



Facultad 6

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



Título: Implementación de modo multijugador al módulo JDProg perteneciente a la herramienta educativa SMProg (Software – Motivación - Programación)

Autora: Eilín Hernández Rivera

Tutores: Ing. José Carlos Santiesteban Rojas

La Habana, junio de 2015



Declaración de autoría

Declaro ser la única autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Diplomante

Eilín Hernández Rivera

Tutor

Ing. José Carlos Santiesteban Rojas



Datos de contacto

Autor:

- ✓ Eilin Hernández Rivera
- ✓ Universidad de las Ciencias Informáticas.
- ✓ e-mail: eilindr@uci.cu

Tutor:

- ✓ Ing. José Carlos Santiesteban Rojas.
- ✓ Universidad de las Ciencias Informáticas.
- ✓ e-mail: jcrojas@uci.cu

Síntesis del tutor:

Graduado de Ingeniero en Ciencias Informáticas en el año 2008. Se ha desempeñado como profesor de la disciplina de programación y desarrollador de software en la Universidad de las Ciencias Informáticas.

Dedicatoria

Dedico este trabajo a mis padres, porque más que darme la vida me han enseñado a vivirla, con sus consejos, su amor y ejemplo.

A mi hermana por ser mi mayor inspiración y la persona que más quiero.

A mi compañero de vida por ayudarme a sacar fuerzas y salir adelante cada vez que sufría un revés.

A mima, mi segunda mamá y una de las personas más especiales para mí.

A mis abuelos queridos Miguel y Noemí.

Para todos ellos este trabajo con todo mi amor.

Eilín



Agradecimientos

A mis padres, quiero agradecerles por todos los valores que me han inculcado en la vida, que han hecho de mí una mejor persona, estaré siempre agradecida.

A mi querida hermana Gréttel por apoyarme en momentos difíciles y ser mi ejemplo a seguir.

A Igna, mi novio, por ser lo mejor que me ha sucedido y la persona más especial que he conocido en estos años, por todo su amor, su preocupación, su paciencia ante mis cambios de ánimo y por soportarme, por eso y mucho más...gracias.

A mi tía que a pesar de la distancia no deja de enseñarme las cosas de la vida con un punto de vista diferente, por apoyarme con su cariño incondicional y sus consejos, y por preocuparse por mí en cada momento.

A Betty por brindarme su apoyo y consejos en momentos difíciles.

A Fide un amigo que me ayudó con la programación en el momento que más lo necesité.

A todos los profesores del CPT que me brindaron su apoyo incondicional en el transcurso de la carrera.

A mi grupo del CPT: Aylemis, Duleyvis, Harold, Michel, Eileen, Nosdaly, Portillo, Fernando, Oscar, Dalinda, Amado, Marydunia, Samuel, Luisito, Kariheilyny Alejandro.

A los miembros del tribunal y al oponente.

A mi tutor Jose Carlos que no se imagina el avance que significó para mí su ayuda.

A mis compañeros del departamento: Lilita mi jefa, Yoha, Sayu, Ofe, Yusle y Kessel que siempre estuvieron al tanto de la investigación y dándome ánimos.

A todas aquellas personas que me ayudaron en la realización de este trabajo.

Gracias.



Resumen

El vertiginoso desarrollo de la humanidad ha requerido que cada vez se investigue más sobre nuevas vías para apropiarnos del conocimiento y se apliquen nuevos métodos de enseñanza. En este ámbito el término “juegos serios” es cada vez más utilizada. Se refiere, de forma informal, al uso de juegos para transmitir valores, formar en conocimientos o actitudes, o simplemente obtener un resultado diferente al propio entretenimiento. Estos juegos complejos e interactivos, se convierten en un importante ámbito de aprendizaje, entrenamiento y desarrollo de habilidades especialmente significativas para el mundo laboral. Nuestro país no se mantiene al margen en este tema. Como prueba de ello en la Universidad de las Ciencias Informáticas (UCI) se ha desarrollado una aplicación con las características de estos juegos serios llamada SMProg, que contiene un módulo para juegos didácticos con tableros llamado JDProg. El presente trabajo aborda la implementación del modo multijugador al módulo JDProg de la herramienta educativa SMProg, permitiendo así que dos o más jugadores puedan interactuar en una misma partida. Para guiar el proceso de desarrollo se utiliza la metodología RUP. Además se selecciona la arquitectura de forma que sea compatible con la herramienta educativa SMProg. Mediante el proceso de prueba se detectaron y corrigieron las no conformidades obteniendo como resultado una herramienta útil para el apoyo al Proceso de Enseñanza Aprendizaje.

Palabras clave: juegos serios, didácticos, implementación, multijugador.



Abstract

The overwhelming development of mankind has required to research about new ways of acquiring knowledge and new teaching methods. In this context, the term “serious games” is more often used. In an informal way, it refers to the use of games for transmitting human values, developing knowledge or attitudes or simply obtaining another result apart from just entertainment. These complex and interactive games become an important environment for learning, entertaining and developing those special outstanding skills for the working world. Our country is well aware of this, therefore an application, SM Prog, with the characteristics of these serious games has been developed at the University of Informatic Sciences. The application includes a module for didactic games with a board called JDProg. This paper deals with the implementation of the multiplayer mode to the JDProg of the educational tool SMProg. This allows two or more players to interact in the same match. RUP methodology is used to lead the development process. The architecture is chosen so that it is compatible with the SMProg educational tool. By means of the testing process the non-conformances were detected and corrected resulting in a useful tool to support the teaching -learning process.

Key words: serious games, didactic, implementation, multiplayer



Índice de contenido

<i>Introducción</i>	1
<i>Capítulo 1: Fundamentación Teórica y Selección de las Herramientas</i>	4
1.1 Introducción	4
1.2 El proceso de desarrollo de juegos didácticos en aplicaciones web	4
1.3 Desarrollo de juegos didácticos en aplicaciones web con la modalidad multijugador.....	4
1.4 Los juegos educativos haciendo uso de multijugadores, su aplicación en el mundo y Cuba	5
1.5 Marco Teórico-Conceptual de la investigación	7
1.6 Antecedentes de la investigación	7
1.7 Tendencias actuales a considerar en el desarrollo del sistema.....	8
1.8 Metodología, lenguaje y herramienta de modelado	8
1.9 Herramientas, tecnologías y lenguajes para el desarrollo	9
1.10 Fundamentación de las herramientas, tecnologías y lenguajes a utilizar.....	14
1.11 Conclusiones parciales	15
<i>Capítulo 2: Análisis y diseño de la propuesta de solución</i>	17
2.1 Introducción	17
2.2 Propuesta del sistema para la ejercitación de contenidos	17
2.3 Modelo de dominio	17
2.4 Levantamiento de requisitos	19
2.5 Modelado del sistema	21
2.6 Diseño del sistema.....	27
2.7 Descripción de la arquitectura	30
2.8 Diagramas de clases del diseño.....	32
2.9 Diagramas de secuencia	34
2.10 Modelo de datos.....	34
2.11 Conclusiones parciales	35
<i>Capítulo 3: Implementación y validación de la solución</i>	36
3.1 Introducción	36
3.2 Estándar de codificación.....	36
3.3 Diagrama de componentes	37
3.4 Diagrama de despliegue.....	38



3.5 Pruebas de software.....	40
3.6 Resultados de las pruebas.....	46
3.7 Conclusiones parciales.....	46
<i>Conclusiones generales</i>	47
<i>Recomendaciones</i>	48
<i>Referencias bibliográficas</i>	49
<i>Bibliografía consultada</i>	52
<i>Glosario de términos</i>	55
<i>Anexos</i>	58
Anexo #1 Entrevista realizada a experto.....	58
Anexo #2 Tipos de juegos didácticos del sistema.....	58
Anexo #3 Descripción textual de los casos de uso del sistema: C.U. Autenticar.....	59
Anexo #4 Descripción textual de los casos de uso del sistema: C.U. Cumplir reglas juego multijugador.....	61
Anexo #5 Diagramas de secuencia.....	63
Anexo #6 Diagramas de secuencia.....	64
Anexo #7 Caso de prueba CU Autenticar usuario.....	64
Anexo #8 Caso de prueba para el CU Cumplir reglas multijugador.....	65



Índice de figuras

Figura 1. Modelo de dominio del sistema JDProg	18
Figura 2. Diagrama de casos de uso del sistema.....	22
Figura 3 Arquitectura cliente/servidor en 3 capas	27
Figura 4 Arquitectura Peer to Peer	28
Figura 5. Spring MVC.....	31
Figura 6. Diagrama de clases: CU Autenticar	32
Figura 7. Diagrama de clases: Jugar multijugador	33
Figura 8. Diagrama de clases: CU Cumplir reglas multijugador	33
Figura 9. Diagrama de secuencia: CU Jugar multijugador sección Unirse a un juego creado	34
Figura 10. Modelo entidad-relación	35
Figura 11 Estándar de codificación <i>camelCase</i> para el nombre de las clases.	36
Figura 12. Estándar de codificación <i>camelCase</i> para métodos y variables.....	37
Figura 13 Diagrama de componentes	38
Figura 14. Diagrama de despliegue.....	39
Figura 15 Gráfico de No conformidades por iteraciones	45
Figura 16 Resultado de las pruebas.....	46
Figura 17. Diagrama de secuencia caso de uso Autenticar	63
Figura 18 Diagrama de secuencia CU Cumplir reglas multijugador.....	64



Índice de tablas

Tabla 1 Usuario del sistema y su descripción.....	21
Tabla 2. Caso de uso Jugar multijugador	22
Tabla 3 Caso de prueba CU Jugar multijugador Sección Crear una nueva partida	41
Tabla 4 Caso de prueba CU Jugar multijugador Sección Unirse a una partida creada.....	43
Tabla 5. Caso de uso Autenticar	59
Tabla 6. Caso de uso Cumplir reglas juego multijugador.....	61
Tabla 7 Caso de prueba CU Autenticar usuario	64
Tabla 8 Caso de prueba para el CU Jugar Multijugador	65



Introducción

El desarrollo de las Tecnologías de la Información y las Comunicaciones (TICs) en el mundo ha influido en los sectores de la sociedad dando como resultado una mejora en los procesos de gestión de la información (Rosario 2010). Uno de los ámbitos que utiliza las bondades de las TICs es el de los juegos, diseñados con un ambiente motivador y envolvente gracias a los desafíos adecuados al nivel de competencia.

En la actualidad, existe una variedad de juegos orientados a la educación que incluyen juegos tradicionales, de competencia, de roles, de actuación, de recreación, educativos, interactivos en computadora, didácticos, aplicados y orientados a la enseñanza y aprendizaje de materias. Estos inician a partir de las investigaciones realizadas por K. Groos al definir el juego como un adiestramiento para futuras capacidades serias (Gross 2008).

El desarrollo de estas tecnologías interviene actualmente de forma directa en la educación. En consecuencia centros educativos de varios países del mundo en aras de contribuir a mejorar la integración de los estudiantes en el Proceso de Enseñanza-Aprendizaje (PEA), se han incorporado de manera creciente a la creación y utilización, desde herramientas hasta software educativos para apoyar la docencia. Un ejemplo particular lo constituyen las aplicaciones con juegos didácticos, incluidos entre los llamados juegos serios¹ (Ferreiro 2011).

Estos juegos centran su diseño, desarrollo, aplicación y uso para otros fines que no son solo el entretenimiento; sino también para desarrollar una nueva forma de enseñanza y aprendizaje mediante la interconexión masiva y sincronizada de personas en juegos con la modalidad multijugador. Esta modalidad permite que más de una persona pueda jugar en el mismo entorno de juego al mismo tiempo, proporcionando una comunicación que está ausente en un solo jugador. Además el jugador ya no se enfrentaría a desafíos pre-programados y controlados por la inteligencia artificial, sino con la flexibilidad y el ingenio del pensamiento humano regular.

A pesar de todo lo atractivo que puede ser este novedoso medio de enseñanza, hasta este momento en Cuba el uso de los juegos serios es limitado, debido a que no hay una estrategia que reúna los resultados y procesos de diseños sistemáticos a seguir, pues en la bibliografía científica se describe más bien un conjunto de recomendaciones generales.

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 Proceso de Enseñanza-Aprendizaje.

Entre sus centros educacionales de nivel superior se encuentra la Universidad de las Ciencias Informáticas (UCI), universidad encargada de producir software, garantizando su desarrollo y el de la informática en el país. Como apoyo a esta búsqueda, en el marco del Proyecto de

¹Objetos y/o herramientas de aprendizaje que poseen en sí mismos y en su uso objetivos pedagógicos, didácticos, autónomos, autosuficientes y reutilizables, que posibilitan a los jugadores a obtener un conjunto de conocimientos y competencias predominantemente prácticos.



Innovación-Investigación para las asignaturas de Técnicas de Programación, se desarrolló en la UCI una aplicación con las características de estos juegos serios, lo cual facilita la creación de juegos didácticos para la ejercitación de contenidos.

Esta herramienta educativa de nombre SMProg (Software – Motivación - Programación) es desarrollada por un equipo de trabajo compuesto por profesores y estudiantes de la facultad 1 de la UCI, para integrar conocimiento y recreación de manera tal que el estudiante lejos de sentir la presión de aprender la asignatura como una carga, lo asumiera como una forma creativa y entretenida potenciando de esta manera el aprendizaje. 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 se le señalan algunos inconvenientes entre los cuales se encuentra que no permite el acceso concurrente de profesores y estudiantes.

Debido a estas desventajas en el año 2014, como trabajo de diploma, la Ingeniera Luisa Yanara Cuenca desarrolló el módulo de nombre JDProg que permite la ejercitación y evaluación de contenidos con la utilización de juegos didácticos estilo tablero en la web, cuyo objetivo está dirigido a despertar el interés de manera creativa e interactiva de los estudiantes por el estudio de la disciplina Técnicas de Programación de Computadoras (TPC). Aún con la adición de este nuevo módulo persiste el obstáculo de no permitir el acceso de dos o más jugadores en el mismo entorno de juego al mismo tiempo, de manera que puedan competir entre sí.

Según lo planteado anteriormente se determina como **problema a resolver**: ¿Cómo garantizar que la herramienta educativa SMProg permita la interacción de varios jugadores en una misma partida?

El problema anterior conlleva a definir como **objeto de estudio**: el proceso de desarrollo de juegos didácticos en aplicaciones web delimitando el **campo de acción** al desarrollo de juegos didácticos en aplicaciones web con la modalidad multijugador.

Este trabajo tiene como **objetivo general**: Implementar el modo multijugador al módulo JDProg perteneciente a la herramienta educativa SMProg para posibilitar la interacción de varios jugadores en una misma partida.

Se plantea como **idea a defender** que la implementación del modo multijugador al módulo JDProg perteneciente a la herramienta educativa SMProg permitirá la interacción de varios jugadores en una misma partida.

Para alcanzar el objetivo propuesto se definieron las siguientes **tareas investigativas**:

- ✓ Determinación de los fundamentos teóricos acerca del proceso de desarrollo de juegos didácticos en aplicaciones web y desarrollo de juegos didácticos con la modalidad multijugador.
- ✓ Determinación de las herramientas y tecnologías a utilizar en el desarrollo del modo multijugador.
- ✓ Implementación del modo multijugador al módulo JDProg perteneciente a la herramienta educativa SMProg que permita la interacción de varios jugadores en una misma partida.
- ✓ Diseño y aplicación de las pruebas al modo multijugador del módulo JDProg de la herramienta educativa SMProg.

Para cumplir con las tareas de investigación se utilizaron los siguientes **métodos investigativos**:

**Métodos teóricos:**

- ✓ Histórico - Lógico: para determinar las tendencias actuales del desarrollo de los juegos didácticos con la modalidad multijugador.
- ✓ Analítico–Sintético: la utilización de este método permitió determinar las funcionalidades que conlleva en si la implementación del modo multijugador y analizar los artefactos y la documentación obtenida en el desarrollo del módulo JDProg.

Métodos empíricos:

- ✓ Entrevista: Este método se utilizó para obtener la información referente al sistema JDProg asociado a las funcionalidades que son necesario implementar. El tipo de entrevista utilizada es la no estructurada. Se aplica a especialistas en el tema, es una forma de obtener criterios de expertos **(Ver Anexo #1)**.

El presente trabajo de diploma cuenta con introducción, 3 capítulos, conclusiones, recomendaciones, bibliografía, glosario de términos y anexos.

Capítulo 1: “Fundamentación Teórica y selección de las herramientas”. En este capítulo se determinan los fundamentos teóricos acerca del proceso de desarrollo de juegos didácticos en aplicaciones web y desarrollo de juegos didácticos con la modalidad multijugador. Además se seleccionan las herramientas y tecnologías para el desarrollo.

Capítulo 2: “Análisis y Diseño de la propuesta de solución”: En este capítulo se identifican los requisitos funcionales y no funcionales del sistema, se diseñan los diagramas de clases, se analizan las funcionalidades a implementar, especificando las pautas y patrones de diseño utilizados y se plantea la propuesta de solución.

Capítulo 3: “Implementación y Validación de la solución”. En este capítulo se elabora el modelo de despliegue el cual representa la composición física en términos de módulos y elementos de implementación, así como la realización de los diagramas de componentes. Se confeccionan las pruebas a realizar en el sistema y se exponen los resultados de las mismas utilizadas para certificar la solución propuesta.



