

MASTER EN SOFTWARE LIBRE

PEC3_ESP_AN Software Libre

Proyecto de Investigación

Presentado Por:

SEGUNDO MANUEL BLANCO PALENCIA

Fecha de entrega

Enero 13 de 2015

Juego educativo enfocado al desarrollo de habilidades para aprender a modelar requerimientos con artefactos UML

*Segundo Manuel Blanco Palencia*¹, *Francisco Eugenio López*², *Alexandre Viejo Galicia*³
Universidad Abierta de Cataluña

RESUMEN

El siguiente trabajo describe los resultados de una experiencia práctica con un Serious Game llamado: “MODELA”, dirigido tanto a alumnos de Ingeniería de Software como a los profesores que trabajen en esa área, con la intención que practiquen como modelar requerimientos con casos de uso y el modelo correcto de un diagrama de caso de uso. Para este desarrollo se partió del planteamiento, bastante difundido por los teóricos de la pedagogía y la didáctica, que mediante los juegos educativos se llena el espacio vacío entre la teoría y la práctica, lo que permite asegurar que los Serious Games son herramientas excelentes para involucrar a los usuarios en los contenidos y valores que se quieren transmitir.

La ruta metodológica que se consideró para el desarrollo del juego, como trabajo de investigación, partió de una revisión bibliográfica sobre juegos serios y herramientas de tipo juego, así como de la identificación y análisis del modelo de aprendizaje que se espera que el estudiante desarrolle mediante el juego.

La experiencia permite concluir que los Serious Games son una herramienta innovadora, atractiva y con grandes posibilidades en el mundo de hoy para un proceso enseñanza-aprendizaje práctico y orientado a modelar “artefactos UML” con mayor consistencia en el área de ingeniería de software.

Palabras clave: Caso de uso, juegos, Requisito, Ingeniería de software, Serious Games.

Educational game focused on the development of skills to learn how to modeling requirements with UML artifacts

ABSTRACT

The following paper describes the result of a practical experience with a game called “MODELA”, which is aimed at engineering students as well as teachers who work in the same field. This game intends to give people the chance to practice how to model requirements with use cases and the appropriate model of a use case diagram. To carry out this study, the basic assumption that the gap between theory and practice is filled up through educative games was taken into consideration. This is a well-known premise related to didactic and pedagogy. This allows us to assure that *Serious Games* are excellent tools that involve users in the learning of the values and the contents that are intended to be conveyed.

The methodology used for the development of the game as a research work, started with the literature review about serious games, game tools, the identification and the analysis of the learning model that the student is expected to accomplish through the game.

¹ Segundo M. Blanco, Candidato a maestría en software libre, UOC. sblancopa@uoc.edu

² Francisco Eugenio López, Docente investigador, ITM, franciscolopez@itm.edu.co

³ Alexandre Viejo Galicia, Docente Consultor, UOC, aviejo@uoc.edu

This experience leads us to conclude that the Serious Games are an attractive and innovative tool with a wide range of possibilities offered by the current world to experience a realistic teaching and learning process oriented to model “artifacts UML” with greater consistency in the field of software engineering.

Key words: Use cases, games, requirement, Software Engineering, Serious Games.

1. INTRODUCCIÓN

La ingeniería del software es una de las áreas principales de la ingeniería y la informática que brinda métodos y técnicas para desarrollar y mantener software de calidad, además, aborda todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistema de información [1].

Lo que se pretende al poner en práctica la ingeniería del software, es obtener programas computarizados que sean funcionales, estables, confiables y seguros [2] y para ello utiliza el modelado (un conjunto de abstracciones o realidades seleccionadas que se construyen para entender un problema) como herramienta para la solución de problemas de la vida diaria, antes de implementar una solución. En Ingeniería del Software, el modelado se ha apoyado en análisis y diseño orientado a objetos [3], y más recientemente en UML [4], cuya versión 2.0 se basa en tres clases de diagramas: estructurales, de interacción y de comportamiento. Estos diagramas se pueden usar conjuntamente para modelar el mismo problema desde diferentes ópticas o puntos de vista, que pueden ser de tipo estructural o dinámico dependiendo del interesado que los emplee; para lograr este objetivo, tales diagramas deben ser coherente entre sí, de forma que los requisitos de una especificación no se contradigan entre sí [5].

