Actor	Respuesta del Sistema
------------------	-----------------------

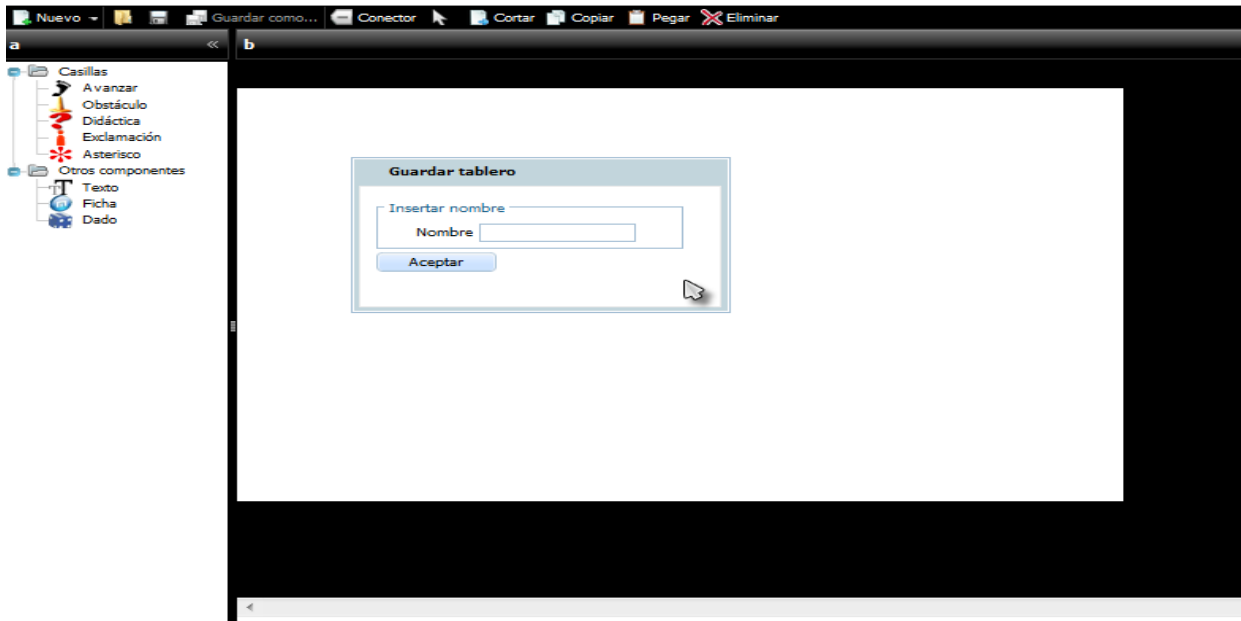


6.2 El profesor escoge la dirección.

6.3 La herramienta muestra un mensaje de error y realiza la petición de introducir un nombre para el tablero.

6.4 Se retorna la paso 6.1 del flujo normal de los eventos.

Prototipo de Interfaz de Usuario



Sección: "Modificar tablero"

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
7.1 El profesor selecciona el tablero en el que está trabajando.	7.1 La herramienta activa una paleta de propiedades con las siguientes opciones: a. Dimensiones b. Alto y Ancho c. Fondo
7.2 El profesor cambia alguno de los valores que les muestra la aplicación.	7.3 La herramienta actualiza los datos del tablero.