Capítulo 1: *Fundamentación Teórica y Selección de las Herramientas*

1.1 **Introducción**

En el presente capítulo se describen los elementos fundamentales que constituyen el soporte teórico de la investigación y se realiza un análisis de las características importantes relacionados con la homologación de los sistemas informáticos ya existentes tanto en el ámbito internacional como nacional. Se valoran las tendencias actuales en el contexto informático y se identifican los elementos claves 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 **El proceso de desarrollo de juegos didácticos en aplicaciones web**

A partir de los estudios efectuados por filósofos, psicólogos y pedagogos, han surgido diferentes teorías que han tratado de dar diversas definiciones acerca del juego. Existen diferentes tipos de juegos: juegos de reglas, juegos constructivos, juegos de dramatización, juegos de creación, juegos de roles, juegos de simulación, y juegos didácticos (Gross 2008). Los juegos didácticos se convierten en una actividad planificada y orientadora que satisface ciertas exigencias académicas que presentan elementos de motivación, competencia, espontaneidad, participación y emulación que ayudan, sin lugar a dudas, a resolver tareas de corte educativo (EcuRed 2010), estos entran en el grupo de los “juegos serios”, para distinguirlos de los puramente orientados al entretenimiento (David 2005).

La educación ha sido una de las disciplinas más beneficiadas con la introducción de las nuevas tecnologías, especialmente las relacionadas a la web (Guerra 2013). Existe una importante línea de desarrollo de software para la web en las que se incluyen las llamadas aplicaciones web, en esencia aplicaciones tradicionales de un PC pero implementadas con la tecnología web y que únicamente requieren un navegador estándar para su utilización (Santos 2009).

La integración de los juegos didácticos con las aplicaciones web permitió que las opciones pedagógicas llegaran a un público masivo, con estructuras que se alejan de los clásicos procedimientos educativos y permiten una distribución rápida y de acceso sencillo, sin que represente para el usuario una carga, sino un aprendizaje (Arcila 2011).

1.3 **Desarrollo de juegos didácticos en aplicaciones web con la modalidad multijugador**

Un juego multijugador es un videojuego en donde pueden participar, e interactuar en un mundo virtual, un gran número de jugadores simultáneamente (multijugador) conectados a través de la red (en línea), normalmente Internet dado el grado de concurrencia que pueden llegar a alcanzar y las características técnicas de los servidores que han de gestionar ese volumen de conexiones. (Domínguez 2012).

Como en otras aplicaciones de las TICs, el ámbito educativo recoge los frutos de los juegos multijugador, no solo como proyectos infantiles sino como programas interactivos dedicados a la enseñanza (Arcila 2011).



Después de revisar la bibliografía escogida, se pudo determinar que independientemente del tipo de juego, la modalidad multijugador cuenta con una serie de funcionalidades en común que caracteriza este modo de juego. Por ejemplo al escoger el modo multijugador, este debe permitir que el usuario pueda unirse a una partida creada, invitar a un jugador a unirse a una partida, ver lista de jugadores conectados y lista de partidas creadas, brindar información visual en tiempo real a todos los usuarios conectados a una partida sobre movimiento y estado de los otros jugadores sin que se le solicite al servidor y tener algún tipo de comunicación entre los jugadores que puede ser mediante un chat.

1.4 Los juegos educativos haciendo uso de multijugadores, su aplicación en el mundo y Cuba

El uso didáctico de los juegos multijugador *online*, conjugado con la subliminalidad en los procesos educacionales, se convierte en un elemento influyente en la construcción holográfica²del concepto aprendido.

Los videojuegos permiten aumentar la motivación para el aprendizaje de diversas materias como las matemáticas y las ciencias. Además pueden ser utilizados como entrenamiento eficaz en programas de tipo viso-motor, desarrollo del pensamiento reflexivo, mejora de las habilidades de los pilotos de avión, reducir el número de errores de razonamiento, conseguir un mayor control de los tiempos de reacción, y servir de enfrentamiento ante situaciones vitales que pueden ser simuladas, como es el caso de la resolución de problemas, tema en el que se muestran muy eficaces(Arcila 2011).

1.4.1 Ámbito internacional

Actualmente, existen varias decenas de juegos didácticos dedicados a propósitos educativos y su número seguirá incrementándose debido a que se hace muy atractivo por la interconexión masiva y el gran impacto comunicacional. Durante la investigación se analizaron algunos ejemplos de juegos con la modalidad multijugador, de los cuales a continuación se realiza una breve descripción por ser los que más aportaron a la investigación en cuanto a: el enfoque a la educación, las funcionalidades y además porque fueron de los que se encontraron mejor documentación debido a que son juegos implementados por instituciones de investigación, universidades e institutos de tecnología que se han dedicado también a la utilización de las nuevas tecnologías como recurso educativo.

TRINX: desarrollo y evaluación de juegos educativos multijugador: es un video juego educativo desarrollado por el grupo HCI-RG (*Human Communication and Interaction Research Group*), del departamento de informática de la Universidad de Oviedo, España. Es un juego multijugador de aventuras, por lo que varios jugadores podrán enfrentarse a la aventura juntos, potenciando las posibilidades educativas. Cuenta con una evaluación por cada estudiante a medida que van interactuando en tiempo real. Está estructurada en componentes desarrollados en una arquitectura en 3 capas con la idea de que sea posible adaptar el entorno al centro que lo utilice configurando algunos de los componentes y recursos existentes(Gayo 2011).

² Imagen completa, tridimensional.



The radix endeavor: es un juego online multijugador de STEM (ciencia, tecnología, ingeniería y matemáticas) de aprendizaje en la escuela secundaria, realizado por el laboratorio de investigación y desarrollo ubicado en el Instituto de Tecnología de Massachusetts (MIT por sus siglas en inglés), que explora lo que es auténticamente lúdico en el aprendizaje. En la fase inicial se tratarán temas de la biología, álgebra, geometría, probabilidad y estadística, proporcionando a los estudiantes una experiencia de colaboración social en un mundo de juego basado en sistemas donde pueden explorar cómo funciona el medio y descubren importantes conceptos científicos. Las evaluaciones están basadas en las tareas innovadoras por cada área temática, donde el juego utilizará un registro de datos para comparar y analizar estos en tiempo real, ofreciéndole al jugador una retroalimentación si la idea es errónea (Arcade 2014).

Extensión a multijugador de ViRPlay 3D, un juego educativo de enseñanza de orientación a objetos usando role-play³: desarrollado por el Departamento de Ingeniería del Software e Inteligencia Artificial de la Facultad de Informática de la Universidad Complutense de Madrid, este proyecto ha consistido en la extensión de una herramienta educativa monousuario para la enseñanza de la programación orientada a objetos usando la técnica del role-play. Su objetivo principal es transmitir mediante un juego, como herramienta educativa, el paradigma de la programación orientada a objeto (García 2008), a partir de la colaboración entre sí de varios jugadores en un mismo escenario, motivando a todos a participar activamente en el desarrollo del mismo.

1.4.2 Ámbito nacional

Es importante destacar que en Cuba se lleva a cabo un arduo trabajo para aprovechar las posibilidades de las nuevas tecnologías, como recurso importante, para apoyar el Proceso de Enseñanza-Aprendizaje. Algunos ejemplos son:

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 aquellos a quienes están dirigidos, 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 (Hechavarría 2010).

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, que siguen el camino de la web pero es **Infodrez-Módulo de torneo online**, un módulo desarrollado para la plataforma web de Infodrez el que permite específicamente jugar torneos de ajedrez en tiempo real. Entre sus funcionalidades principales se encuentran crear torneos, invitar árbitros y jugadores al torneo, jugar las partidas en cada ronda y publicar los resultados de las rondas y el resultado final. Fue realizado en lenguaje PHP con la herramienta Zend Studio considerada por muchos como herramienta líder en la programación web con PHP (Garrido 2010).

³ Es la interpretación de diferentes roles o papeles que interactúan entre sí dentro de un entorno y situación concretos.



UCIScrabble: Módulo partida entre 2 jugadores: este trabajo debe estar vinculado en una plataforma web que permita que dos jugadores puedan jugar Scrabble⁴ entre ellos en red y medir su fuerza con relación a su vocabulario. Desarrollado también en lenguaje PHP y con la herramienta Zend Studio permite elevar el nivel ortográfico de la comunidad universitaria, midiendo su capacidad referente al dominio de la lengua materna; presentando además una interfaz sencilla, amigable, fácil de entender y usar, que posibilita una rápida navegación y desarrollo del juego (Velázquez 2009).

1.5 Marco Teórico-Conceptual de la investigación

Después de realizar un estudio de algunos juegos con propósitos educativos, utilizados en el mundo, el país y la universidad, se pudo apreciar, que a pesar de las amplias posibilidades que brindan, en cuanto al desarrollo de recursos educativos, en su generalidad fueron diseñados según las particularidades del nivel de enseñanza donde son aplicados.

En el caso de los sistemas a nivel internacional suelen ser diversos los que incluyen los juegos didácticos como alternativa motivadora e interactiva en su solución. Durante la presente investigación se analizaron algunos ejemplos, que aunque tienen temáticas totalmente distintas, comparten en común que son multijugador y *online*. Además solo se necesita de un navegador web para jugarlas.

Este análisis fue necesario para tener en cuenta los aspectos técnicos de los juegos ya existentes, lo que influyó en la decisión de las funcionalidades a incorporar en el módulo JDProg de la herramienta educativa SMProg, que permita el modo multijugador y en la arquitectura de red cliente/servidor, que como se ha visto anteriormente es la más utilizada, además de la posibilidad de aprovechar las nuevas tecnologías.

Por otra parte Infodrez-Módulo de torneo online y UCIScrabble: Módulo partida entre 2 jugadores que aunque está desarrollado en la web y permite también la interacción entre los jugadores, no se adecua a las necesidades descritas anteriormente, pues aunque está muy bien diseñado, para utilizarlas en el nivel universitario enfocándolo directamente a las materias que se imparten se requiere de un alcance mayor.

1.6 Antecedentes de la investigación

En la Universidad de las Ciencias Informáticas (UCI) se desarrolló en el 2011 la herramienta educativa SMProg que le permite al profesor definir el sistema de preguntas que desea se muestren a los estudiantes que se encuentren “jugando”. Las preguntas pueden agruparse por temas y se muestran en un inicio de forma aleatoria de acuerdo al tema que escoja el estudiante, el propio software mostrará las preguntas de ese tipo y realizará la evaluación del mismo.

Esta herramienta también permite al profesor realizar el diseño del tablero 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 desarrollaron tres módulos diferentes estrechamente vinculados entre sí y con similar ambiente, estos son:

⁴ Consiste en formar palabras sobre un tablero de quince columnas e igual número de filas, de manera vertical y horizontal



Módulo Juegos didáctico con Tableros (JDProg lógica de negocio): desde este módulo se integran los restantes, es por esto que se nombra módulo principal. Desde el mismo es posible seleccionar la asignatura, los tableros de juegos e iniciar el juego, con sus características propias, manteniendo la comunicación con la base de datos.

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 (**Ver Anexo #2**). 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.

Gestión de Contenidos (GesCont): herramienta que posibilita en gran medida la interacción con la base de datos, y su objetivo principal es la definición de temas y sistemas de preguntas de las asignaturas que serán utilizados en la solución.

1.7 Tendencias actuales a considerar en el desarrollo del sistema

La presente investigación se propone como alcance, garantizar, a través de la implementación del modo multijugador en el módulo JDProg de la herramienta informática educativa SMPProg, la ejercitación de los contenidos o conocimientos a través de un juego didáctico estilo tablero en la web.

Para la implementación del modo multijugador se realizó un estudio sobre las principales herramientas a utilizar, así como las metodologías a ser empleadas durante su elaboración. Analizando las tendencias actuales y las novedades existentes en el campo del desarrollo y sobre todo teniendo en cuenta el módulo que se quiere modificar, se llega a las siguientes propuestas sobre las metodologías, lenguajes de programación y otras herramientas que serán empleadas durante su desarrollo.

1.8 Metodología, lenguaje y herramienta de modelado

1.8.1 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 (Visual 2012).

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. No presenta pasos que estén firmemente establecidos sino que cuenta con un conjunto de flujos de trabajo adaptables al contexto y necesidades de cada organización.

1.8.2 Herramienta CASE y lenguaje de modelado



Las herramientas CASE (*Computer Aided Software Engineering*, por sus siglas en inglés, o Ingeniería de Software Asistida por Ordenador) son aplicaciones informáticas que utiliza el Lenguaje Unificado de Modelado (UML) como notación para elaborar los modelos de la manera más eficiente posible y comunicarlos a todos los agentes del proyecto, pues son determinantes para cumplir con los objetivos de este.

Visual Paradigm versión 8.0

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 (Visual 2012).

UML versión 2.1

El Lenguaje Unificado de Modelado más conocido por todos como UML (*Unified Modeling Language* por sus siglas en inglés), es un lenguaje utilizado para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software (Jacobson 2000). 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 que son fundamentales para el correcto uso del mismo (Larman 2003), los cuáles son: de estructura estática (clases, objetos y casos de uso), de comportamiento (interacción, estado, actividad) y de implementación (componentes, despliegue).

1.8.3 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. 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.

1.9 Herramientas, tecnologías y lenguajes para el desarrollo

1.9.1 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 (Gutiérrez 2005).

Spring versión 3.0.6



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 *Java Enterprise Edition* (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(Johnson 2004).

Node.js versión 0.10.24

Es un entorno de programación en la capa del servidor basado en el lenguaje de programación Javascript, con I/O de datos en una arquitectura orientada a eventos, y basado en el motor JavaScript V8. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, servidores web.

Node puede ser utilizado para desarrollar aplicaciones que manejen altos volúmenes de conexiones con otros sistemas. En este tipo de aplicaciones los tiempos de respuesta bajos marcan una delgada línea entre el éxito o fracaso. Es especialmente bueno en aplicaciones web que necesiten una conexión persistente con el navegador del cliente, mediante una aplicación que envíe datos al usuario en tiempo real; es decir, que el navegador mantenga la conexión siempre abierta y reciba continuamente nuevos datos cuando los haya. Para servidores y aplicaciones clásicas que no estén preparadas para mantener muchas conexiones, la forma más sencilla es solicitar desde el navegador cada X segundos nueva información. Si hay muchos usuarios a la vez abriendo conexiones y haciendo peticiones a la BD se encontrarán con que el servidor no da más de sí y deja de atender peticiones(Herron 2011).

Para ejecutar el código JavaScript en la parte del servidor, este necesita ser interpretado y ejecutado, Esto es lo que Node.js realiza, haciendo uso de la máquina virtual V8 de Google, el mismo entorno de ejecución para JavaScript que Google Chrome utiliza.

Node cuenta con varios módulos útiles lo que lo convierte en dos cosas: un entorno de ejecución y una librería.

Socket.IO es una librería que nos permite manejar eventos en tiempo real mediante una conexión TCP y todo ello en JavaScript. Es realmente potente y podemos hacer todo tipo de aplicaciones en tiempo real(Teixeira 2013).

Una lista de aplicaciones para las que Node encajaría perfectamente:

- ✓ Juegos *online*.
- ✓ Gestores de correo *online*: de esta manera teniendo el navegador abierto podríamos ver notificaciones en tiempo real de nuevos correos recibidos.



- ✓ Herramientas de colaboración.
- ✓ Chats.
- ✓ Redes sociales: por ejemplo para actualizar automáticamente tu muro de novedades.
- ✓ Herramientas de traducción en tiempo real.

1.9.2 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 objeto y de una plataforma independiente. Los programas en Java generalmente son compilados y luego interpretados por una máquina virtual. Esta última sirve 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 versión 1.5

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)(Arana 2012).

1.9.3 Lenguaje de Marcado Hipertextual (HTML) versión 5.0

Según Chuck el lenguaje de marcado hipertextual (Hypertext Markup Language por sus siglas en inglés) es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define las sintaxis y coloca instrucciones especiales que no muestra el navegador(JOOMLA-CHILE 2012). En este sistema se utiliza para indicar como desplegar el contenido de las interfaces de usuarios, incluyendo elementos de texto, imágenes y otros medios soportados.

Hojas de estilo en cascada (CSS) versión 3.0



Las hojas de estilo en cascada (Cascading Style Sheets por sus siglas en inglés) hacen referencia a un lenguaje creado para controlar y mejorar el aspecto o presentación de un documento electrónico. En la aplicación para ejercitación de contenidos se utilizan estos estilos para mejorar el diseño de las interfaces de usuario y mostrar cada una de las vistas según las propiedades que brindan los CSS, al permitir designar la ubicación de los contenidos, incluyendo textos e imágenes (Gauchat 2013).

1.9.4 Sistema Gestor de Base de Datos

Un Sistema Gestor de Base 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 versión 9.2.4

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 (Sitio 2010).

1.9.5 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.

1.9.6 Servidor Web



Un servidor web es un programa que permite atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP (Hyper Text Transfer Protocol por sus siglas en inglés) 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 versión 7.0.34

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 SunMicrosystem(Godfrey 2003).

1.9.7 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 versión 7.4

El Entorno de Desarrollo Integrado 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, C, C++, XML y lenguajes HTML.
- ✓ Herramientas con asistentes para facilitar la escritura de código (Sánchez 2004).



1.9.8 Herramienta ORM (Object-Relational Mapping)

Una herramienta ORM constituye una técnica de programación para convertir datos entre el lenguaje de programación orientado a objetos y el sistema de base de datos relacional utilizados en el desarrollo de una aplicación informática (Carrero 2013).

Las herramientas ORM son utilizadas por lo general cuando las bases de datos relacionales solo pueden guardar datos primitivos, pues al no poder almacenar objetos creados en el transcurso del desarrollo de una aplicación, se convierten los datos del objeto en datos primitivos y de esta manera quedan almacenados en las tablas correspondientes de la base de datos. Si se necesita ese objeto en alguna parte de la aplicación, se recuperan los datos primitivos de la base de datos y se construye el objeto.