Zapata y Awad [6] afirman que los ingenieros de software deben tener aptitudes de tipo administrativo, que poco se cultivan en la enseñanza tradicional. Hoy en día, la enseñanza no se basa sólo en los conceptos impartidos por el maestro, sino que se vuelca hacia el estudiante como elemento central de la clase. Es por ello que ahora se emplean los juegos como herramienta pedagógica.

Lo anterior justifica el planteamiento que afirma que en las últimas décadas se han vivido

importantes cambios en el proceso educativo y, más aun, en la educación superior. Estos nuevos retos imponen un enfoque socio constructivista y un modelo más flexible, móvil, pertinente y autónomo centrado más en aprender-haciendo que en enseñar [7].

2. JUEGOS EN LA ENSEÑANZA DE INGENIERÍA DE SOFTWARE

La ingeniería de software es un área de la informática que ofrece métodos y técnicas a través de los cuales se puede desarrollar y mantener software de calidad.

Para la enseñanza de estos conceptos se percibe que las metodologías tradicionales no son suficientes, ya que existen conceptos que no surgen de la parte teórica, pero sí pueden surgir de la parte práctica y la vivencia en los proyectos.

Para resolver este problema en la enseñanza de la ingeniería de software, Wankat y Oreovicz [8] propusieron un conjunto de estrategias didácticas para la enseñanza de la Ingeniería en general, entre las que se incluyen las clases magistrales y los proyectos prácticos; otras estrategias, tales como los juegos, los estudios de casos y la educación personalizada, poco se han aplicado en la enseñanza de la Ingeniería y menos en la enseñanza de la Ingeniería de Software. Estos autores, además, afirman que la clase magistral como estrategia didáctica requiere ser complementada por otras estrategias para alcanzar objetivos cognitivos de más alto nivel, y en esto coinciden con Rugarcia *et al.* [9], quienes sugieren la complementación de la enseñanza tradicional de la Ingeniería con nuevos métodos alternativos que ofrezcan buenas posibilidades de éxito.

Teniendo en cuenta lo anterior, los juegos educativos constituyen estrategias que pueden

complementar la enseñanza tradicional; no se trata de sustituir las clases magistrales y los proyectos prácticos, sino de suministrar otros espacios que permitan afianzar los conceptos que se imparten a partir de los modelos tradicionales del proceso enseñanza-aprendizaje.

Kober y Tarca [10] afirman que los juegos de simulación tienen muchas ventajas, ya que permiten que los estudiantes incrementen la motivación, desarrollen comunicación, y el trabajo en grupo. A su vez, Klassen y Willoughby [11] resaltan que los juegos en clase incrementan la velocidad de aprendizaje, aumentan la posibilidad de recordar conceptos, y mejoran la retención de los mismos. Los juegos de simulación, además de introducir al estudiante de manera controlada en el mundo real, permiten que el estudiante asuma un rol definido, que le permita tomar decisiones con seguridad y afrontar las consecuencias que ello implica. Existen juegos de simulación en áreas financieras y administrativas –como el juego de la cerveza– y también existen micromundos como el *beefeater* [12], que introducen al jugador en una situación determinada. En la Ingeniería de Software, se han desarrollado algunos juegos, entre los que se encuentran el Juego de los Requisitos [14] y *Problems and Programmers* [13]

3. MARCO CONCEPTUAL

3.1. Ingeniería del Software

La ingeniería del software es una disciplina o área de las ciencias de la computación muy especial, pues sus practicantes requieren conocimientos y habilidades ligadas simultáneamente con muchas áreas del conocimiento; en particular, se requieren habilidades en comunicación y administración de proyectos. La enseñanza de la ingeniería de software se ha realizado tradicionalmente con una combinación de clases expositivas y pequeños proyectos prácticos; también, se han usado estrategias como los estudios de casos con fines de instrucción. Sin embargo, las necesidades de la industria del software y por lo tanto del proceso enseñanza-aprendizaje, plantean nuevos retos educativos que requieren también una combinación de nuevas estrategias

complementarias a la educación tradicional, como son los juegos serios [15].