Prototipo de Interfaz de Usuario

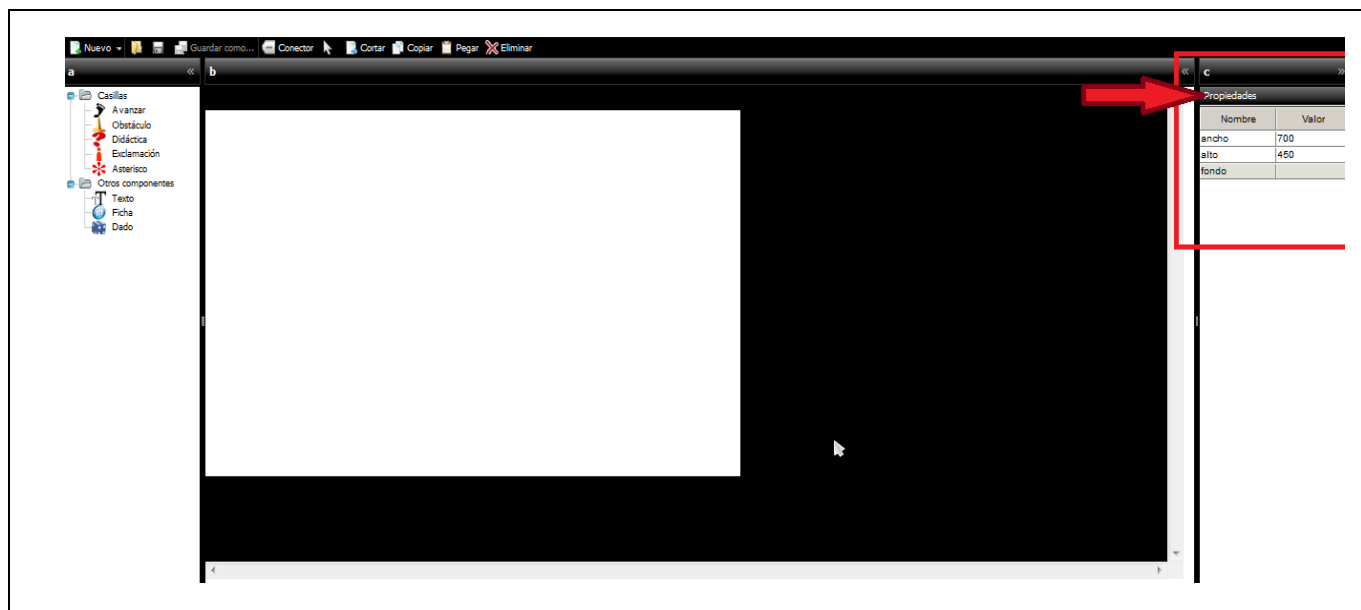


Tabla 19: Descripción detallada del CUs Gestionar casilla.

CUS-2	Gestionar casilla
Propósito	Permitir crear, modificar, eliminar y mostrar casilla.
Actores	Profesor
Resumen:	El usuario puede: seleccionar el tipo de casilla que desea insertar, ya sea de avanzar, de obstáculo o didáctica y seleccionar el tipo de casilla que desea modificar, estas pueden ser transformadas, eliminadas o duplicadas.
Precondiciones	El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe haber creado o cargado un tablero en la aplicación. El tablero debe estar creado y debe existir al menos una casilla insertada.
Referencias	RF 5, RF 6, RF 7, RF 8
Prioridad	Crítico
Poscondiciones	Las casillas diseñadas se guardan en el fichero en el que se encuentra el juego ya previamente diseñado. El sistema almacena los cambios hechos a las casillas en el tablero.
Sección: "General"	
Flujo Normal de Eventos	



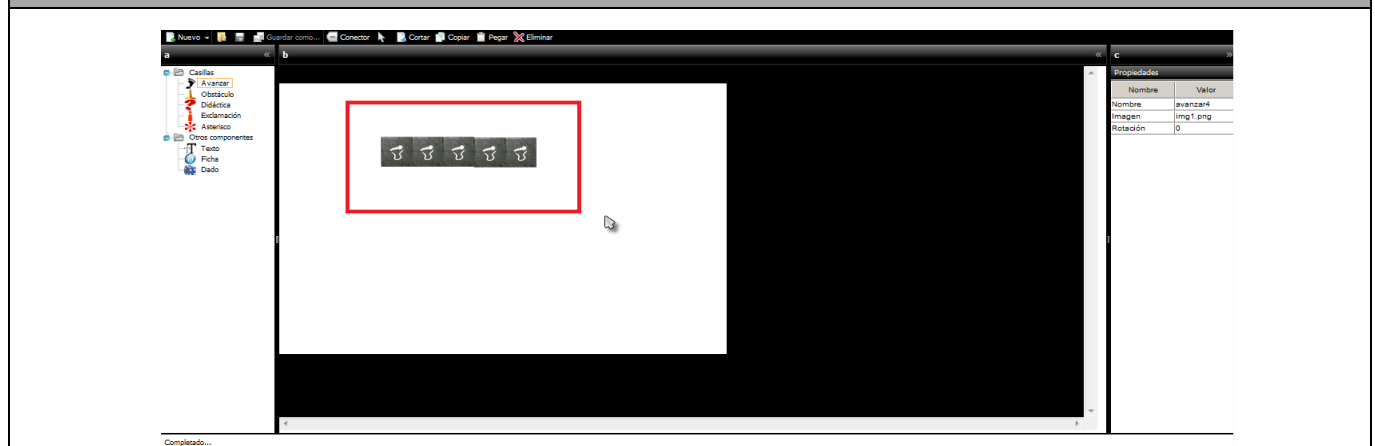
Acción del Actor	Respuesta del Sistema
1. El profesor decide la casilla que desea insertar.	2. La herramienta muestra varias opciones: a. casilla avanzar b. casilla obstáculo c. casilla didáctica
3. El profesor selecciona la casilla que desea.	4. Según la selección realizada la herramienta muestra: 4.1 Si selecciona la casilla avanzar, ver sección: "Avanzar". 4.2 Si selecciona la casilla obstáculo, ver sección: "Obstáculo". 4.3 Si selecciona la casilla didáctica, ver sección: "Didáctica".
6. El profesor mantiene seleccionada la casilla.	7. La herramienta posibilita las siguientes opciones: 7.1 Cambiar alguna de sus propiedades, ver sección: "Modificar casilla". 7.2 Borrar una casilla, ver sección: "Eliminar casilla".