Hibernate versión 3.2.5

Para la solución de posibles problemas relacionados con las diferencias entre el modelo orientado a objetos y el usado en la base de datos, se decidió el uso de Hibernate para la solución argumentando que es un poderoso medio para mapear clases de java a tablas de la base de datos, procedimiento que facilita la configuración de ficheros XML y a la vez desacopla la base de datos de las clases de la aplicación.

Hibernate está diseñado para ser flexible si se trata del esquema de tablas utilizado, para poder adaptarse sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de un modelo de objetos existente(Ojeda 2008).

1.10 Fundamentación de las herramientas, tecnologías y lenguajes a utilizar

Entre los proyectos de Innovación – Investigación que lleva a cabo la dirección del Dpto. de Programación de la Facultad 1 se encuentra la herramienta educativa SMProg. Dicha dirección tiene como objetivo fundamental estandarizar el uso de las herramientas, tecnologías y lenguajes en los diferentes proyecto que desarrolla. Para el desarrollo del presente trabajo se utilizará 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 del 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 es multiplataforma.

2. Se puede utilizar 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 utilizarse 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 características de Java es que el sistema multijugador que se deriva de esta investigación será ejecutado.

Se propone la utilización de Spring Framework debido a que es un *framework* 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 “*ExpertOne-on-One Java EE Design and Development*”. Es el único *framework* que interviene en todas las capas arquitectónicas de una aplicación JEE, además está diseñado para facilitar una flexibilidad arquitectónica.

Una de las causas por las que se propone Node.js es porque este nuevo motor es un cambio innovador en términos de fiabilidad, hay mucho de código compartido entre el navegador y el servidor para reducir al mínimo la necesidad de mantenimiento del código y reducir el margen de error y las incompatibilidades. Es un motor para la fiabilidad y soporta la comunicación bidireccional, basada en evento para enviar datos desde el servidor al cliente sin que este lo solicite.

Como sistema gestor de base de datos se utilizará PostgreSQL, ya que es uno de los gestores de bases de datos relacional orientado a objetos de software libre más utilizado en el mundo. Es 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 Apache TomCat 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.

1.11 Conclusiones parciales

Luego de realizar la investigación de las principales funcionalidades del modo multijugador en cada uno de los sistemas descritos anteriormente en los ámbitos analizados, se pudo percibir que ninguno contiene tableros de juegos didácticos como recurso educativo con el modo multijugador. Sin embargo su estudio permitió conocer acerca de cómo estos manejan el tratamiento de la información utilizando este modo de juego y sus principales características, las cuales son que debe permitir crear partida, unirse a una partida creada, invitar a un jugador determinado a unirse a una partida, ver contrarios en el entorno de juego y realizar evaluación por jugador.



Se definieron las tecnologías y herramientas para el desarrollo del multijugador teniendo en cuenta, sobre todo, las características del módulo al que se le incorporará las funcionalidades de este modo de juego.



Capítulo 2: Análisis y diseño de la propuesta de solución

2.1 Introducción

El objetivo que se persigue con la elaboración de este capítulo es mostrar la evolución de la solución durante las fases iniciales definidas en la metodología de desarrollo de software RUP, así como presentar los diferentes artefactos generados en la misma, los cuales son importantes para el desarrollo del modo multijugador en el módulo JDProg.

2.2 Propuesta del sistema para la ejercitación de contenidos

El sistema ha sido concebido de forma tal que los usuarios puedan acceder mediante el navegador web a la aplicación contenida en el servidor, donde al interactuar con el juego didáctico se iniciará la ejercitación de los contenidos.

En caso de que el usuario desee ejercitar con la interacción de más jugadores en la misma partida, contará con la posibilidad de crear una nueva partida o unirse a una partida creada. En caso de crear una nueva partida el sistema mostrará los jugadores conectados que no están en partida a los cuales podrá invitar a unirse a su partida. Por otro lado para unirse a una partida el sistema mostrará las partidas creadas que no han sido iniciadas. El sistema asume como jugador con el 1er turno para jugar de forma aleatoria y después de que responda o no las preguntas el juego inhabilitará la ficha del jugador pasando al que está esperando turno, hasta la cantidad de jugadores permitidos en la misma partida. Al concluir el juego el sistema realiza la evaluación de los estudiantes sumando los puntos de respuestas acertadas en la partida.

2.3 Modelo de dominio

Un modelo del dominio es una representación de las clases conceptuales o entidades del mundo real. Esto ayuda a los usuarios, clientes, desarrolladores e interesados; a utilizar un lenguaje común para poder entender el contexto en que se desarrolla el sistema. Es decir, captura los tipos más importantes de objetos que existen, los eventos que suceden en el entorno donde estará el sistema y no incluyen las responsabilidades de las personas que ejecutan las actividades(Larman 2003).

Para el desarrollo de la propuesta no existe un negocio bien definido, porque pertenece a un proyecto de innovación investigación, orientado a la creación de una aplicación que permita ejercitar los contenidos utilizando juegos didácticos estilo tablero en la web. Es por esto que se realiza un modelo de dominio donde se ilustran los principales conceptos para comprender el entorno y describir los componentes del sistema.

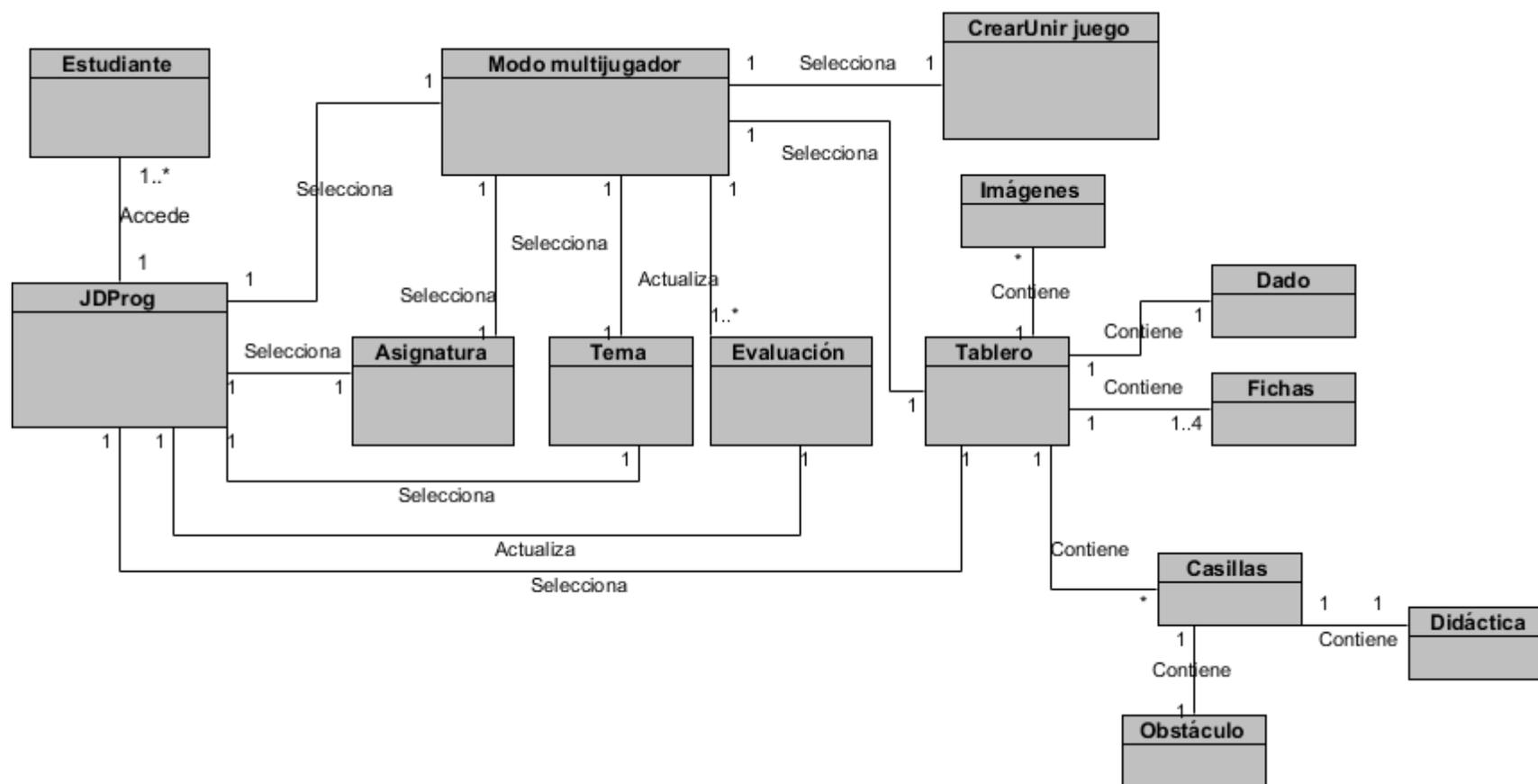


Figura 1. Modelo de dominio del sistema JDProg

En la figura 1 se muestra la relación existente entre las entidades o clases, es posible apreciar como el estudiante es el encargado de interactuar con el sistema JDProg.

Para una mejor comprensión de este modelo de dominio se especifica el glosario de términos de los elementos que lo componen:

- ✓ El **estudiante** es el actor que interactúa con el sistema.
- ✓ El elemento **JDProg** es el sistema capacitado para ejercitar y evaluar los contenidos utilizando juegos didácticos estilo tablero en la web.
- ✓ La **asignatura** hace referencia a la materia que se va a ejercitar durante el juego.
- ✓ El **tema** corresponde al contenido que se ejercitará de la asignatura seleccionada para ejercitar en el juego.
- ✓ El **tablero** es la clase que contiene la información necesaria para ser utilizada en el desarrollo del juego.
- ✓ Las **fichas** es el elemento con el que el jugador se identifica al recorrer el tablero.
- ✓ Las **imágenes** son elementos que componen el tablero para mejorar el entorno.
- ✓ El **dado** es el elemento que compone el tablero para decidir de qué manera se mueven las fichas dentro del tablero.
- ✓ Las **casillas** son los elementos que componen la estructura del tablero, conectadas entre sí según la manera en la que se diseña.
- ✓ Las **casillas didácticas** son los elementos del tablero que contendrán el componente educativo del juego.
- ✓ Los **obstáculos** son los elementos que componen el tablero para darle la complejidad al juego.



- ✓ La **evaluación** es el elemento que determina el resultado obtenido al finalizar el juego.
- ✓ El **modo multijugador** permitirá al estudiante interactuar con otro jugador en la misma partida.
- ✓ **CrearUnir juego** corresponde al modo multijugador, permite crear una nueva partida o unirse a una partida existente.

2.4 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.

2.4.1 Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Los requisitos funcionales identificados para el desarrollo del modo multijugador son:

- RF1** Autenticar Usuario: esta funcionalidad permite verificar usuario y contraseña de quienes desean acceder al sistema.
- RF2** Establecer Nick: esta funcionalidad permite que el jugador establezca el Nick que desee, después de escoger el modo multijugador.
- RF3** Mostrar listado de usuarios conectados: esta funcionalidad permite que se muestren los usuarios conectados que no se encuentren en partida.
- RF4** Enviar invitación a un usuario determinado: esta funcionalidad permite que un jugador seleccione a otro jugador de la lista de jugadores conectados y le envíe una invitación para unirse a una partida. El sistema mostrará un mensaje de invitación.
- RF5** Aceptar o rechazar invitación: esta funcionalidad permite al usuario aceptar o rechazar una invitación para unirse a una partida. En caso de que acepte la invitación pasa al room del juego creado al que se ha unido.
- RF6** Mostrar lista de juegos creados: esta funcionalidad permite mostrar los juegos creados que no han sido iniciados.
- RF7** Unirse a un juego creado: esta funcionalidad permite seleccionar de la lista de partidas creadas, la partida a la cual el jugador quiere unirse, pasando al *room* del juego creado al que se ha unido.
- RF8** Crear Partida: esta funcionalidad permite al jugador crear una partida con la asignatura, tema y tablero de su preferencia.
- RF9** Mostrar jugadores contrarios: esta funcionalidad permite que se muestren a todos los jugadores en la partida, contra quienes están jugando.
- RF10** Permitir comunicación entre jugadores de una partida mediante chat: esta funcionalidad permite a los jugadores conectados a una partida, comunicarse entre si antes de iniciar la partida.
- RF11** Iniciar partida: esta funcionalidad permite al creador de la partida iniciar la misma una vez se hayan incorporado al menos el número mínimo de jugadores necesarios.
- RF12** Pasar turno de jugador: Habilita para que juegue a aquel jugador al cual le corresponde su turno, e inhabilita a los demás, los que permanecerán a la espera de que les corresponde su turno para jugar.



RF13 Deshabilitar ficha al desconectarse: esta funcionalidad permite que al desconectarse un jugador, por cualquier motivo, este sale de la partida.

RF14 Actualizar puntuación de jugadores: esta funcionalidad va sumando a cada jugador los puntos obtenidos por cada respuesta correcta.

RF15 Mostrar evaluación por jugador en la partida: Permite a cada jugador visualizar su puntuación así como la de sus oponentes durante la partida.

2.4.2 Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Son características que hacen al producto atractivo, usable, rápido o confiable (Pressman 2012). Seguidamente se especifican los requisitos no funcionales del sistema a desarrollar.

Apariencia o interfaz externa

- ✓ La aplicación debe contar con una interfaz sencilla siguiendo una arquitectura de información que le permita al usuario trabajar cómodamente aún cuando no posean conocimientos avanzados sobre el manejo en computadoras; cada funcionalidad debe estar descrita explícitamente con un texto en correspondencia con la función que realiza y las interfaces deben estar combinadas correctamente en cuanto a los colores, tipo de letra y tamaño.

Seguridad

- ✓ La información manejada por el sistema está protegida de acceso no autorizado y divulgación. En este caso los usuarios que accedan a la aplicación deben autenticarse, mediante el directorio activo (LDAP) de la UCI y una vez autenticados podrán interactuar con las funcionalidades que ofrece el sistema.

Portabilidad

- ✓ Las tecnologías utilizadas en el sistema son multiplataforma, por lo que puede ser desplegado en sistemas operativos que tengan soporte para estas tecnologías

Software (Cliente)

- ✓ Navegador web Mozilla Firefox 26 o superior (recomendado) o Google Chrome 30

Software (Servidor)

- ✓ Se debe disponer de los sistemas operativos Windows 7 o superior, y Linux, en ambos casos debe tener instalada la Máquina Virtual de Java a partir de la versión 7.0.
- ✓ Servidor web Apache versión 2.0 o superior.



Hardware (Cliente)

- ✓ Requerimientos mínimos: procesador Pentium IV o superior. Microprocesador con una velocidad a 2.0 GHz o superior. Memoria RAM de 512 MB o superior.
- ✓ Conexión de red.

Hardware (Servidor)

- ✓ Para el servidor de aplicaciones se requiere de una computadora con procesador de 1.80GHZ o superior, 1 GB de memoria RAM y capacidad de disco duro de 60 GB como mínimo.

Restricciones de diseño e implementación

- ✓ Lenguaje de programación: Java.
- ✓ Entorno Integrado de Desarrollo Netbeans a partir de la versión 7.1 o superior.
- ✓ Máquina virtual de Java: JDK 7.0.
- ✓ Servidor de bases de datos PostgreSQL 9.2.4.
- ✓ Servidor web Apache Tomcat 7.0.34.
- ✓ Entorno de programación en la capa del servidor basado en el lenguaje de programación Javascript Node js 0.10.24.
- ✓ Visual Paradigm 8.0 para UML 2.1.
- ✓ Spring 3.0.6.

2.5 Modelado 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.

Para iniciar el modelado de la aplicación se realiza una descripción de los actores que intervienen en el sistema.

Tabla 1 Usuario del sistema y su descripción

Actor	Descripción
ESTUDIANTE	Usuario autenticado que puede, a partir de la previa selección del modo multijugador, un tablero y de un sistema de preguntas correspondiente a un tema, ejercitar sus conocimientos y evaluarse durante el juego.

2.5.1 Modelo de casos de uso del sistema del módulo JDProg



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.