3.2. Lenguaje unificado de modelado (UML)

El lenguaje unificado de modelado o notación (UML) cubre varias fases en el proceso de desarrollo de un producto software. Este lenguaje está controlado por el grupo de administración de objetos OMG y adoptado como un estándar de descripción de esquemas de software, que ayuda a especificar, visualizar y documentar esquemas de sistemas de software orientado a objetos. Se puede usar UML para modelar negocios o algunos sistemas cuya solución no necesariamente sea un producto software [4]. UML versión 2.0 define 13 diagramas, que se dividen en tres categorías: estructura estática (clases, objetos, componentes, estructura de composición, paquetes, despliegue), que representan los elementos de una especificación sin tomar en consideración el tiempo, comportamiento (casos de uso, máquina de estados y actividades), que describen las características dinámicas de un objeto, tales como sus operaciones y métodos, e interacción (secuencias, comunicación, vista de interacción y tiempos), que se derivan de los diagramas de comportamiento y que hacen énfasis en la manera como los objetos interactúan entre sí [16]. El OMG (Object Management Group) en la versión 2.5, ha incorporado 3 nuevos diagramas (Diagrama de Modelo, Diagrama de Manifestación y Diagrama de Arquitectura de Red).

3.3. Modelado de Requisitos como parte Fundamental de la Ingeniería de Requisitos

La ingeniería de requisitos es uno de los procesos más importantes, pero a su vez más críticos, dentro del desarrollo de software ya que es donde se define el diseño de la solución según las necesidades del cliente [17].

La Ingeniería de requisitos nace de la necesidad de administrar y revisar requisitos de los clientes en el momento de hacer la especificación del producto. Para lograr un control más eficiente se generó una serie de pasos que se denominan *Ciclo de vida de los requisitos* (Figura 1) [18].

Figura 1. Proceso de la Ingeniería de requisitos

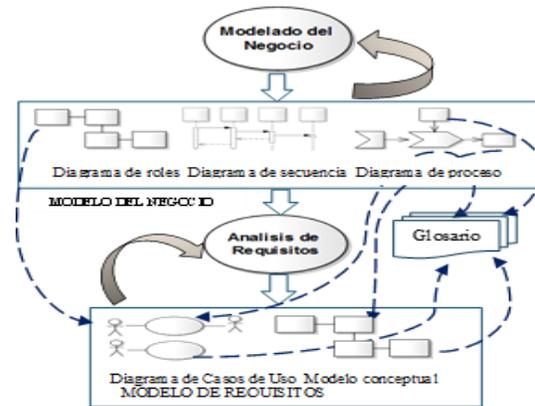


Fuente adaptado de [18]

Para cada uno de los subprocesos existen una serie de técnicas que pueden ser aplicadas, a continuación se describen algunas de ellas de acuerdo con Trujillo (2010) y Sommerville (2005).

- **Levantamiento de requisitos.** Las técnicas más utilizadas son las entrevistas formales e informales, encuestas o talleres de requisitos (lluvia de ideas, juego de roles, etc.) con los stakeholders⁴ del sistema.
- **Análisis de requisitos.** Contempla el modelado del negocio, el cual se puede realizar a través de diagramas de clases, procesos, roles o secuencias; a partir del modelo del negocio obtenido se pasa al análisis de requisitos en busca del modelo de requisitos, este se puede llevar a cabo utilizando diagramas de casos de uso, modelos conceptuales, entre otros. La figura 2 presenta un ejemplo de un diagrama de clases utilizado para elaborar un modelado del negocio y de las herramientas que se pueden utilizar para hacer esta actividad así como para realizar el análisis de requisitos.

Figura 2. Técnicas utilizadas en el subproceso de análisis de requisitos



Fuente. Adaptado de Anaya & Trujillo (2010)

- **Especificación de requisitos.** La técnica más utilizada es el documento de especificación de requisitos del sistema. Actualmente la mayoría de los sistemas de modelamiento con UML traen sus propias plantillas de especificación de requisitos.
- **Verificación de requisitos.** Se pueden utilizar los talleres de requisitos, las pruebas de requisitos (Examinación, Búsqueda de Factores) y la inspección de requisitos.
- **Trazabilidad.** Generalmente se utilizan los puntos de casos de uso basados en puntos de función. También se pueden utilizar estimaciones de esfuerzo y estimaciones de cronograma.
- **Control de cambios.** Se pueden manejar planillas de control de solicitudes (formalización y clasificación de la solicitud, estimación del impacto, evaluación y aprobación de la solicitud) y seguimientos al control de cambios (desarrollo, documentación, auditoria y liberación del cambio).