Sección: "Avanzar"

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
4.1 El profesor selecciona el componente avanzar y lo arrastra hacia el tablero.	4.1.1 La herramienta inserta dentro del tablero en el lugar donde se da un clic una casilla con un icono (Pie).

Prototipo de Interfaz de Usuario





Sección: "Obstáculo"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
4.2 El profesor selecciona el componente obstáculo y lo arrastra hacia el tablero.	4.2.1 La herramienta muestra las propiedades que va a contener la casilla: <ul style="list-style-type: none"> a. Seleccionar obstáculo: consiste en seleccionar la condición que va a realizar el jugador al llegar a esa casilla. b. Número: corresponde al número de espera al llegar a esa casilla. c. Descripción: una pequeña explicación de lo que consiste la acción aplicada en la casilla.
4.2.2 El profesor selecciona las propiedades y presiona el botón Aceptar.	4.2.3 La herramienta inserta la casilla con un icono (cono).

Prototipo de Interfaz de Usuario

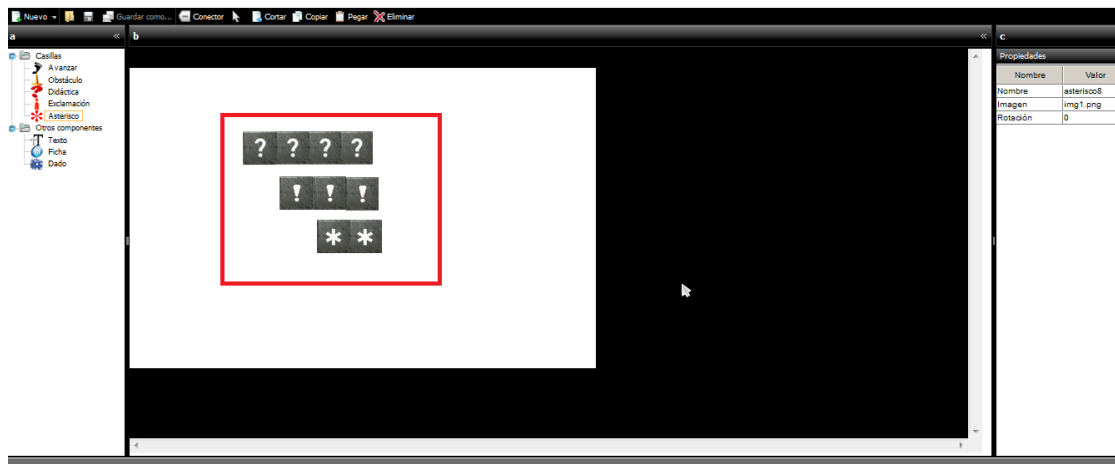


Sección: "Didáctica"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema



<p>4.3 Si el profesor selecciona el componente didáctica y lo arrastra hacia el tablero.</p>	<p>4.3.1 La herramienta muestra una ventana donde se puede seleccionar el tema de pregunta al que va a estar asociada la casilla:</p> <ul style="list-style-type: none"> a. Seleccionar pregunta: contiene tres tipos de temas de preguntas. b. Descripción: breve explicación de lo que consiste el tema seleccionado.
<p>4.3.2 El profesor selecciona las propiedades y presiona el botón Aceptar.</p>	<p>4.3.3 Según la selección realizada la herramienta inserta en el tablero:</p> <ul style="list-style-type: none"> a. Si selecciona el primer tema de pregunta, la casilla se muestra con el icono interrogación “?”. b. Si selecciona el segundo tema de pregunta, la casilla se muestra con el icono exclamación “!”. c. Si selecciona el tercer tema de pregunta, la casilla se muestra con el icono asterisco “*”.

Prototipo de Interfaz de Usuario



Sección: “Modificar casilla”

Flujo Normal de Eventos

<p>Acción del Actor</p>	<p>Respuesta del Sistema</p>
--------------------------------	-------------------------------------



<p>7.1 El profesor selecciona con un clic del mouse una casilla ya puesta dentro del tablero.</p>	<p>7.1.1 La herramienta muestra en el lado derecho un panel de propiedades con las opciones que pueden ser modificadas sin tener en cuenta el tipo de casilla seleccionada.</p> <ul style="list-style-type: none"> a. Posición dentro del tablero (x, y) b. Dimensiones (alto y ancho) c. Rotación (ángulos en los que puede rotar)
<p>7.1.2 El profesor cambia alguna de las propiedades ya predeterminadas.</p>	<p>7.1.3 La herramienta muestra los cambios realizados.</p>