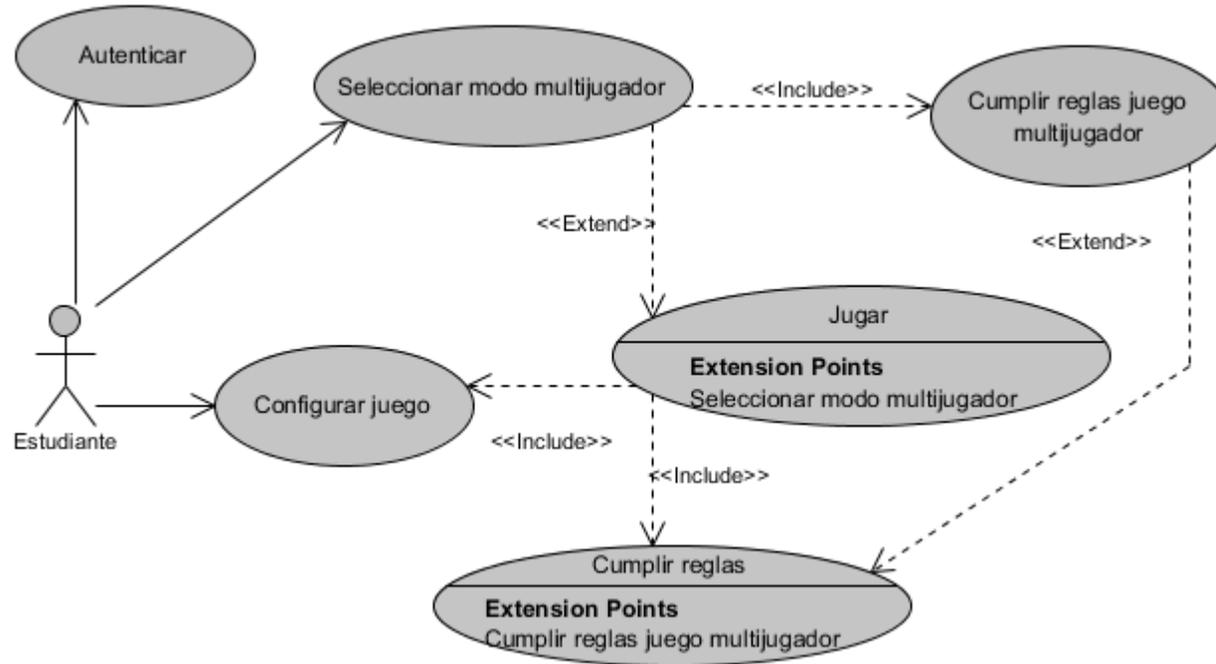


Figura 2. Diagrama de casos de uso del sistema

2.5.2 Descripción de los casos de usos del sistema

Un caso de uso contiene una descripción textual de todas las maneras que los actores previstos podrían interactuar con el sistema. Se redacta con lenguaje informal y no describen ni explican cómo se implementará, simplemente muestran los pasos que el actor sigue para realizar una tarea.

En este epígrafe se muestra la descripción del Caso de Uso Jugar multijugador, para consultar los Casos de Uso Autenticar y Cumplir reglas multijugador **(Ver Anexo #3 y #4)**.

2.5.2.1 Descripción textual del caso de uso Jugar multijugador

Tabla 2. Caso de uso Jugar multijugador

Caso de uso	Jugar multijugador
Actores	Estudiante



Resumen	<p>El caso de uso inicia cuando el estudiante escoge en la opción Modos de Juego la variante Multijugador. El sistema ejecuta las acciones correspondientes, luego de establecer un Nick y de haber seleccionado el tablero, el tema, la asignatura y accede a las opciones que le permiten enviar una invitación a un jugador determinado, carga el juego según la configuración especificada y permite la interacción con el mismo, finalizando así el caso de uso.</p> <p>En caso de escoger Unirse a un juego creado, el sistema muestra un listado de los juegos creados identificados con el nombre de la partida. El estudiante escoge al juego que desea unirse y el sistema ejecuta las acciones correspondientes permitiendo la interacción con el juego, finalizando así el caso de uso.</p>
Precondiciones	El estudiante debe estar autenticado.
Referencias	RF2, RF3, RF4, RF5, RF6, RF7, RF8, RF9, RF10, RF11
Prioridad	Crítica
CURSO NORMAL DE EVENTOS	
<u>Acción del actor</u>	<u>Respuesta del sistema</u>
1-Selecciona la opción "Modos de Juego. 2-Escoge "Multijugador".	3-Muestra una interfaz para que el jugador establezca un Nick.
4-El jugador introduce el Nick y presiona el botón "OK, a jugar!!"	5-Muestra una interfaz con las variantes "Crear partida" y "Unirse a un juego creado".



<p>Prototipo de interfaz</p>	
<p>Sección "Crear una partida"</p>	
<p>6-Introduce nombre de la partida. 7-Selecciona la Asignatura.</p>	<p>8-El sistema busca la asignatura seleccionada y muestra el listado de temas correspondientes a esa asignatura en otro campo de selección.</p>
<p>9-Selecciona el Tema.</p>	<p>10-Carga las preguntas asociadas a ese tema.</p>
<p>11-Selecciona el Tablero.</p>	<p>12-Carga el tablero seleccionado con el sistema de preguntas asociado al tema especificado.</p>
<p>13-Preiona el botón "Crear partida".</p>	<p>14-Muestra una interfaz que contiene un chat para comunicarse con los jugadores que se van uniendo a la partida y la lista de estos, además de la opción "Invitar a un jugador".</p>
<p>15-Escoge "Invitar a un jugador".</p>	<p>16-Muestra una ventana con los jugadores conectados que no están en partida.</p>
<p>17-Selecciona al jugador y presiona</p>	



<p>el botón "Invitar"</p> <p>18- El usuario que crea la partida Presiona en botón "Iniciar"</p>	<p>19 Carga el tablero seleccionado las preguntas asociadas del tema escogido para dicho tablero.</p> <p>20-Muestra el tablero y la lista de usuarios conectados listo para jugar.</p>
<p>Prototipo de interfaz</p>	
<p>CURSO ALTERNO DE EVENTOS</p>	
	<p>5.1-Si el Nick ya existe, el sistema muestra un cartel de que el Nick ya está en uso.</p> <p>5.2-Si el jugador no introduce un Nick, el sistema muestra una alerta de que introduzca un Nick de al menos 4 caracteres.</p>
	<p>14.1-Si el campo del nombre de la partida está vacío, muestra un error diciendo que hay que establecerle un nombre a la partida</p>



	19.1-Si el jugador que crea la partida presiona el botón “iniciar sin que nadie se conecte a la partida, se muestra un cartel informando que tiene que tener al menos 2 jugadores para iniciar la partida”.
Sección “Unirse a una partida creada”	
CURSO NORMAL DE EVENTOS	
	6-Muestra una lista de las partidas creadas.
7-Selecciona la partida a la cual se va a unir y presiona el botón “Unirse”	8- Muestra una interfaz que contiene un chat para comunicarse con los jugadores que se van uniendo a la partida y la lista de estos.
9- El usuario que crea la partida Presiona en botón “Iniciar”	10-Carga el tablero y selecciona las preguntas asociadas al tema escogido para dicho tablero. 11-Muestra el tablero y la lista de usuarios conectados listo para jugar.
Prototipo de interfaz	
CURSO ALTERNO DE EVENTOS	
	9.1 Si el jugador que crea la partida presiona el botón “iniciar sin que nadie se conecte a la partida, se muestra un cartel informando que tiene que tener al menos 2 jugadores para iniciar la partida”.



Poscondiciones	El entorno de juego queda listo con los elementos escogidos para jugar.
----------------	---

2.6 Diseño del sistema

En la fase del diseño se indica cómo se construirá la solución definiendo y estructurando una arquitectura que responda a la satisfacción de los requisitos funcionales y no funcionales, también se realiza la identificación de los objetos lógicos que componen al sistema para su posterior implementación.

2.6.1 Principales arquitecturas de red que soportan el modo multijugador

Una arquitectura de red es un marco para la especificación de los componentes físicos de una red y de su organización funcional y configuración. Las principales arquitecturas de red que dan soporte al modo multijugador son la arquitectura cliente/servidor y la arquitectura punto a punto (*peer to peer*), las cuales se describen brevemente a continuación.

Arquitectura cliente/servidor en 3 capas

Esta arquitectura implica la existencia de una relación entre procesos que solicitan servicios (clientes) y procesos que responden a estos servicios (servidores). La principal ventaja de esta arquitectura es que facilita la separación de las funciones según su servicio en 3 capas (Mora 2002):

- ✓ Lógica de presentación: se encarga de la entrada y salida de la aplicación con el usuario.
- ✓ Lógica de negocio: gestiona los datos a nivel de procesamiento actuando de puente entre el usuario y los datos.
- ✓ Lógica de datos: se encarga de gestionar a nivel de almacenamiento asegurando la integridad de los mismos.

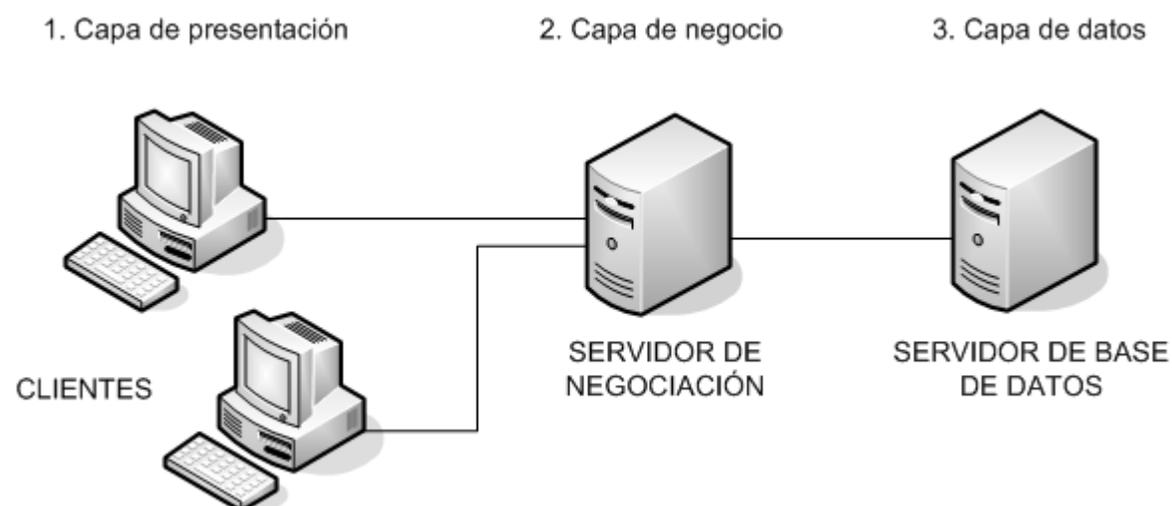


Figura 3 Arquitectura cliente/servidor en 3 capas



Arquitectura *peer to peer*

Una red *peer to peer* no trabaja con clientes o servidores fijos, sino una serie de nodos que se comportan a la vez como clientes y como servidores de los demás nodos de la red, por lo que mejora las comunicaciones en tiempo real, la colaboración, la distribución de contenido y el procesamiento distribuido. Como no depende de un servidor, es más robusta, con una escalabilidad potencialmente infinita (Steinmetz 2005). Cuenta con la desventaja de que consume gran cantidad de ancho de banda en las peticiones haciendo el tiempo de respuesta mayor y muy variable, el bajo nivel de estandarización debido a que se necesita interactuar con aplicaciones de diferente soporte y la mayor desventaja en los juegos multijugador son los problemas de escalabilidad, debido a que en una red que se disponga de ancho banda limitado el número de jugadores será limitado, lo que provoca el poco uso de este paradigma en los juegos (Mora 2002).

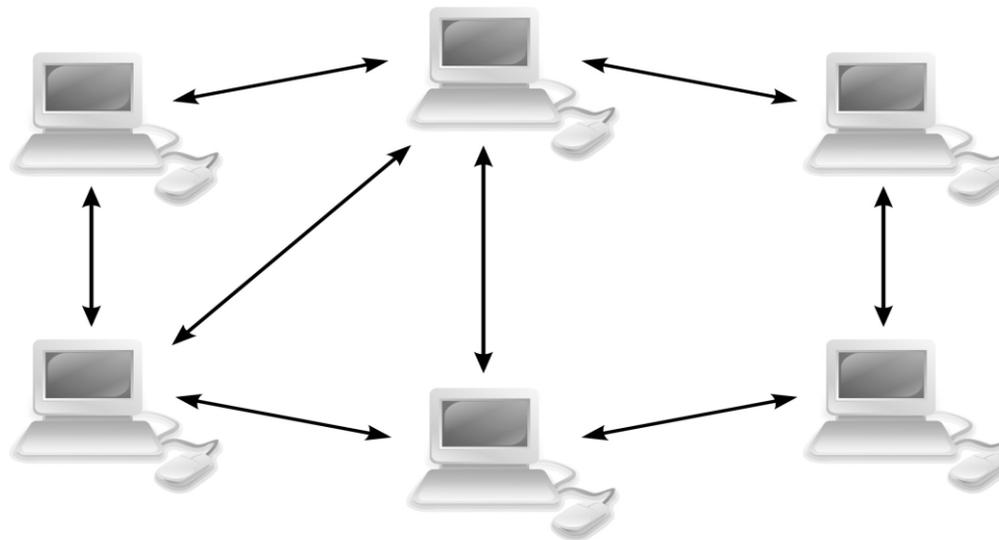


Figura 4 Arquitectura Peer to Peer

Para la implementación del modo multijugador en el módulo JDProg de la herramienta educativa SMProg, se escogió la arquitectura de red cliente/servidor, porque es la que se adapta a nuestras necesidades de un servidor que recibe peticiones y las muestra al cliente mediante un navegador web. Además hay un gran beneficio en esta estructura, incluyendo el hecho de que el ancho de banda requiere crecer solo linealmente con el número de jugadores, el juego puede ser también protegido de varias trampas o engaños debido a estar corriendo en un servidor seguro y fiable.

2.6.2 Patrones de diseño

Los patrones de diseño son valorados debido a que imponen reglas sobre la arquitectura y expresan esquemas para solucionar problemas de un mismo tipo, que pueden presentarse durante el desarrollo de un software.



En la presente investigación se utilizaron los patrones GRASP, los cuales serán explicados en este epígrafe. GRASP es un acrónimo que significa *General Responsibility Assignment Software Patterns* (Patrones Generales de Software para Asignar Responsabilidades), donde una responsabilidad es un método en los artefactos UML, por lo que los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos. Esta familia está compuesta por nueve patrones: experto, creador, alta cohesión, bajo acoplamiento, controlador, polimorfismo, fabricación pura, indirección y no hables con extraños (Larman 2003). Entre los patrones que serán explicados de manera sintetizada utilizados para el diseño del sistema se encuentran:

Creador: responsable a la creación de nuevas instancias de objetos de alguna clase. Este patrón es empleado 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 la clase que contiene la información necesaria para ello. Su uso permite crear las dependencias mínimas necesarias entre las clases, beneficiando el mantenimiento del sistema y brindando mejores oportunidades de reutilización. Por ejemplo en el sistema se manifiesta cuando se crea una nueva instancia de la clase *ConfiguracionAccionJuegoMultijugador* que contiene objetos de otras clases, las cuales para ser instanciadas necesitan del suministro de la información necesaria.

Experto: el patrón Experto indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para realizarla (Larman 2003). La clase *EscogerAccionMultijugador* es el experto en la creación de los multijugadores pues la misma contiene la información suficiente para la confección de un multijugador así como los métodos asociados a dicha funcionalidad.

Controlador: un controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema y define para ello el método de su operación. La mayor parte de los sistemas reciben eventos de entrada externa, los cuales generalmente incluyen una interfaz gráfica para el usuario dirigido por una persona, para manejar estos eventos de entrada se eligen los controladores (Larman 2003). En este caso la clase *Controlador*, contiene un conjunto de métodos correspondientes a las funcionalidades que ofrece el multijugador.

Alta Cohesión: la cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con funciones estrechamente relacionadas que no realicen un trabajo enorme (Larman 2003). Está basado en la asignación de responsabilidades teniendo en cuenta que la cohesión permanezca alta. Su utilización facilita la comprensión del diseño e incrementa las capacidades de reutilización.

Bajo Acoplamiento: plantea que debe existir una alta reutilización entre las funcionalidades de las clases con una mínima dependencia, contribuyendo al mantenimiento de las mismas. Este patrón es fundamental siempre que se desee realizar un diseño de clases independientes que puedan soportar los cambios de manera sencilla. El empleo de los patrones Experto y Creador favorecen al bajo acoplamiento entre las clases del sistema.



2.7 Descripción de la arquitectura

La arquitectura es la representación del diseño e implementación de estructuras de sistemas informáticos de alto nivel. Define los componentes funcionales que la integran y las interacciones entre ellos. Es el resultado de ensamblar cierto número de elementos arquitectónicos de forma adecuada y orientados a satisfacer los requerimientos funcionales y no funcionales de un determinado sistema (Amo 2005).

La propuesta de solución para facilitar la ejercitación de contenidos en la disciplina TPC en la web, se diseña sobre una arquitectura por capas (específicamente de tres capas), estilo de programación que tiene como objetivo fundamental la separación de la lógica del negocio de la lógica del diseño, a partir de la utilización de un patrón arquitectónico. Esta se considera ventajosa, pues durante su aplicación el desarrollo se puede llevar a cabo en varios niveles, y en el caso de que ocurra algún cambio este sólo afectará el nivel requerido.

Capa de presentación: es aquí donde se realiza la presentación del sistema al usuario, esta debe comunicar y capturar la información de este en un mínimo de procesos, comunicándose únicamente con la capa de negocio, en ocasiones es nombrada interfaz gráfica.

Capa de negocio: en esta capa residen las funcionalidades que debe realizar la aplicación, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio porque es donde se establecen todas las reglas que deben cumplirse. Esta mantiene comunicación con la capa de presentación para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él.

Capa de datos: es donde se accede a los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Patrón arquitectónico Modelo-Vista-Controlador (MVC)

La implementación estará regida por el patrón arquitectónico Modelo-Vista-Controlador con el objetivo de mantener la independencia de los datos y brindar la posibilidad de incluir nuevos juegos didácticos, y poder modificar funcionalidades e interfaces de la forma más sencilla posible.

El modelo es el encargado de agrupar el estado de la aplicación, responder a los requerimientos, así como mostrar la funcionalidad y/o notificar los cambios ocurridos en la vista. Esta última no sólo interpreta el modelo, sino que solicita actualizaciones necesarias, enviando las peticiones de los usuarios al controlador. Los controladores funcionan uno por método, definiendo de este modo el comportamiento del sistema, mapeando las acciones de los usuarios a actualizaciones del modelo y seleccionando finalmente la interfaz que le será mostrada al usuario.

El sistema de ejercitación y evaluación de contenidos está implementado sobre la base del marco de trabajo Spring, gran exponente del patrón arquitectónico MVC.

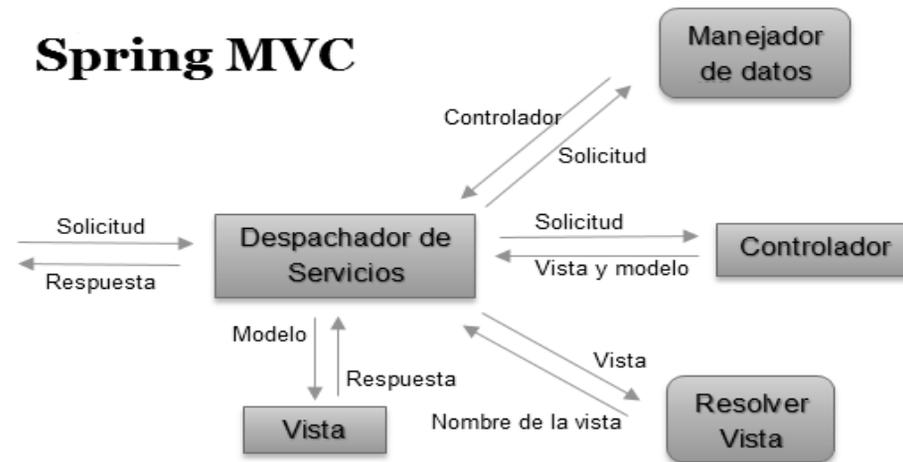


Figura 5. Spring MVC

Para aplicar el patrón MVC vinculado a Spring se utiliza lo siguiente:

- ✓ Una capa de vista, formada de JSP, HTML, CSS, con elementos de JavaScript y etiquetas personalizadas.
- ✓ Una capa de modelo, que cuenta con las subcapas de servicios, persistencia de datos (Data Access Object (DAO))⁵ y dominios (beans) (García 2008).

Sin embargo, se necesita una capa de control, que se compone de lo siguiente:

- ✓ *DispatcherServlet*: controlador frontal, que recibe y gestiona todas las peticiones (*request*). Resulta oculto al programador y es creado por Spring.
- ✓ *HandlerMapping*: analiza cada petición y determina el controlador que la gestiona, mediante varios manejadores en función de las diversas estrategias de mapeo.
- ✓ Controladores: cada controlador recibe las peticiones correspondientes a su página, delega en el dominio y recoge los resultados. Lo que hace es devolver un modelo a la vista que ha seleccionado (por medio del controlador frontal).

Lo que devuelve cada controlador es un objeto del tipo *ModelAndView*. Este se compone de lo siguiente:

- ✓ Una referencia a la vista destino.

⁵ El DAO es el Data Access Object, es decir, será la clase donde resida la lógica de manejo de Hibernate. De esta forma la lógica del negocio es independiente de Hibernate, y siempre que se quiera acceder a los datos lo hará usando este tipo de clase.



- ✓ El modelo: un conjunto de objetos utilizados para componer (*render*) la vista destino; por ejemplo, un bean o dominio en el sistema en cuestión sería Temas o una lista de beans (temas) que se ha obtenido de un DAO.

2.8 Diagramas 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 las interfaces involucradas. Estos diagramas contienen información útil para el usuario como por ejemplo clases, asociaciones, navegabilidad y dependencia existente (Larman 2003).

2.8.1 Diagrama de clases: CU Autenticar

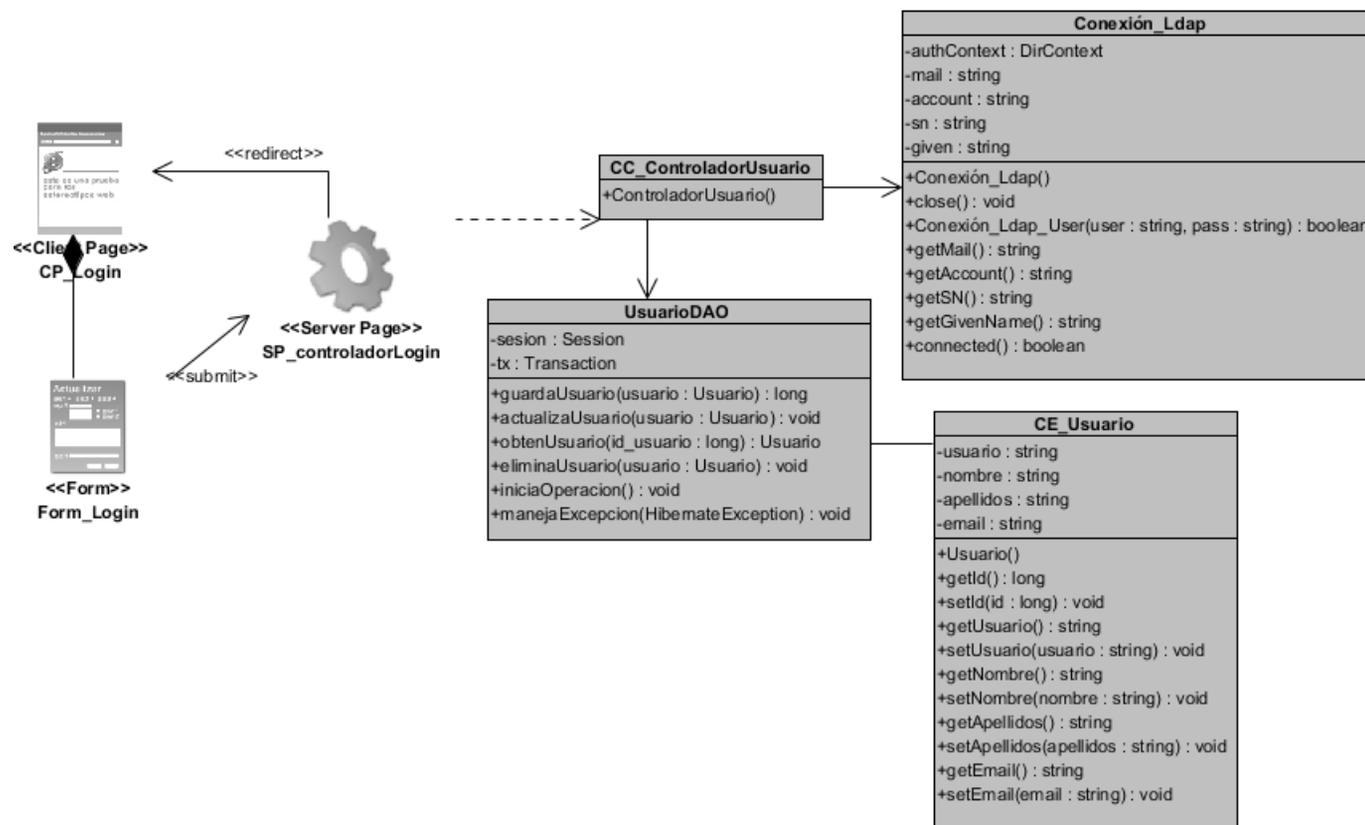


Figura 6. Diagrama de clases: CU Autenticar

2.8.2 Diagrama de clases: CU Jugar multijugador

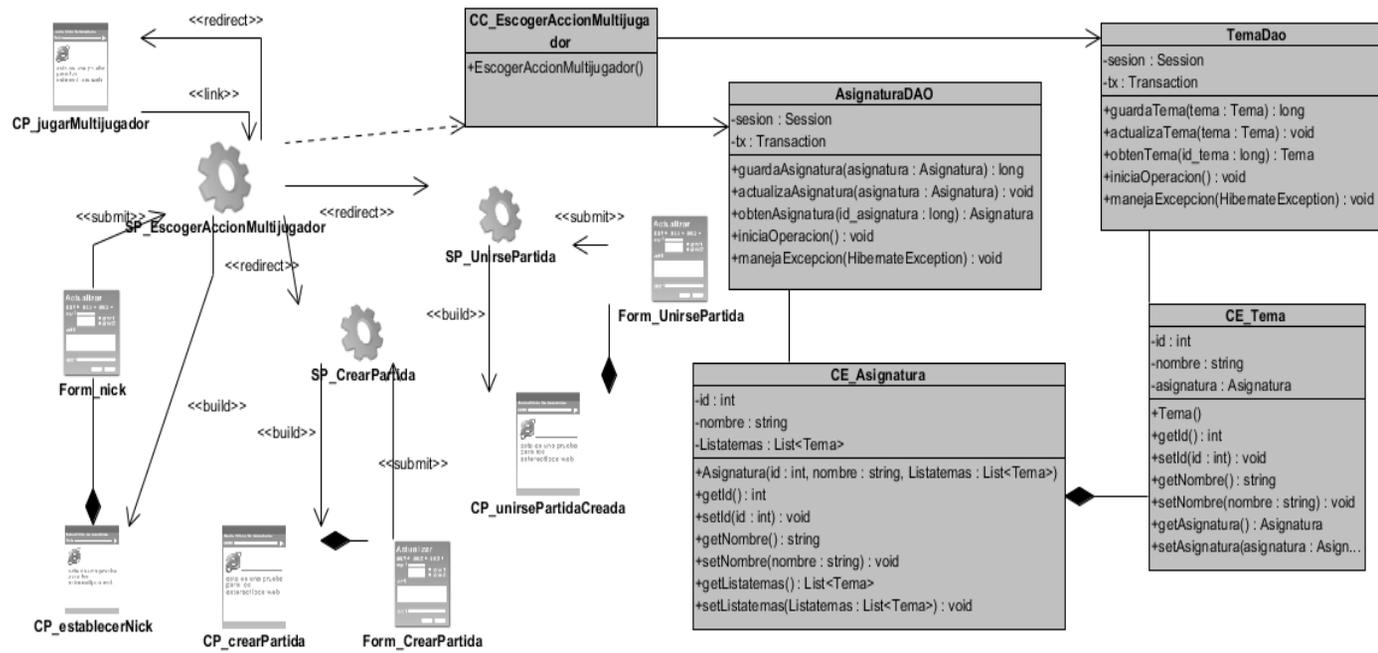


Figura 7. Diagrama de clases: Jugar multijugador

2.8.3 Diagrama de clases: CU Cumplir reglas multijugador

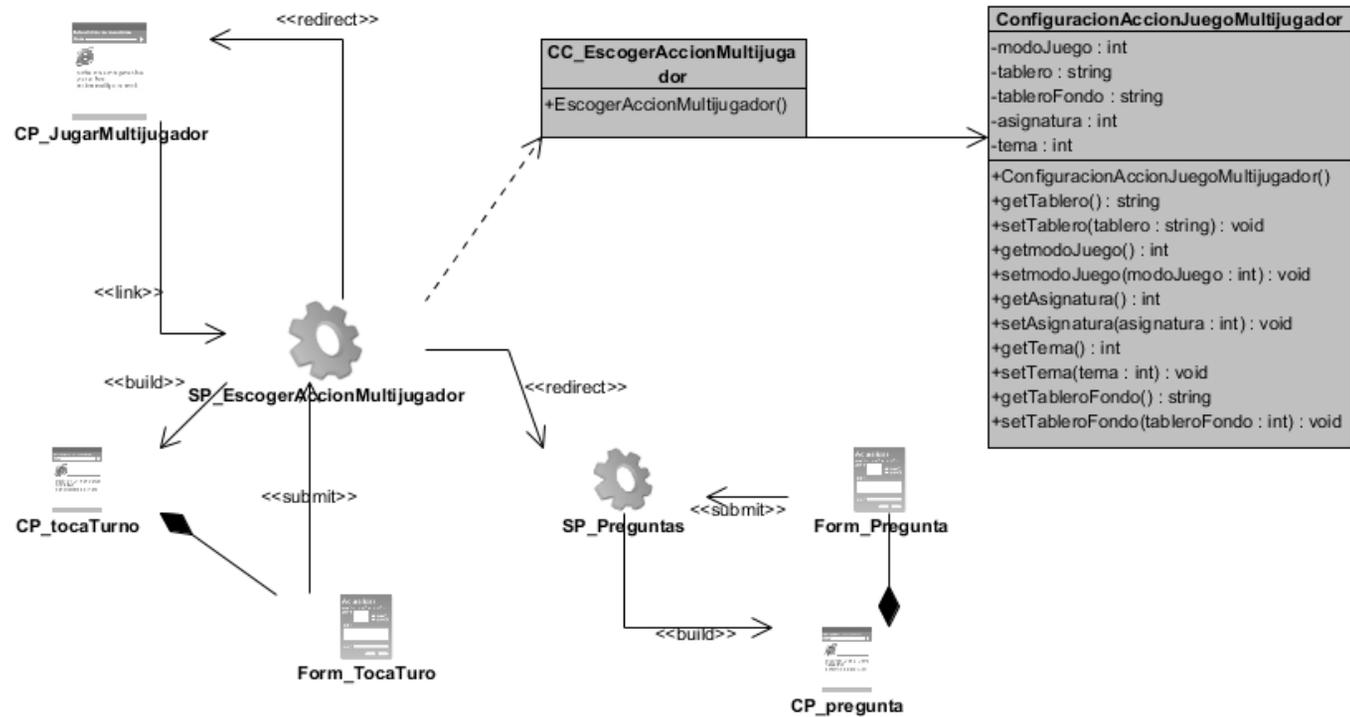


Figura 8. Diagrama de clases: CU Cumplir reglas multijugador



2.9 Diagramas de secuencia

Un diagrama de secuencia muestra una interacción que está organizada como una secuencia temporal. En particular, muestra la interacción existente entre los objetos que participan definiendo sus líneas de vida y los mensajes que intercambian. Su objetivo es brindar a los programadores, sin incurrir en demasiado nivel de detalle, lo necesario para la realización de los casos de uso. Este artefacto ahorra tiempo de diseño, ya que su realización no requiere demasiadas especificaciones y simplifica de manera considerable los diagramas.

El siguiente diagrama simula la secuencia que se realiza al efectuar el caso de uso Jugar multijugador en la sección Unirse a un juego creado. Para una mejor comprensión de los diagramas realizados [\(Ver Anexo #5 y #6\)](#).

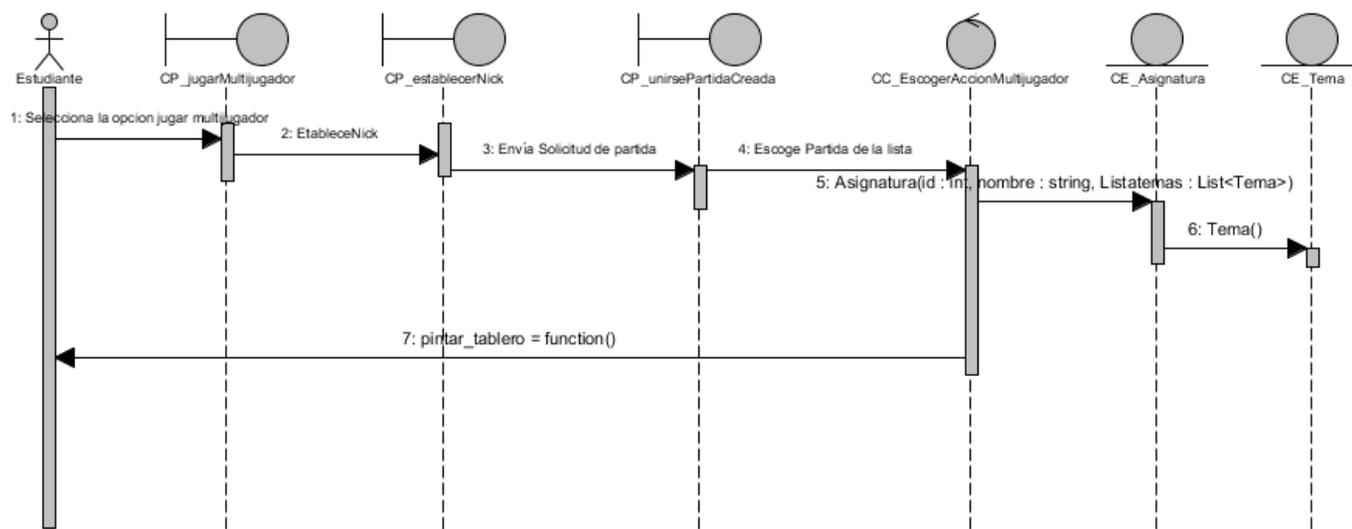


Figura 9. Diagrama de secuencia: CU Jugar multijugador sección Unirse a un juego creado

2.10 Modelo de datos

Un modelo de datos permite describir las estructuras de datos de la Base de Datos (BD), el tipo de los datos y la forma en que se relacionan. Un modelo de datos siempre está orientado a resolver un problema determinado, por lo que los dos enfoques propuestos son necesarios en cualquier desarrollo de software.

Los patrones de diseño de una BD ayudan a conformar la guía que permite al usuario diseñar y crear una BD más fortalecida, donde de estos aspectos dependen su calidad y comportamiento futuro. A continuación se define el patrón que se utilizó.

Patrón de llaves subrogadas

Este plantea que se debe generar una llave primaria única para cada entidad, en vez de usar un atributo identificador en el contexto dado, lo que permite que las tablas sean fáciles de consultar a partir del identificador, pues todos tienen el mismo tipo en cada una de las tablas.



Bajo la estructura de la BD se encuentra el modelo de entidad-relación que utiliza el sistema, reflejado mediante las relaciones que existen entre las tablas, especificando la cardinalidad correspondiente a cada relación como se muestra en la Figura 10.