Flujos Alternativos

Acción del Actor	Respuesta del Sistema
<p>7.1 Si el profesor selecciona con el mouse una casilla ya puesta dentro del tablero y da clic derecho encima de ella.</p>	<p>7.1.1 La herramienta muestra un menú con dos opciones:</p> <ul style="list-style-type: none"> a. Transformar: permite cambiar la forma de la casilla seleccionada. b. Fondo: la herramienta muestra una ventana con todas las opciones de fondos que pueden tener las casillas con el icono que le corresponde según la seleccionada.
<p>7.1.2 El profesor selecciona otra forma u otro fondo para la casilla.</p>	<p>7.2.3 La herramienta realiza y muestra los cambios indicados.</p>

Prototipo de Interfaz de Usuario

The prototype displays a dialog box titled "Seleccione la imagen de fondo" with a dropdown menu showing "Fondo Avanzar" and "Gris redondeado", and an "Aceptar" button. To the right, three "Propiedades" panels are shown, each with a table of attributes:

Nombre	Valor
Nombre	didactica3
Imagen	img1.png
Rotación	0

Nombre	Valor
Nombre	obstaculo9
Castigo	Seleccionar
Rondas	2
Imagen	img1.png
Rotación	0

Nombre	Valor
Nombre	avanzar10
Imagen	img1.png
Rotación	0



Sección: "Eliminar casilla"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
7.2 El profesor selecciona con el mouse una casilla dentro del tablero. 7.2.1 Selecciona en el menú principal la opción "Eliminar".	7.2.2 La herramienta muestra una interfaz para confirmar la acción indicada.
7.2.3 El profesor selecciona la opción "Aceptar".	7.2.4 La herramienta elimina la casilla seleccionada.
Prototipo de Interfaz de Usuario	

Tabla 20: Descripción detallada del CUs Gestionar ficha.

CUS-3	Gestionar ficha
Propósito	Permitir seleccionar, eliminar o modificar ficha.
Actores	Profesor
Resumen:	El usuario 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.
Precondiciones	El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe haber creado o cargado un tablero antes de comenzar el caso de uso.
Referencias	RF 9, RF 10, RF 11, RF 12
Prioridad	Secundario
Poscondiciones	Las fichas seleccionadas se guardan en el fichero en el que se encuentra el tablero ya previamente diseñado y actualiza la lista de fichas a seleccionar según las opciones realizadas.
Sección: "General"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema



1. El profesor selecciona el componente estrella.	2. La herramienta muestra un menú con el ítem ficha donde se podrá escoger una ficha.
3. El profesor selecciona la ficha deseada.	4. La herramienta inserta la ficha en el panel inferior izquierdo del tablero al lado del dado.
5. El profesor selecciona con un clic izquierdo del mouse la ficha que desea gestionar.	7. Regresar al paso 2 del flujo normal de los eventos.
6. Regresar al paso 1 del flujo normal de eventos.	
8. Regresar al paso 3 del flujo normal de eventos.	9. La herramienta realiza el intercambio y muestra la nueva ficha.

Flujos Alternativos

Acción del Actor	Respuesta del Sistema
5 El profesor selecciona con un clic izquierdo del mouse la ficha que desea gestionar.	7 La herramienta muestra un mensaje de confirmación de la acción.
6 El profesor selecciona en el menú Editar la opción "Eliminar".	
8 El profesor selecciona la opción "Aceptar"	9 La herramienta elimina la ficha seleccionada.

Prototipo de Interfaz de Usuario

The screenshot displays the user interface for the didactic game board design tool. It is divided into three main sections:

- Properties Panel (c):** A table showing the properties of a selected element.

Nombre	Valor
Nombre	ficha0
Imagen	img1.png
- Menu (a):** A vertical list of tool options including 'Casillas', 'Avanzar', 'Obstáculo', 'Didáctica', 'Exclamación', 'Asterisco', 'Otros componentes', 'Texto', 'Ficha', and 'Dado'. The 'Ficha' option is currently selected.
- Game Board (b):** A large white rectangular area representing the game board. A red angry face icon (the 'Dado' or die) is positioned at the bottom left of the board.

Below the properties panel, there is a dialog box titled 'Seleccione ficha' (Select piece) with a dropdown menu showing 'Fondo ficha: Yin-Yan' and an 'Aceptar' (Accept) button.



Tabla 21: Descripción detallada del CUs Gestionar texto.

CUS-4	Gestionar texto	
Propósito	Permitir insertar, modificar, mostrar o eliminar texto.	
Actores	Profesor	
Resumen:	El usuario introduce un texto en el tablero. Los textos pueden ser modificados o eliminados una vez insertados.	
Precondiciones	El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe haber creado o abierto un tablero (fichero) en la aplicación.	
Referencias	RF 13, RF 14, RF 15, RF 16	
Prioridad	Secundario	
Poscondiciones	El texto insertado se guarda en un nuevo tablero. El texto modificado se actualiza en el tablero que se encuentra previamente diseñado.	
Sección: "General"		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El profesor selecciona el componente texto.	2. La herramienta le permite realizar varias opciones: a. Insertar texto b. Modificar texto c. Eliminar texto	
3. El profesor selecciona la opción deseada.	4. La herramienta si el profesor decide realizar la opción insertar, ver sección: "Insertar texto". 5. La herramienta si el profesor decide realizar la opción modificar, ver sección "Modificar texto". 6. La herramienta si el profesor decide realizar la opción eliminar, ver sección: "Eliminar texto".	
Sección: "Insertar texto"		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	



<p>8.El profesor realiza la opción “Insertar texto” arrastrando el componente hacia dentro del tablero.</p>	<p>4.1 La herramienta crea en el tablero un cuadro de texto.</p> <p>4.2 La herramienta muestra el cuadro de texto activado para insertar el texto.</p> <p>4.3 La herramienta guarda los cambios realizados.</p>
Prototipo de Interfaz de Usuario	
Sección: “Modificar texto”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>5 El profesor selecciona con el mouse un cuadro de texto ya puesto dentro del tablero.</p>	<p>5.1 La herramienta muestra en el lado derecho un panel de propiedades con las opciones que pueden ser modificadas:</p> <ul style="list-style-type: none"> a. Posición dentro del tablero (x, y) b. Alineación (para posicionar la casilla) c. Fuente (tipo y tamaño)
<p>5.2 El profesor cambia alguna de las propiedades ya predeterminadas.</p>	<p>5.3 La herramienta actualiza y muestra los cambios realizados.</p>
Prototipo de Interfaz de Usuario	
Sección: “Eliminar texto”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>6 El profesor selecciona con el mouse un cuadro de texto dentro del tablero.</p> <p>6.1 El profesor selecciona en el menú Editar la opción “Eliminar”.</p>	<p>6.2 La herramienta solicita la autorización para realizar la acción indicada.</p>
<p>6.3 El profesor confirma la acción.</p>	<p>6.4 La herramienta elimina el cuadro de texto.</p>
Flujos Alternativos	



Acción del Actor	Respuesta del Sistema
6.3 El profesor niega la acción.	6.4 La herramienta no realiza ningún cambio.
Prototipo de Interfaz de Usuario	

Tabla 22: Descripción detallada del CUs Insertar imagen.

CUS-5	Insertar imagen
Propósito	Permitir insertar imagen.
Actores	Profesor
Resumen:	El usuario puede introducir cualquier imagen dentro del tablero.
Precondiciones	El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe haber creado o cargado un tablero antes de comenzar el caso de uso.
Referencias	RF 17
Prioridad	Auxiliar
Poscondiciones	La imagen escogida queda guardada en el tablero.
Sección: "General"	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1.El profesor selecciona el componente imagen.	2. La herramienta muestra una interfaz para que el profesor seleccione la dirección de donde quiere cargar la imagen que desea insertar.
3.El profesor selecciona la imagen que desea cargar.	4. La herramienta muestra la imagen cargada y activa la opción eliminar. (ver CUs Eliminar Imagen)
Prototipo de Interfaz de Usuario	



Tabla 23: Descripción detallada del CUs Eliminar imagen.

CUS-6	Eliminar imagen	
Propósito	Permitir eliminar imagen.	
Actores	Profesor	
Resumen:	El usuario puede eliminar cualquier imagen que se encuentre dentro del tablero en caso de no ser requerida en el diseño de tableros de juegos didácticos.	
Precondiciones	El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe haber creado o cargado un tablero antes de comenzar el caso de uso. El usuario tiene que haber insertado una imagen previamente en el tablero.	
Referencias	RF 18	
Prioridad	Auxiliar	
Poscondiciones		
Sección: "General"		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
5.El profesor selecciona con el mouse una imagen dentro del tablero. 6.Selecciona en el menú principal la opción "Eliminar".	7. La herramienta solicita la autorización para realizar la acción indicada.	
8.El profesor confirma la acción.	9. La herramienta elimina la imagen seleccionada.	
Flujos Alternativos		
Acción del Actor	Respuesta del Sistema	
4. El profesor cancela la acción.	5. La herramienta no realiza ningún cambio.	
Prototipo de Interfaz de Usuario		

Tabla 24: Descripción detallada del CUs Insertar dado.