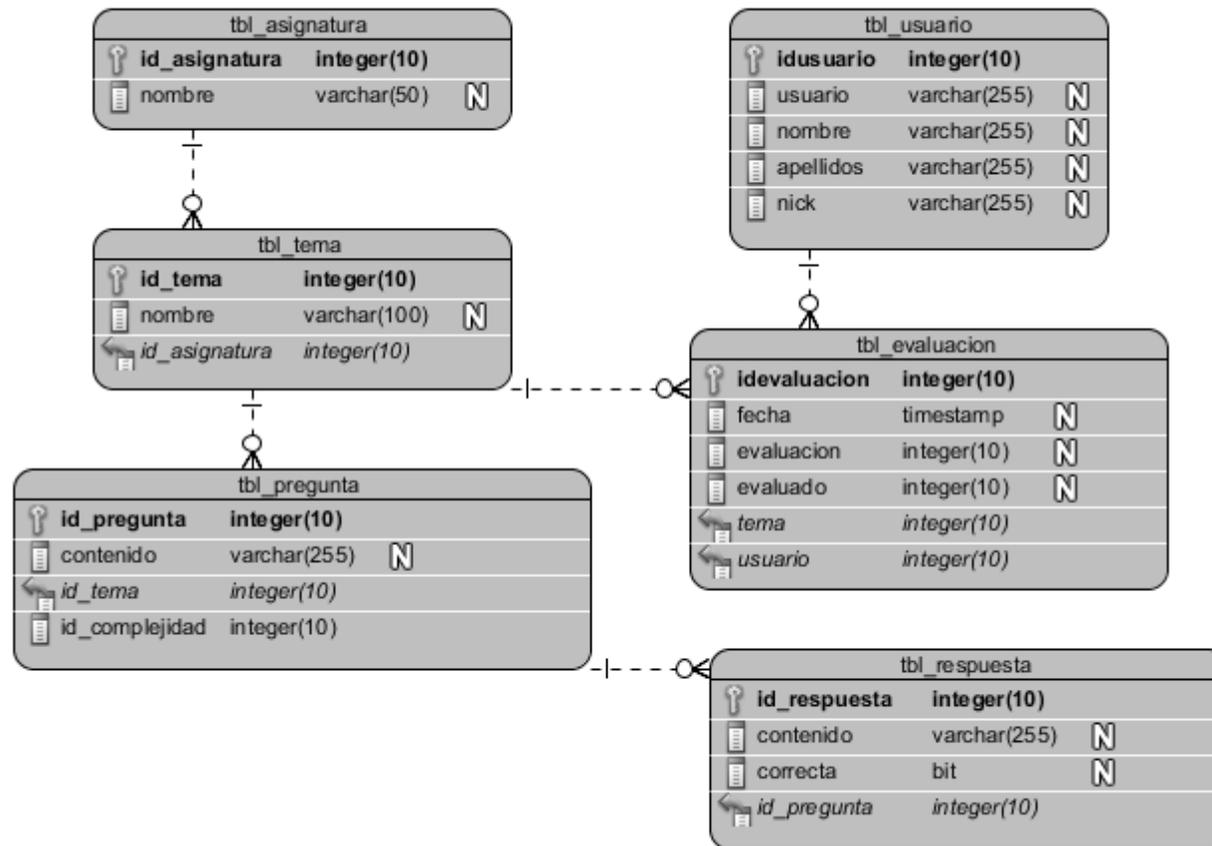


Figura 10. Modelo entidad-relación

2.11 Conclusiones parciales

El presente capítulo detalla la descripción del sistema propuesto. Se realizó el análisis del dominio de la aplicación, se definieron los requisitos funcionales y no funcionales que influyeron en la elaboración del Modelo de Caso de Uso del Sistema, se especifican los principales artefactos del análisis y diseño, así como los diagramas de clases y diagramas de interacción. También se describieron los patrones y arquitectura utilizada en el sistema.



Capítulo 3: Implementación y validación de la solución

3.1 Introducción

La implementación de un sistema inicia con el resultado del diseño y se construye en términos de componentes; es decir ficheros de código y ejecutables. Durante esta fase todos los componentes, características y requisitos deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del producto. El propósito principal de la implementación es desarrollar la arquitectura y el sistema como un todo.

En el presente capítulo se expresan los artefactos relacionados con la fase de iteración realizados mediante las tareas de ingeniería, visualizadas en forma de diagramas o interfaces de usuarios. Es aquí donde se efectúan las pruebas para validar el funcionamiento de los requisitos establecidos.

3.2 Estándar de codificación.

Un estándar de codificación consiste en reglas que se siguen para la escritura del código fuente. Para una mejor comprensión del código se definieron una serie de estándares basados en diversas reglas.

Estilos para la capitalización

Clases: Para todo nombre de clase, la primera letra debe de ser mayúscula, si son varias palabras se debe de intercalar entre mayúsculas y minúsculas. Este mecanismo de nombre es llamado camelCase.

```
public class ValidadorConfiguraci3nMultijugador implements Validator {
```

Figura 11 Estándar de codificación camelCase para el nombre de las clases.

Métodos: Para los métodos de clases la primera letra debe ser minúscula, si son varias palabras se debe de intercalar entre minúsculas y mayúsculas.

Variables: Para las variables se aplica el caso de los métodos, donde la primera letra es minúscula, y las demás se deben de guiar por el mecanismo de camelCase, deben ser cortos y descriptivos en sí mismo.



```
@Override
public void validate(Object o, Errors errors) {
    ConfiguracionJuego cj = (ConfiguracionJuego) o;

    boolean dirTableroFondo = false;
    boolean tablero = false;
    boolean asignatura = false;
    boolean tema = false;
    boolean modoJuego = false;
```

Figura 12. Estándar de codificación *camelCase* para métodos y variables.

Llaves y espacios.

Las llaves de las distintas estructuras lógicas se encuentran al mismo nivel del código que las contiene. También son utilizadas sobre líneas separadas y no sobre la misma línea del código que las contiene.

Se ha definido usar una línea en blanco para separar agrupaciones lógicas del código y entre cada método dentro de una clase.

Estructuras de control.

Entre las estructuras de control y los paréntesis se consideró que debía existir un espacio, ya que se recomienda utilizar siempre llaves de apertura y cierre incluso en situaciones en las que técnicamente son opcionales, en aras de lograr la legibilidad y disminuir la probabilidad de errores lógicos.

3.3 Diagrama de componentes

Un diagrama de componentes es, como su nombre lo indica, un esquema o diagrama que muestra las interacciones y relaciones de los componentes de un modelo. Entendiéndose como componente a una clase de uso específico que puede ser implementada desde un entorno de desarrollo, ya sea de código binario, fuente o ejecutable.

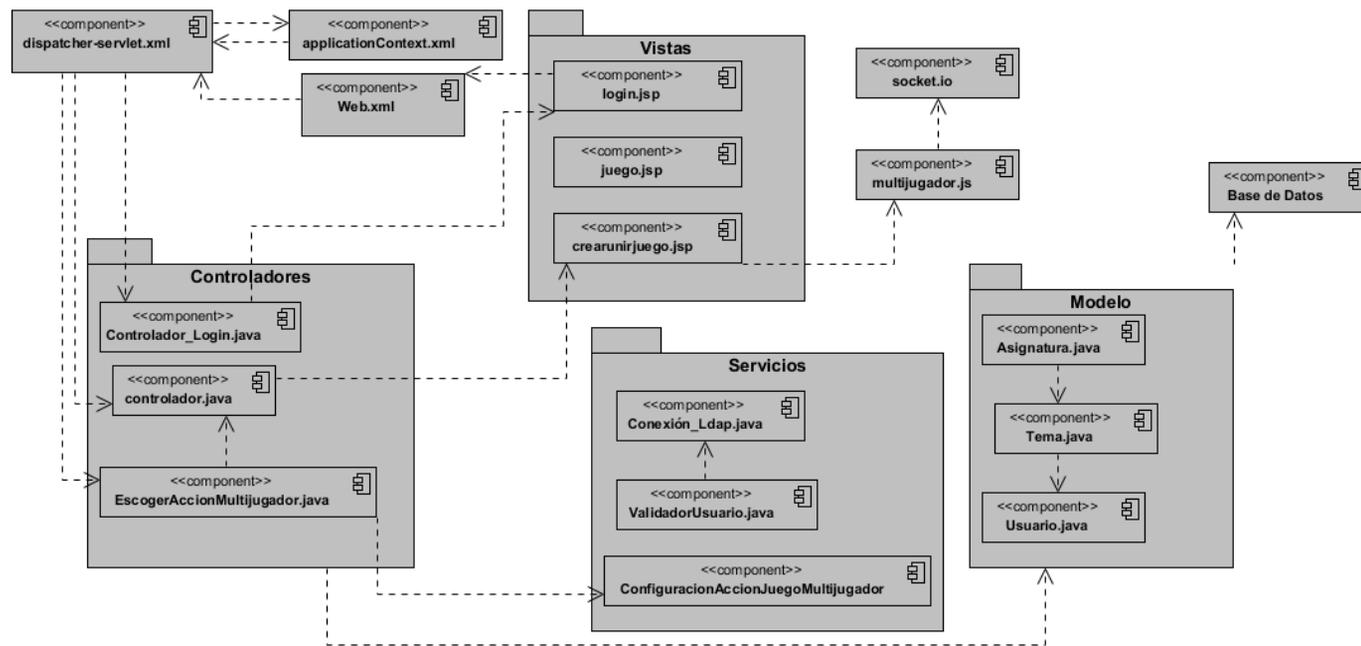


Figura 13 Diagrama de componentes

3.3.1 Descripción del diagrama de componentes.

El módulo JDProg de la herramienta educativa SMProg cuenta con 4 paquetes en los que se incluyeron los componentes del modo multijugador, los cuales se muestran en la figura anterior.

Dentro del paquete **Controladores** se encuentran los controladores encargados de manejar los eventos de entrada de la capa **Vistas** durante la interacción del usuario con las interfaces que brinda el sistema.

El paquete **Servicios** agrupa los componentes encargados de la entrada a la aplicación, así como los elementos de configuración del juego.

El componente **multijugador.js** es el que implementa toda la lógica respecto a la parte del multijugador, desde aquí se crea una conexión al servidor Node con el cliente de socket.io, es el que permite las funcionalidades del multijugador en tiempo real.

En el **Modelo** se especifican los componentes contenedores de información sobre los datos manejados.

3.4 Diagrama de despliegue

El diagrama de despliegue es utilizado para capturar los elementos de configuración del procesamiento y las conexiones entre ellos, describiendo la arquitectura física del sistema durante la ejecución en términos de procesadores, dispositivos y componentes de software.

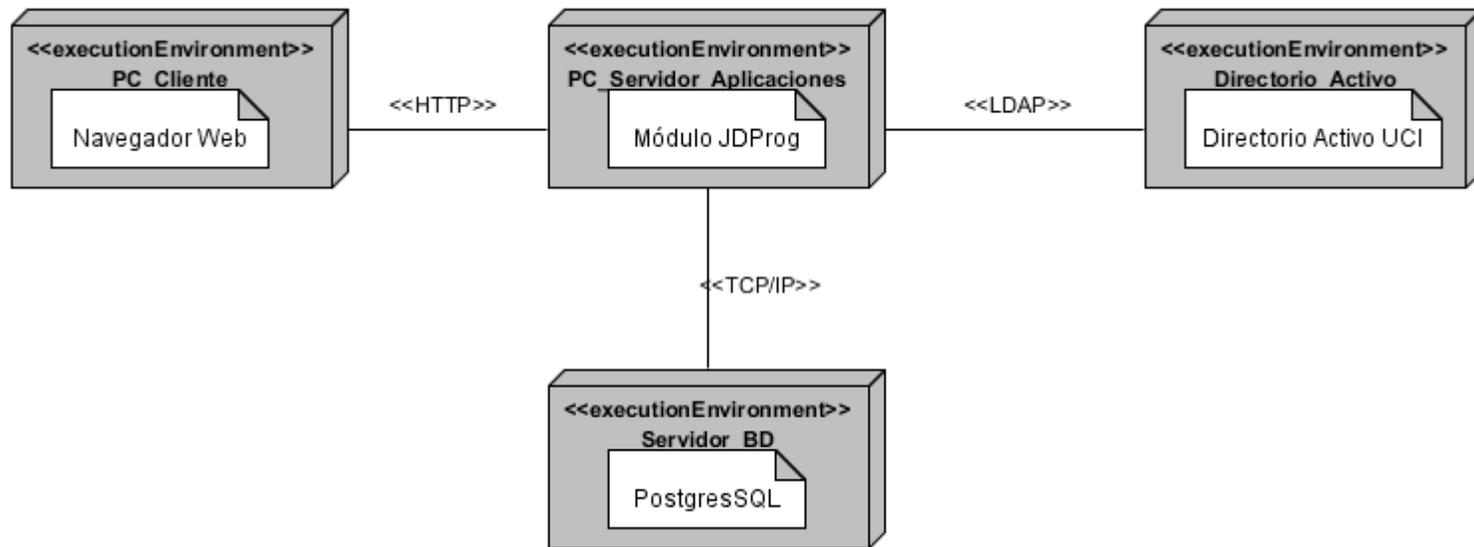


Figura 14. Diagrama de despliegue

Descripción de los nodos

- ✓ Nodo **Cliente**: en este nodo físico estará alojado el navegador web con el que interactúa el usuario para acceder a las interfaces del sistema.
- ✓ Nodo **Servidor de Juego**: en este nodo estará desplegado el servidor de aplicaciones Apache Tomcat y el gestor de base de datos PostgreSQL 9.2.4. En el caso particular de este sistema, en este nodo se encuentran los componentes correspondientes a la capa de negocio mostrados en el diagrama de despliegue.
- ✓ Nodo **Servidor de BD**: es donde se encuentra la Base de Datos (PostgreSQL) que utiliza el sistema para la ejercitación y evaluación de contenidos para almacenar, de forma estructurada, información asociada al tratamiento de los datos manejados en el juego.

Descripción de las relaciones entre los nodos

Inicialmente el usuario tendrá acceso al sistema JDPProg al interactuar con la PC cliente por la comunicación que se establece con el servidor de juego, utilizando un navegador web y el protocolo HTTP, lo cual posibilita ejercitar los contenidos según el rol del usuario autenticado contra el directorio activo (LDAP) de la UCI. Luego el servidor de aplicaciones establece una conexión con el servidor de base de datos utilizando el protocolo TCP/IP, para de este modo acceder a la información almacenada en la Base de Datos dando respuesta a la petición realizada por este servidor.



3.5 Pruebas de software

Las pruebas de software son procesos que permiten verificar la calidad de un producto. Se utilizan para identificar errores de implementación o usabilidad de los programas computacionales. Es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y codificación del mismo.

Las Pruebas que se le realizan a los productos de software tienen como objetivo:

- ✓ Verificar la interacción entre objetos.
- ✓ Verificar la interacción apropiada de todos los componentes del software.
- ✓ Verificar que todos los requisitos hayan sido implementados correctamente.
- ✓ Identificar y asegurar que los defectos se hayan resuelto antes del despliegue del software.

A continuación se efectúan las pruebas de caja negra, con el objetivo de Detectar posibles errores en el módulo JDProg.

3.5.1 Pruebas de caja negra

Las pruebas de Caja Negra son aplicadas principalmente a los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejecuten completamente todos los requisitos funcionales de un programa, ignorando las estructuras de control. En el módulo JDProg con la modalidad multijugador se aplican para mejorar la calidad de la aplicación al descubrir en su interfaz la mayor cantidad posible de errores, funciones incorrectas o faltantes, así como posibles errores de comportamiento o desempeño para su posterior corrección.

3.5.1.1 Partición Equivalente

La partición equivalente es un método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de pruebas. Un caso de prueba ideal descubre de forma inmediata una clase de errores, que de otro modo requerirían la ejecución de muchos casos antes de detectar el error genérico.

La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número de casos de prueba que hay que desarrollar.

Casos de prueba

Un caso de prueba permite detallar la forma en que se va a probar el sistema, incluyendo los datos de entrada con que se realiza la prueba correspondiente, las condiciones de ejecución y resultados obtenidos.

Los casos de prueba de caja negra pretenden demostrar que:

- ✓ Las funciones del software son operativas.



- ✓ La entrada se acepta de forma correcta.
- ✓ Se produce una salida correcta.
- ✓ La integridad de la información externa se mantiene.

Diseño de casos de prueba

En este epígrafe se realiza a partir de la técnica de prueba basada en los casos de uso, un caso de prueba por cada uno de estos.

Para ver los restantes casos de pruebas **(Ver Anexo #7 y #8)**.

Caso de prueba para el CU Jugar multijugador

Descripción general: el CU inicia cuando el usuario escoge la opción Multijugador.