CUS-7	Insertar dado
Propósito	Permitir insertar un dado.
Actores	Profesor



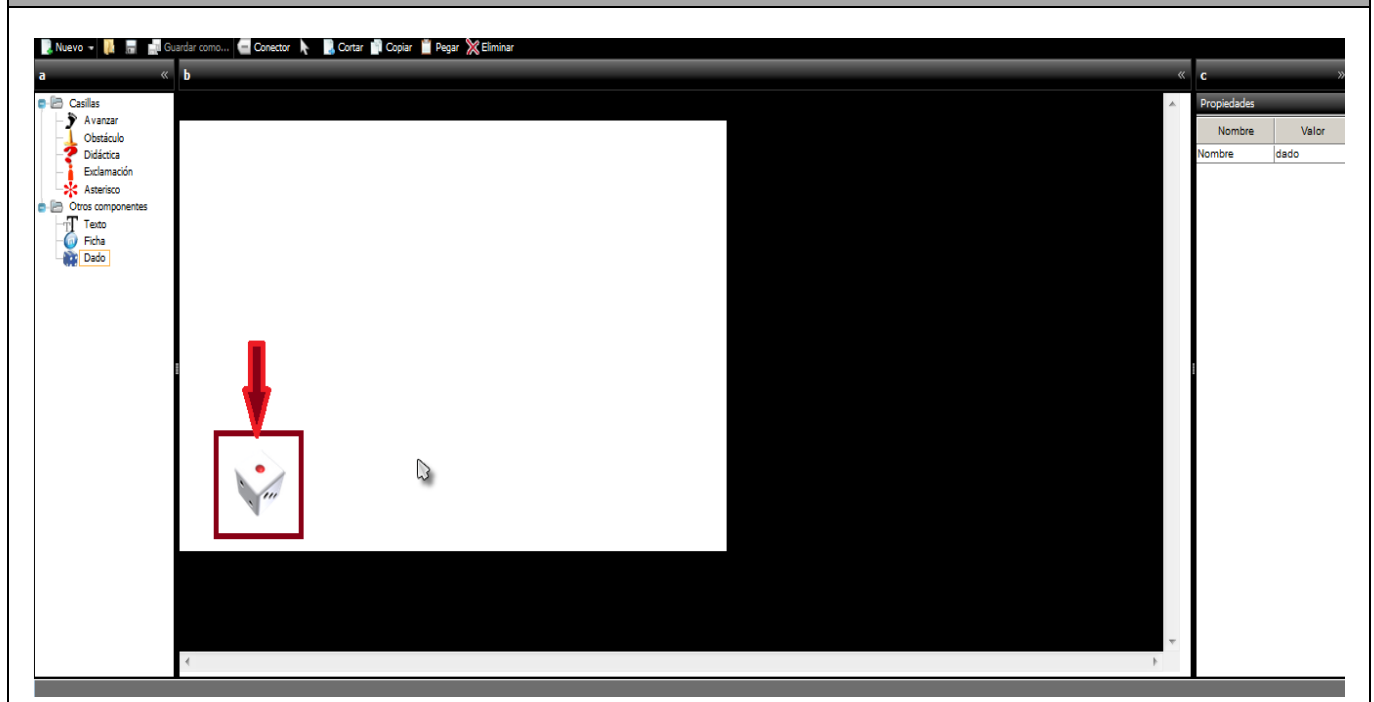
Resumen:	El usuario puede introducir un dado dentro del tablero, siempre y cuando este sea de tipo Avanzar hacia la meta o Sube y Baja.
Precondiciones	El usuario debe estar autenticado en el Sistema como Profesor. El usuario debe crear un tablero antes de comenzar el caso de uso.
Referencias	RF 19
Prioridad	Auxiliar
Poscondiciones	La imagen escogida queda guardada en el tablero.

Sección: "General"

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
10. El profesor selecciona el componente estrella.	11. La herramienta muestra un menú con el ítem dado donde se podrá escoger el dado.
12. El profesor selecciona el dado.	13. La herramienta inserta el dado en el panel inferior izquierdo del tablero, antes de las fichas.

Prototipo de Interfaz de Usuario





Anexo #3. Diagrama de clases del diseño.



Ilustración 7: Diagrama de clases de la herramienta DEGames Board.

Anexo #4. Estructura del fichero XML.

Para que un documento XML esté bien formado debe regirse por las siguientes reglas:

1. La declaración del documento debe encabezar al mismo.
2. Los elementos que contengan datos deben tener las etiquetas de comienzo y final.



3. Los elementos que no contengan datos y sean usados como etiquetas simples deben terminar con />.
4. Todas las etiquetas deben estar correctamente cerradas.
5. El documento debe contener exactamente un elemento que contenga completamente todos los demás, es decir, que exista una etiqueta raíz.
6. Las etiquetas deben estar correctamente anidadas.
7. Una etiqueta no puede tener dos atributos con el mismo nombre.
8. Los valores de los atributos deben ser citas (00 o "").
9. Los valores <, > y & solamente pueden ser utilizados como etiquetas y referencias a entidades respectivamente.

Anexo #5. Ejecución de los casos de prueba.

Tabla 25: SC1_Crear_tablero_CU1.

ID-EC	Escenario	Nombre	Tipo	Alto	Ancho	1era iteración	2da iteración	3era iteración
1	EC 1.1	v	v	v	v	No satisfactorio	No satisfactorio	Satisfactorio
2	EC 1.2	v	v	v	v	No satisfactorio	No satisfactorio	Satisfactorio
3	EC 1.3	v	v	v	v	No satisfactorio	No satisfactorio	Satisfactorio

Tabla 26: SC2_Cargar_tablero_CU1.

ID-EC	Escenario	Tablero	1era iteración	2da iteración	3era iteración
1	EC 2.1	v	No satisfactorio	Satisfactorio	Satisfactorio
2	EC 2.2	v	No satisfactorio	Satisfactorio	Satisfactorio
3	EC 2.3	v	No satisfactorio	No satisfactorio	Satisfactorio

Tabla 27: SC3_Guardar_tablero_CU1.

ID-EC	Escenario	Nombre	1era iteración	2da iteración	3era iteración
-------	-----------	--------	----------------	---------------	----------------



1	EC 3.1	v	Satisfactorio	Satisfactorio	Satisfactorio
2	EC 3.2	v	No satisfactorio	No satisfactorio	Satisfactorio

Tabla 28: SC4_Modificar_tablero_CU1.

ID-EC	Escenario	Fondo	Alto	Ancho	1era iteración	2da iteración	3era iteración
1	EC 4.1	-	v	-	Satisfactorio	Satisfactorio	Satisfactorio
2	EC 4.2	-	-	v	No satisfactorio	No satisfactorio	Satisfactorio
3	EC 4.3	v	-	-	Satisfactorio	Satisfactorio	Satisfactorio
4	EC 4.4	v	-	-	No satisfactorio	No satisfactorio	Satisfactorio

Tabla 29: SC1_Avanzar_CU2.

ID-EC	Escenario	1era iteración	2da iteración	3era iteración
1	EC 1.1	Satisfactorio	Satisfactorio	Satisfactorio
2	EC 1.2	Satisfactorio	Satisfactorio	Satisfactorio

Tabla 30: SC2_Obstáculo_CU2.

ID-EC	Escenario	1era iteración	2da iteración	3era iteración
1	EC 2.1	Satisfactorio	Satisfactorio	Satisfactorio
2	EC 2.2	Satisfactorio	Satisfactorio	Satisfactorio

Tabla 31: SC3_Didáctica_CU2.

ID-EC	Escenario	1era iteración	2da iteración	3era iteración
1	EC 3.1	Satisfactorio	Satisfactorio	Satisfactorio
2	EC 3.2	Satisfactorio	Satisfactorio	Satisfactorio
3	EC 3.3	Satisfactorio	Satisfactorio	Satisfactorio
4	EC 3.4	Satisfactorio	Satisfactorio	Satisfactorio



5	EC 3.5	Satisfactorio	Satisfactorio	Satisfactorio
6	EC 3.6	Satisfactorio	Satisfactorio	Satisfactorio

Tabla 32: SC4_Modificar_casilla_CU2.

ID-EC	Escenario	Fondo	Angulo	Ronda	Castigo	1era iteración	2da iteración	3era iteración
1	EC 4.1	v	-	-	-	Satisfactorio	Satisfactorio	Satisfactorio
2	EC 4.2	v	-	-	-	No satisfactorio	Satisfactorio	Satisfactorio
3	EC 4.3	-	v	-	-	Satisfactorio	Satisfactorio	Satisfactorio
4	EC 4.4	-	-	v	-	Satisfactorio	Satisfactorio	Satisfactorio
5	EC 4.5	-	-	-	v	Satisfactorio	Satisfactorio	Satisfactorio
6	EC 4.6	-	-	-	v	No satisfactorio	Satisfactorio	Satisfactorio

Tabla 33: SC5_Eliminar_casilla_CU2.

ID-EC	Escenario	1era iteración	2da iteración	3era iteración
1	EC 5.1	No satisfactorio	Satisfactorio	Satisfactorio

Tabla 34: SC1_Insertar_ficha_CU3.

ID-EC	Escenario	1era iteración	2da iteración	3era iteración
1	EC 1.1	No satisfactorio	Satisfactorio	Satisfactorio
2	EC 1.2	No satisfactorio	Satisfactorio	Satisfactorio

Tabla 35: SC2_Modificar_ficha_CU3.

ID-EC	Escenario	Fondo	1era iteración	2da iteración	3era iteración
1	EC 2.1	V	No satisfactorio	Satisfactorio	Satisfactorio
2	EC 2.2	v	No satisfactorio	No satisfactorio	Satisfactorio



Tabla 36: SC3_Eliminar_ficha_CU3.

ID-EC	Escenario	1era iteración	2da iteración	3era iteración
1	EC 3.1	No satisfactorio	Satisfactorio	Satisfactorio

Tabla 37: SC1_Insertar_texto_CU4.

ID-EC	Escenario	1era iteración	2da iteración	3era iteración
1	EC 1.1	No satisfactorio	Satisfactorio	Satisfactorio
2	EC 1.2	No satisfactorio	No satisfactorio	Satisfactorio

Tabla 38: SC2_Modificar_texto_CU4.