Condiciones de ejecución: el usuario debe estar autenticado

Tabla 3 Caso de prueba CU Jugar multijugador Sección Crear una nueva partida

Sección Crear una nueva partida			
Escenario	Entrada	Resultados esperados	Resultados obtenidos
EC1 Llenar el campo del Nick.	Introducción por el usuario de un Nick.	Al insertar correctamente el Nick y pulsar el botón "Ok a jugar" la aplicación entra de forma correcta a la pantalla para que el usuario introduzca el nombre de la partida y seleccione la asignatura, el tema y el tablero.	Satisfactorio
EC2 insertar un Nick que ya está en uso	Introducción por el usuario de un Nick que ya está siendo usado por otro jugador.	El sistema muestra un mensaje de error "El Nick ya está en uso "	Satisfactorio
EC3 Establecer el nombre de la partida y escoger la	El usuario inserta un nombre para la partida y selecciona una	Al insertar el nombre de partida y escoger las demás opciones presiona en botón "Crear partida" el sistema muestra una pantalla	Satisfactorio



asignatura, tema y tablero.	asignatura, un tema y el tablero que desee.	con el nombre de la partida creada.	
EC4 Dejar el campo del nombre de la partida vacío.	El usuario no inserta un nombre en el campo del nombre de la partida.	Al presionar el botón de crear partida sin establecer un nombre de la partida el sistema muestra el mensaje "la partida debe tener un nombre".	Satisfactorio
EC5 insertar un nombre de partida que este en uso.	El usuario ingresa un nombre de partida que ya está siendo usado por otro jugador.	El sistema identifica que el nombre de la partida ya está en uso y muestra un cartel "ese nombre de partida ya existe, por favor inserte uno nuevo".	Satisfactorio
EC6 Invitar un jugador a una partida.	Después de creada la partida, el usuario escoge la opción "Invitar a un jugador".	El sistema muestra una ventana con los jugadores conectados al sistema que no están en otras partidas.	Satisfactorio
EC7 Seleccionar al jugador a invitar.	De la lista de usuarios conectados, el usuario selecciona al jugador que desea invitar.	El sistema envía un mensaje al jugador seleccionado de que ha sido invitado a unirse a una partida.	Satisfactorio



EC8	Iniciar partida.	Al contar con al menos 2 jugadores conectados el creador de la partida inicia la misma.	El sistema muestra el tablero donde se efectúa el juego con las fichas de correspondientes a cada jugador e indicando con una alerta al que le toca el 1er turno de tirar los dados.	Satisfactorio
EC9	Iniciar partida con un solo jugador en ella.	El usuario inicia la partida sin esperar a que estén al menos 2 jugadores conectados.	El sistema muestra el mensaje de error "para iniciar la partida deben existir al menos 2 jugadores conectados".	Satisfactorio

Tabla 4 Caso de prueba CU Jugar multijugador Sección Unirse a una partida creada

Sección Unirse a un juego creado			
Escenario	Entrada	Resultados esperados	Resultados obtenidos
EC1 Llenar el campo del Nick.	Introducción por el usuario de un Nick.	Al insertar correctamente el Nick y pulsar el botón "Ok a jugar" la aplicación entra de forma correcta a la pantalla para que el usuario introduzca el nombre de la partida y seleccione la asignatura, el tema y el tablero.	Satisfactorio
EC2 insertar un Nick que ya está en uso.	Introducción por el usuario de un Nick que ya está siendo usado por otro jugador.	El sistema muestra un mensaje de error "El Nick ya está en uso".	Satisfactorio



EC3 Seleccionar unirse a una partida creada.	El usuario escoge la opción de unirse a una partida creada.	El sistema muestra una lista de las partidas creadas.	Satisfactorio
EC4 Seleccionar a la partida que se desea unir.	El usuario selecciona a la partida que desea unirse.	Al presionar el botón de "Unirse" el sistema muestra una pantalla con las personas que ya están conectadas a esa partida.	Satisfactorio
EC5 Iniciar partida.	Al contar con al menos 2 jugadores conectados el creador de la partida inicia la misma.	El sistema muestra el tablero donde se efectúa el juego con las fichas de correspondientes a cada jugador e indicando con una alerta al que le toca el 1er turno de tirar los dados.	Satisfactorio

El método partición equivalente permite dividir la etapa de prueba en iteraciones con el objetivo de que en cada iteración se pueda solucionar las no conformidades de la iteración anterior. En este caso fueron comprobadas las funcionalidades del módulo JDProg en tres de ellas. Para una primera iteración se obtuvieron siete no conformidades las cuales fueron solucionadas, en una segunda iteración se detectaron tres no conformidades las cuales fueron también solucionadas, y en una tercera iteración no se obtuvieron no conformidades, para un resultado satisfactorio.

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

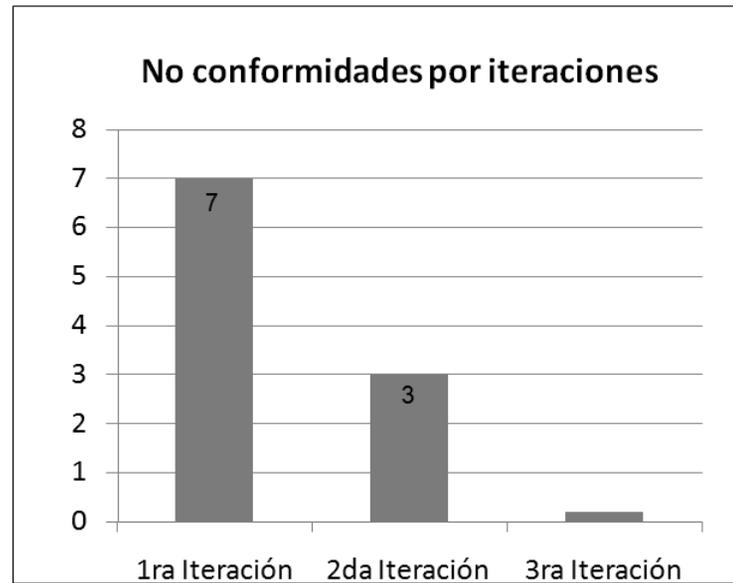


Figura 15 Gráfico de No conformidades por iteraciones

3.5.2 Pruebas de rendimiento

Las pruebas de aceptación se encuentran dirigidas a evaluar la conformidad de un sistema o componente con requerimientos específicos. Su desarrollo por lo general se hace utilizando una herramienta de prueba automática para simular un gran número de usuarios, carga y volumen de información; aunque en el sistema en cuestión se enfatiza en el uso de esta prueba para monitorear el desempeño del software. En este caso para comprobar el funcionamiento de la aplicación propuesta se utilizó la herramienta JMeter versión 2.9, teniendo en cuenta que se destaca dentro de las herramientas de libre distribución empleadas en la UCI para efectuar pruebas de rendimiento sobre aplicaciones web.

Esta presenta una estructura en árbol, permitiendo que sea la imaginación de quien la use la que ponga los límites a la hora de diseñar el plan de prueba, brindando mayor cantidad de variantes para recoger los resultados obtenidos, lo que permite hacer un análisis exhaustivo de las pruebas ejecutadas.

Entre los tipos de pruebas para verificar el rendimiento de un sistema se destacan:

Pruebas de carga

Las pruebas de carga consisten en probar el funcionamiento del software bajo condiciones extremas. Estudia la especificación del software, las funciones que debe realizar, las entradas y las salidas analizando los valores límites.

Pruebas de estrés



Las pruebas de estrés están diseñadas para enfrentar al programa a condiciones anormales, donde se ejecute de manera que demande recursos en cantidad, frecuencia y volumen.

Es preciso realizar dichas pruebas al sistema a partir de la necesidad de comprobar el rendimiento del mismo soportando la cantidad de usuarios que accedan a la aplicación, así como su comportamiento al aumentar esta carga con los mismos recursos disponibles.

3.6 Resultados de las pruebas

Para realizar dichas pruebas se preparó un servidor con las siguientes propiedades: Procesador AMD con una velocidad de 1.65 GHz. Memoria RAM: 3 GB. Capacidad: 160 GB.

El ordenador cliente se probó con las siguientes propiedades: Procesador Pentium IV con una velocidad a 1.65 GHz. Memoria RAM: 512 Mb. Capacidad: 80 Gb.

La prueba fue realizada a una muestra de 50 usuarios haciendo peticiones concurrentemente y los resultados se muestran en la figura 16, siendo estos satisfactorios.

Informe Agregado

Nombre: Informe Agregado

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos Config

Etiqueta	# Muestras	Media	Mediana	Línea de 90%	Mín	Máx	% Error	Rendimiento
/SistemaJD...	50	557	2167	4325	152	4715	0.00%	9.0/sec
/SistemaJD...	50	3099	710	8704	108	9847	0.00%	2.4/sec
/SistemaJD...	50	1904	1386	44471	11996	49366	0.00%	55.6/min
/SistemaJD...	50	355	189	823	105	2482	0.00%	1.2/sec
/SistemaJD...	50	314	474	32104	6754	40676	0.00%	55.8/min
/proyectos/j...	50	141	104	261	20	505	0.00%	1.8/sec
/socket.io/	50	1057	1052	1100	1000	1274	0.00%	4.6/sec
/socket.io/?...	50	1056	1055	1101	996	1282	0.00%	3.4/sec

Figura 16 Resultado de las pruebas

3.7 Conclusiones parciales

La realización de los diferentes diagramas de ingeniería durante el desarrollo de este capítulo posibilitó la definición de los elementos físicos relevantes para el despliegue del sistema, donde quedaron definidos también los tipos y casos de pruebas realizados al sistema, permitiendo comprobar su funcionamiento. Con las pruebas ejecutadas a la aplicación se validó y verificó las funcionalidades descritas, dando solución a errores encontrados durante su ejecución, comprobando que el sistema cumple los requerimientos definidos.



Conclusiones generales

Una vez concluida la presente investigación se arriba a las siguientes conclusiones:

1. El estudio sobre la utilización de los juegos didácticos en el Proceso de Enseñanza Aprendizaje posibilitó comprender mejor la importancia del presente trabajo, así como determinar las características fundamentales con que debe contar la solución final.
2. La investigación realizada para la selección de las metodologías y herramientas permitió escoger las más adecuadas y acordes con la arquitectura del sistema SMProg.
3. La incorporación del modo multijugador al módulo JDProg permite la interacción de los estudiantes en una misma partida, incentivando la utilización de esta herramienta para ejercitar los conocimientos.
4. El modelo arquitectónico utilizado, y la utilización de cada patrón de diseño proporciona uniformidad, robustez y flexibilidad a la solución desarrollada, al mismo tiempo que favorecen la rigurosidad en su implementación.
5. Mediante la aplicación de las pruebas de software al módulo JDProg con el modo multijugador se detectaron varios errores, los cuales fueron corregidos aumentando así la calidad del producto final.
6. La culminación del presente trabajo constituye un aporte a los esfuerzos que se realizan en nuestro país, especialmente en nuestra universidad, por avanzar en lo referente al Proceso de Enseñanza Aprendizaje mediante la utilización de técnicas novedosas como los juegos didácticos.



Recomendaciones

Como resultado del proceso de investigación e implementación, surgieron recomendaciones a tener en cuenta para un futuro perfeccionamiento del sistema, a continuación se listan las mismas:

- ✓ Realizar la integración con los módulos GesCont y DesBoard para hacer uso del sistema de preguntas que conformen los profesores y sea efectiva la ejercitación de contenidos.
- ✓ Integrarle efectos de sonido a las preguntas y los obstáculos con el objetivo de crear un ambiente más alegre que motive la utilización de la herramienta para la ejercitación de contenidos.



Referencias bibliográficas

- Amo, F. A., Normand, L. M., y Segovia, F. J. (2005). El Proceso Unificado está centrado en la arquitectura. Introducción a la ingeniería del software, Delta publicaciones.
- Arana, J. (2012). "Conozca más sobre la tecnología Java ", from <http://www.scoop.it/t/innovacion-tecnologia-y-redes-sociales/p/3546134097/2012/11/30/conozca-mas-sobre-la-tecnologia-java>.
- Arcade, M. s. E. (2014). "The Radix Endeavor." from <http://mitstep.org/projects/radix-endeavor>.
- Arcila, J. B., Quintero, E. F., Aguilar, S. K. de., y De Peña, N. T. (2011) "La Simulación y los Juegos en Linea como Herramienta para la Inmersión Educativa." Dialnet.
- Carrero, A. (2013). "Conceptos básicos de ORM (Object Relational Mapping).Programación en Castellano."
- David, M., y Sande, C. (2005). Serious Games.Games that Educate, trains and informs. Thonsom, Canadá.
- Domínguez, F. R. (2012). El videojuego en red social: un nuevo modelo de comunicación., Universidad de Extremadura.
- EcuRed. (2010). from http://www.ecured.cu/index.php/Software_educativos_en_Cuba.
- Ferreiro, D. F. (2011). Herramienta para la creación y uso de juegos didácticos en la enseñanza de la Programación. Departamento Técnicas de Programación. La Habana, Universidad de las Ciencias Informáticas: 81.
- García, A. P. (2008) "Spring + Hibernate + Anotaciones = Desarrollo Rápido en Java."
- García, P. F., Herena, G. F., y Miguélez, R. G. (2008). Extensión a multijugador de ViRPlay 3D, un juego educativo de enseñanza de orientación a objetos usando role-play. Departamento de Ingeniería del Software e Inteligencia Artificial, Facultad de Informática Universidad Complutense de Madrid.
- Garrido, T. R., y Rodríguez, J. A. (2010). Infodrez - Módulo de Torneos Online, Universidad de las Ciencias Informaticas.
- Gauchat, J. D. (2013). "El gran libro de HTML5, CSS3 y Javascript .".
- Gayo, J. F. V., y Fernández, M. R. (2011) "TRINX: Desarrollo y Evaluación de Juegos Educativos Multijugador."
- Godfrey, R. J., y Eaves, J. (2003). Apache Tomcat Bible, Wiley India Pvt.
- Gross, K. (2008). El juego como escuela de la vida. MAGISTER.



- Guerra, Y. A. (2013). Conceptualización de una red social educativa que integre de forma colaborativa las aplicaciones e-learning de la Universidad de las Ciencias Informáticas. Revista Cubana de Ciencias Informáticas.
- Gutiérrez, J. J. (2005). ¿ Qué es un framework web? .
- Hechavarría, R. A., y Del Río, D. D. (2010). Análisis, diseño e implementación de la versión 2.0 de los juegos del primer ciclo de la colección multisaber. La Habana, Universidad de las Ciencias Informáticas.
- Herron, D. (2011). Node Web Development. A practical introduction to Node, the exciting new server-side JavaScript web development stack. S. Cullington.
- Jacobson, I. (2000). "El proceso unificado de desarrollo de software." 438
- Johnson, R., y Hoeller, J. (2004). Expert one-on-one J2EE development without EJB.
- Joomla-Chile. (2012). "Joomla! y sus lenguajes de programación: PHP, HTML, XHTML, SQL, XML, JavaScript, AJAX, CSS ", 2013, from <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-ja>.
- Larman, C. (2003). UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado, Pearson Educación.
- Mora, S. L. (2002). Programación de aplicaciones web: historia, principios básicos y clientes web.
- Ojeda, M. P., y Gago, Y. (2008). Experticias Criminalísticas del Sistema de Investigación e Información Policial, Universidad de las Ciencias Informáticas.
- Pressman, R. S. (2012). Ingeniería de requisitos.
- Rosario, J. (2010) "La tecnología de la información y la comunicación (TIC). Su uso como herramienta para el fortalecimiento y el desarrollo de la Educación Virtual."
- Sánchez, J. (2004). "Net Beans 3.51 guía rápida."
- Santos, J. I., Galán, J. M., Izquierdo, L., y Olmo, R. (2009) "Aplicaciones de las TIC en el nuevo modelo de enseñanza del EEES."
- Sitio Oficial. (2010). "Sobre Postgre SQL." from <http://www.postgresql.org.es/>.
- Steinmetz, R. (2005). Peer to Peer Systems and applications
- Teixeira, P. (2013). PROFESSIONAL Node.js. BUILDING JAVASCRIPT-BASED SCALABLE SOFTWARE.
- Velázquez, J. A., y Muñoz, Y. V. (2009). UCIScrabble: MÓDULO PARTIDA ENTRE 2 JUGADORES, Universidad de las Ciencias Informáticas



Visual Paradigm for UML (2012). "Visual Paradigm for UML 11.0 Modeler Editions."



Bibliografía consultada

1. Amo, F. A., Normand, L. M., y Segovia, F. J. (2005). El Proceso Unificado está centrado en la arquitectura. Introducción a la ingeniería del software, Delta publicaciones.
2. Arana, J. (2012). "Conozca más sobre la tecnología Java ", from <http://www.scoop.it/t/innovacion-tecnologia-y-redes-sociales/p/3546134097/2012/11/30/conozca-mas-sobre-la-tecnologia-java>.
3. Araya, R. (2012). Torneos de matemática y ciencias masivamente multijugador: experiencia y oportunidades desde la práctica. Pensamiento Educativo. Revista de Investigación Educativa Latinoamericana: 65-74.
4. Arcade, M. s. E. (2014). "The Radix Endeavor." from <http://mitstep.org/projects/radix-endeavor>.
5. Arcila, J. B., Quintero, E. F., Aguilar, S. K. de., y De Peña, N. T. (2011) "La Simulación y los Juegos en Línea como Herramienta para la Inmersión Educativa." Dialnet.
6. Calle, J. M. (2012). "Proyecto Newton. Área de Física y Química en Secundaria y Bachillerato.
7. Carrero, A. (2013). "Conceptos básicos de ORM (Object Relational Mapping).Programación en Castellano.
8. Cota, M. P. (2013). Resultados del estudio de requerimientos de la Didáctica para una aplicación de Mundos Virtuales en. [Uso de Mundos Virtuales para Educación]. Vigo, Universidad de Vigo.
9. Cyldigital, C. (2012). "Wikiduca, aprender vocabulario en inglés jugando".
10. David, M., y Sande, C. (2005). Serious Games.Games that Educate, trains and informs. Thonsom, Canadá.
11. Domínguez, F. R. (2012). El videojuego en red social: un nuevo modelo de comunicación., Universidad de Extremadura.
12. EcuRed. (2010). from http://www.ecured.cu/index.php/Software_educativos_en_Cuba.
13. Equipo de desarrollo, d., HAEduc. (2011).
14. Ferreiro, D. F. (2011). Herramienta para la creación y uso de juegos didácticos en la enseñanza de la Programación. Departamento Técnicas de Programación. La Habana, Universidad de las Ciencias Informáticas: 81.
15. García, A. P. (2008) "Spring + Hibernate + Anotaciones = Desarrollo Rápido en Java."
16. García, M. F. (2011). Implementación del nivel A de IMS-LD para la herramienta de autor web CRODA, Universidad de las Ciencias Informáticas.



17. García, P. F., Herena, G. F., y Miguélez, R. G. (2008). Extensión a multijugador de ViRPlay 3D, un juego educativo de enseñanza de orientación a objetos usando role-play. Departamento de Ingeniería del Software e Inteligencia Artificial, Facultad de Informática Universidad Complutense de Madrid.
18. Garrido, T. R., y Rodríguez, J. A. (2010). Infodrez - Módulo de Torneos Online, Universidad de las Ciencias Informáticas.
19. Gauchat, J. D. (2013). "El gran libro de HTML5, CSS3 y Javascript.
20. Gayo, J. F. V., y Fernández, M. R. (2011) "TRINX: Desarrollo y Evaluación de Juegos Educativos Multijugador.
21. Godfrey, R. J., y Eaves, J. (2003). Apache Tomcat Bible, Wiley India Pvt.
22. Gross, K. (2008). El juego como escuela de la vida. MAGISTER.
23. Guerra, Y. A. (2013). Conceptualización de una red social educativa que integre de forma colaborativa las aplicaciones e-learning de la Universidad de las Ciencias Informáticas. Revista Cubana de Ciencias Informáticas.
24. Gutiérrez, J. J. (2005). ¿ Qué es un framework web?
25. Hechavarría, R. A., y Del Río, D. D. (2010). Análisis, diseño e implementación de la versión 2.0 de los juegos del primer ciclo de la colección multisaber. La Habana, Universidad de las Ciencias Informáticas.
26. Herron, D. (2011). Node Web Development. A practical introduction to Node, the exciting new server-side JavaScript web development stack. S. Cullington.
27. Jacobson, I. (2000). "El proceso unificado de desarrollo de software." 438
28. Johnson, R., y Hoeller, J. (2004). Expert one-on-one J2EE development without EJB.
29. JOOMLA-CHILE. (2012). "Joomla! y sus lenguajes de programación: PHP, HTML, XHTML, SQL, XML, JavaScript, AJAX, CSS ", 2013, from <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-ja>.
30. Larman, C. (2003). UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado, Pearson Educación.
31. León, C. M. R. (2004). "Las esferas socioculturales del Software Libre.
32. Mora, S. L. (2002). Programación de aplicaciones web: historia, principios básicos y clientes web.
33. Moreno, J., Montaña, E. A., y Montoya, L. F. (2012). Creación y monitoreo de video juegos educativos multi-jugador masivos en línea, Universidad Nacional de Colombia – Medellín.



34. Ojeda, M. P., y Gago, Y. (2008). Experticias Criminalísticas del Sistema de Investigación e Información Policial, Universidad de las Ciencias Informáticas.
35. Padilla, N., González, J. L., Gutiérrez, F. L., Cabrera, M. J., y Paderewski, P. (2008). "Diseño de videojuegos educativos multijugador. Una visión desde el aprendizaje colaborativo.
36. Pressman, R. S. (2012). Ingeniería de requisitos.
37. Rosario, J. (2010) "La tecnología de la información y la comunicación (TIC). Su uso como herramienta para el fortalecimiento y el desarrollo de la Educación Virtual.
38. Sánchez, J. (2004). "Net Beans 3.51 guía rápida.
39. Santos, J. I., Galán, J. M., Izquierdo, L., y Olmo, R. (2009) "Aplicaciones de las TIC en el nuevo modelo de enseñanza del EEES.
40. Sitio Oficial. (2010). "Sobre Postgre SQL." from <http://www.postgresql.org.es/>.
41. Steinmetz, R. (2005). Peer to Peer Systems and applications.
42. Tabares, R. B. (2012). Revisión bibliográfica de los juegos digitales para el aprendizaje de la programación orientada a objetos, Facultad de Ingeniería, Tecnológico de Antioquia, Medellín (Colombia) pp. 91-106
43. Teixeira, P. (2013). PROFESSIONAL Node.js. BUILDING JAVASCRIPT-BASED SCALABLE SOFTWARE.
44. Tourn, G. (2013). "Jugar en tiempos de Facebook. Un análisis de los videojuegos de las redes sociales.
45. Vallés, M. G., y Azcárate, M. I. (2010). Motor de Juego Multijugador en red. Pamplona, ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN.
46. Velázquez, J. A., y Muñoz, Y. V. (2009). UCIScrabble: MÓDULO PARTIDA ENTRE 2 JUGADORES, Universidad de las Ciencias Informáticas.
47. Visual Paradigm for UML (2012). "Visual Paradigm for UML 11.0 Modeler Editions.



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:

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 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.

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.

CASE (*Computer Aided 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.

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.

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.

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...).



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.

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.

Modo Multijugador: es un modo de juego que posee cualquier videojuego que permita la interacción de 2 o más jugadores al mismo tiempo, ya sea de manera física en la misma consola, o mediante servicios en línea u otro tipo de red con personas conectadas a la misma.

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

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

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.

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.

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



Anexos

Anexo #1 Entrevista realizada a experto

Entrevista realizada a la jefa de departamento de programación de la Facultad 1 de la UCI MSc. Dunia Suárez Ferreiro.

Objetivo: Diagnosticar el estado actual del módulo JDProg perteneciente a la herramienta educativa SMProg.

Preguntas formuladas.

1. Sobre la aplicación de la herramienta:

1.1 ¿Considera usted que la aplicación de la herramienta educativa SMProg será útil como apoyo al PEA en su nivel de enseñanza? Argumente sus razones.

1.2 ¿Considera que será efectiva para la ejercitación de contenidos? Argumente sus razones.

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

2. Explique el funcionamiento del módulo JDProd de la herramienta educativa SMProg.

3. Según su experiencia. ¿Cree usted que la incorporación del modo multijugador al módulo JDProg de la herramienta educativa SMProg, haría más atractiva la ejercitación de contenidos? Argumente sus razones.

Anexo #2 Tipos de juegos didácticos del sistema

El sistema para la ejercitación de contenidos estilo tablero, está concebido para la utilización de tres tipos de tableros, en esta investigación solo se desarrolla el que se describe:

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.

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

El juego contiene casillas didácticas, las cuales estarán señalizadas con imágenes representadas por un signo de interrogación (?)



La aplicación será capaz de interpretar y manejar toda la lógica en este tablero de juego, mientras que el resto de los tableros podrán incluirse en versiones posteriores, siguiendo sus características:

- ✓ Sube y Baja: el tablero consta de casillas en las que se permanece, avanza o retrocede hacia otra casilla ubicada en el mismo.
- ✓ 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.

Anexo #3 Descripción textual de los casos de uso del sistema: C.U. Autenticar

Tabla 5. Caso de uso Autenticar

Caso de uso	Autenticar
Actores	Estudiante
Resumen	El caso de uso inicia cuando el estudiante accede y se autentica en el sistema, inserta los datos necesarios (usuario y contraseña) relativos al dominio UCI y accede a la interfaz de la aplicación, finalizando así el caso de uso.
Precondiciones	El estudiante debe estar registrado en el dominio UCI.
Referencias	RF1
Prioridad	Crítica
CURSO NORMAL DE EVENTOS	
<u>Acción del actor</u>	<u>Respuesta del sistema</u>
1-Introduce usuario y contraseña.	2-El sistema verifica que los datos de autenticación sean correctos y que el actor se encuentre registrado en el dominio UCI.
	3-Muestra la interfaz principal que brinda acceso a las funcionalidades que permiten la ejercitación de contenidos.



Prototipo de interfaz	
CURSO ALTERNO DE EVENTOS	
	2.1-Si hay campos vacios o con datos incorrectos, el sistema muestra un cartel de datos incorrectos con las letras en rojo.
2.2-El usuario introduce nuevamente los datos.	
Prototipo de interfaz	
Poscondiciones	Se muestra la pantalla donde el usuario escoge la opción de modo de juego "Multijugador"


Anexo #4 Descripción textual de los casos de uso del sistema: C.U. Cumplir reglas juego multijugador
Tabla 6. Caso de uso Cumplir reglas juego multijugador

Caso de uso	Cumplir reglas juego multijugador
Actores	Estudiante
Resumen	El caso de uso inicia cuando los actores conectados a la partida comienzan a jugar. El actor con el primer turno lanza el dado y responde las preguntas correspondientes a la casilla didáctica. Si la pregunta es respondida correctamente se inhabilitará la ficha del jugador pasando al que está esperando turno, si la respuesta de la pregunta es incorrecta no se sumarán los puntos y luego pasará al que está esperando turno. El sistema ejecuta las acciones correspondientes que permiten el cumplimiento de las reglas del juego multijugador (si se desconecta un usuario pierde la partida, comportamiento de las casillas, selección de preguntas, comprobación de la veracidad de las respuestas, movimiento del dado y de las fichas, cálculo de la evaluación de cada jugador en la partida) hasta que el juego sea culminado satisfactoriamente. Al concluir se muestra la evaluación obtenida en la partida, finalizando así el caso de uso.
Precondiciones	El estudiante debe estar autenticado.
Referencias	RF12, RF13, RF14, RF15, RF16
Prioridad	Crítica
CURSO NORMAL DE EVENTOS	
<u>Acción del actor</u>	<u>Respuesta del sistema</u>
	1-Escoge al jugador con el primer turno de forma aleatoria. 2-Muestra un cartel al jugador con primer turno que debe cliquear en el dado.
3-Da un clic en el dado.	4-el sistema muestra la ficha del jugador en la casilla según el número que sale en el dado
4.a.1- Selecciona la	4.a-Si la ficha cae en una casilla didáctica, muestra una pregunta de forma aleatoria con las posibles respuestas.



respuesta	4.a.2-Suma los puntos a la evaluación del jugador en la partida.
	4.b-Si la ficha cae en una casilla obstáculo. El sistema hará que el jugador pase 2 turnos.
	.5-Muestra un cartel al próximo jugador que está en turno indicándole que cliquee el dado.
Prototipo de interfaz	
CURSO ALTERNO DE EVENTOS	
	4.1-En caso que no sea el turno de un jugador y cliquee el dado el sistema muestra un cartel indicándole que no es su turno todavía.



<p>Prototipo de interfaz</p>	
<p>Poscondiciones</p>	<p>Se realiza la ejercitación de contenidos a través del juego con el modo multijugador.</p>

Anexo #5 Diagramas de secuencia

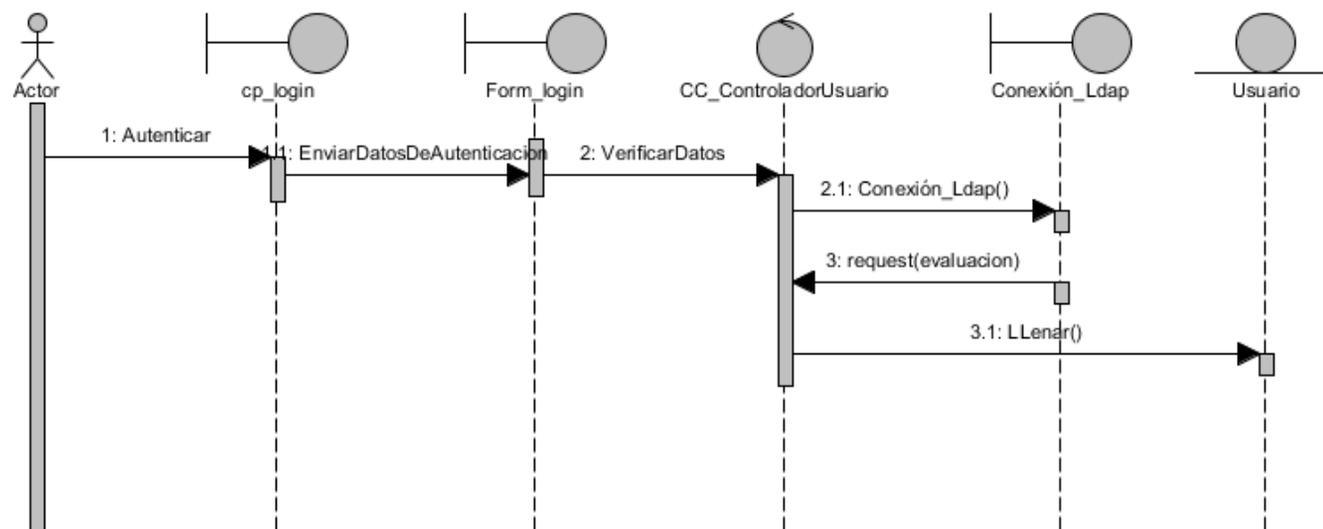


Figura 17. Diagrama de secuencia caso de uso Autenticar



Anexo #6 Diagramas de secuencia

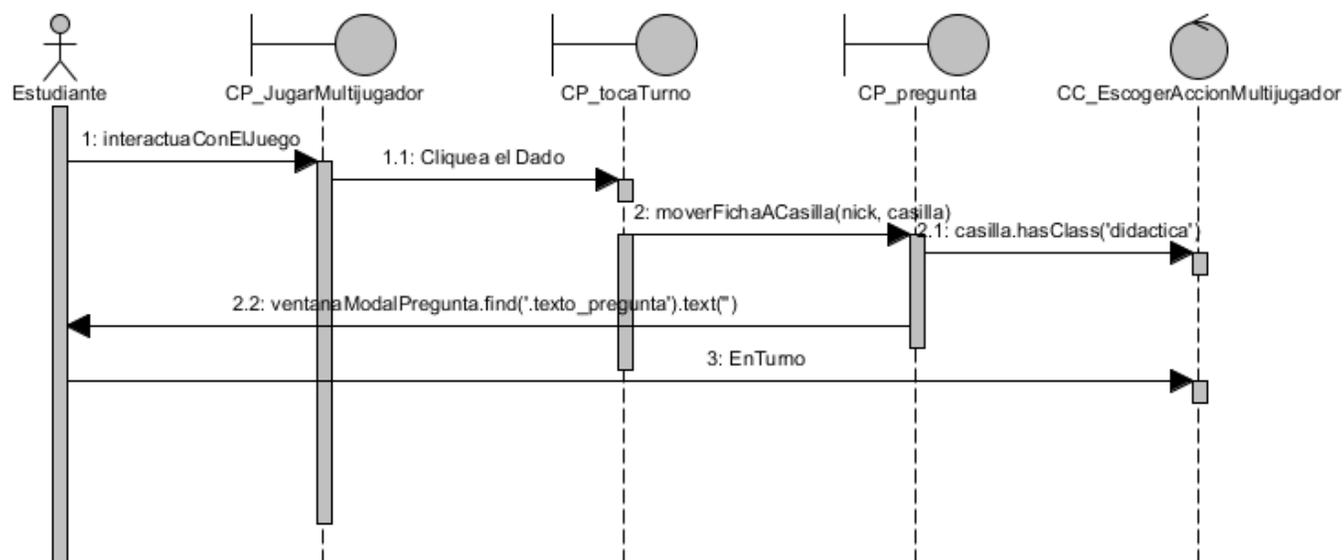


Figura 18 Diagrama de secuencia CU Cumplir reglas multijugador

Anexo #7 Caso de prueba CU Autenticar usuario

Descripción general

El CU inicia cuando el usuario entra sus datos para autenticarse.

Condiciones de ejecución.

Los usuarios del sistema deben estar creados y configurados con sus respectivos permisos.

Tabla 7 Caso de prueba CU Autenticar usuario

Escenario	Entrada	Resultados esperados	Resultados obtenidos
EC1 Llenar los campos correctamente	Introducción correcta de las credenciales del usuario: Usuario: eilinh Contraseña: *****	La aplicación de entrar de forma correcta a la página principal donde se escoge el modo multijugador	Satisfactorio



EC2 Dejar algún campo vacío o con los datos del usuario incorrectos	Introducción incorrecta de las credenciales de usuario: Usuario: vacío Contraseña: vacío	La aplicación muestra un mensaje de error "Datos incorrectos".	Satisfactorio
---	--	--	---------------

Anexo #8 Caso de prueba para el CU Cumplir reglas multijugador

Descripción general

El CU inicia cuando se inicia la partida.

Condiciones de ejecución.

Debe estar iniciada una partida

Tabla 8 Caso de prueba para el CU Jugar Multijugador

Escenario	Entrada	Resultados esperados	Resultados obtenidos
EC1.Tirar el dado al jugador que le toca el 1er turno.	Presionar el botón "iniciar partida"	La aplicación escoge de forma aleatoria al 1er jugador en tirar el dado mostrando el mensaje "Nick, es tu turno, por favor haz clic en el dado".	Satisfactorio
EC2 Dar clic en el dado para mover la ficha del jugador.	El usuario que está en turno da clic en el dado.	El sistema mueve la ficha del jugador en el tablero la cantidad de casillas que indico el dado.	Satisfactorio
EC3 Dar clic en el dado sin que sea el turno del	El usuario que aun no está en turno da un clic	El sistema muestra el mensaje error "Ups, lo siento todavía no es tu turno".	Satisfactorio



jugador.	en el dado.		
----------	-------------	--	--