ID-EC	Escenario	Fondo	1era iteración	2da iteración	3era iteración
1	EC 2.1	V	No satisfactorio	No satisfactorio	Satisfactorio
2	EC 2.2	v	No satisfactorio	No satisfactorio	Satisfactorio

Tabla 39: SC3_Eliminar_texto_CU4.

ID-EC	Escenario	1era iteración	2da iteración	3era iteración
1	EC 3.1	No satisfactorio	No satisfactorio	Satisfactorio

Tabla 40: SC1_Insertar_imagen_CU5.

ID-EC	Escenario	1era iteración	2da iteración	3era iteración
1	EC 1.1	No satisfactorio	No satisfactorio	Satisfactorio
2	EC 1.2	No satisfactorio	No satisfactorio	Satisfactorio
3	EC 1.3	No satisfactorio	No satisfactorio	Satisfactorio



Tabla 41: SC1_Eliminar_imagen_CU6.

ID-EC	Escenario	1era iteración	2da iteración	3era iteración
1	EC 1.1	No satisfactorio	No satisfactorio	Satisfactorio

Tabla 42: SC1_Insertar_dado_CU7.

ID-EC	Escenario	1era iteración	2da iteración	3era iteración
1	EC 1.1	No satisfactorio	Satisfactorio	Satisfactorio
2	EC 1.2	No satisfactorio	Satisfactorio	Satisfactorio

Anexo #6. Validación de la herramienta de autor.

Para valorar la aplicabilidad de la herramienta de autor DEGames Board se realizará una consulta a profesores del Departamento de Programación de la Facultad 1 utilizando el método Delphi. A continuación se describen los pasos que se ejecutarían durante la aplicación del método:

1. Fase preliminar: determinación del grupo de expertos en el tema de investigación. Se identificaron X expertos del Departamento de Programación de la Facultad 1, de ellos X son profesores de Y. En esta fase se presentará la herramienta de autor a valorar al grupo de expertos y se someterá a discusión el material de validación diseñado. De los resultados que se obtengan se nutre el segundo paso.

Tabla 43: Caracterización de los expertos.

Experto	Puesto de trabajo actual	Grado científico	Categoría docente	Años de experiencia
1	UCI (DP_F1)	Máster	Asistente	5 o más
2	UCI (DP_F1)	Licenciados	Asistente	3 o 4
3	UCI (DP_F1)	Ingenieros	Asistente	3 o 4
4	UCI (DP_F1)	Máster	Asistente	5 o más
5	UCI (DP_F1)	Máster	Asistente	5 o más
6	UCI (DP_F1)	Ingenieros	Asistente	3 o 4
7	UCI (DP_F1)	Licenciados	Asistente	2o 4



2. Fase exploratoria: es donde se realizará el segundo paso de la valoración, se le presenta al grupo de expertos la herramienta de autor con los principales señalamientos corregidos y se aplica el material. Se procesan los datos obtenidos y se comunican a los expertos para comprobar la aprobación de los mismos.

Instrumento de valoración para el método Delphi.

El presente cuestionario forma parte de una consulta sobre la base de la aplicación del Método de Expertos Delphi para la validación de la herramienta de autor en desarrollo. Con este fin se solicita afectuosamente su colaboración, teniendo en cuenta que sus opiniones serán de gran valor en este trabajo de investigación para validar o rectificar la propuesta, garantizando en todo el proceso la confidencialidad de sus respuestas, utilizando los datos únicamente con intenciones académico-científicas.

1) Datos personales

Nombre y Apellidos: Centro de trabajo: Grado científico: Categoría Docente: Nivel de enseñanza: Especialidad: Años de experiencia:

2) Marque con una cruz (x), en la casilla que corresponda al grado de conocimientos que usted posee acerca del tema de investigación que se desarrolla, valorándolo en una escala de 0 a 10 (considerando 0 como no tener absolutamente ningún conocimiento y 10 el de pleno conocimiento de la problemática tratada).

0	1	2	3	4	5	6	7	8	9	10

3) Le pedimos responda el siguiente cuestionario para obtener su valoración como experto en el tema de investigación. Utilice los siguientes números para evaluar los diferentes aspectos que se proponen.

2	3	4	5
Deficiente/Desacuerdo/ Poco adecuado	Regular/Término medio	Bueno/Acuerdo/ Adecuado	Excelente/ Total acuerdo/ Muy adecuado

Sobre la estructura de la herramienta educativa:



a. ¿Considera sencillas y amigables las interfaces de la herramienta de autor para el diseño de Tableros?

5___ 4___ 3___ 2___

Sobre la aplicación de la herramienta:

b. ¿Considera que será útil como apoyo al PEA en su nivel de enseñanza?

5___ 4___ 3___ 2___

Observaciones: Algo que desee agregar sobre la estructura de la herramienta educativa.

3. Fase final: se elaborará y distribuirá el informe de resultados obtenidos acerca de la valoración de la aplicabilidad futura de la herramienta.