3.4. Modelado de Requisitos Con Diagramas de Casos de Uso

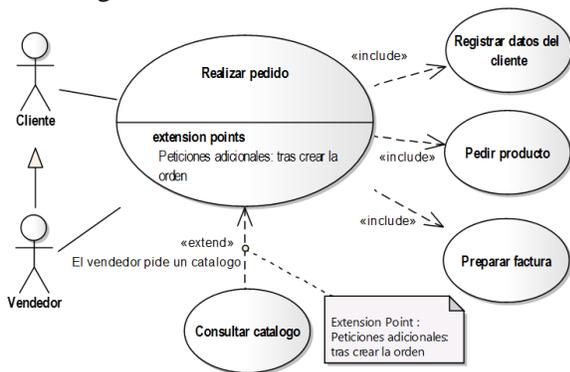
Los diagramas de casos de uso es una de las técnicas más utilizadas para el modelamiento de requisitos y está incorporada en el Lenguaje Unificado de Modelado [19]. Los diagramas de casos de uso permiten modelar la interacción entre los usuarios (actores) y los servicios ofrecidos por el sistema. Cada interacción puede

⁴ Persona que está directa o indirectamente relacionada con el sistema, y ésta puede ser parte de la organización, cliente o usuario final

ser representada a través de un único hilo o múltiples hilos, por lo cual se deberán identificar las interacciones normales y las interacciones alternativas.

Los diagramas de casos de uso, documentan el comportamiento de un sistema desde el punto de vista del usuario. Lugar donde se definen los actores, las interacciones y los casos de uso, los cuales se representan como se muestra en la figura 3 y se describe en la tabla 1.

Figura 3. Ejemplo diagrama de casos de uso y su simbología



Fuente. Elaboración propia

Tabla 1. Notaciones en diagramas de casos de uso.

Notación	Descripción
Actor	Usuario concreto que interactúa con el sistema
Caso de uso	Especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores.
Relaciones	
Asociación	Se encarga de comunicar un actor con un caso de uso.
Include	Relación entre dos casos de uso. Implica que un caso de uso base incorpora explícitamente el comportamiento de otro en algún lugar de su secuencia.
Extend	Relación entre dos casos de uso. Implica que un caso de uso base incorpora implícitamente el comportamiento de otro caso de uso en el lugar especificado indirectamente por este otro

	caso de uso.
Generalización (Relación de herencia)	Son aplicables a actores o casos de uso. Significa que un actor hijo hereda el comportamiento y significado del actor padre.

Fuente. Elaboración propia.

4. DISEÑO EXPERIMENTAL DEL JUEGO MODELA

MODELA es un juego serio donde los estudiantes, de Ingeniería del software, pueden aprender a modelar requerimientos con casos de uso.

4.1. Objetivo del Juego

Lograr que los estudiantes puedan realizar a través de requerimientos extraídos de un problema real diagramas UML de forma correcta y consistente, como: diagrama de casos de uso, de clases y de secuencias, correspondientes al modelo digital de un problema específico.

4.2. Hipótesis

Cuando un sujeto experimental se somete a la aplicación de “El juego Modela”, puede deducir algunos aspectos relativos a la ingeniería del software y puede clarificar los conceptos en relación a diagramas de casos de uso y realizar diagramas UML de forma correcta y consistente.

4.3. Sujetos Experimentales

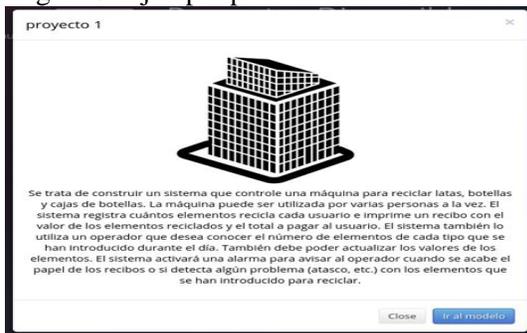
El juego va dirigido tanto a alumnos de ingeniería como a los profesores y al público en general que trabaje en el área de ingeniería. Permite a estos practicar el aprendizaje de cómo modelar requerimientos a través de artefactos UML.

4.5. Material Experimental

El juego está compuesto, básicamente, por el siguiente material:

- Modelo verbal: Es un discurso en un lenguaje controlado que representa el proyecto que se va a modelar y que se encuentra previamente registrado en la base de datos, en la plantilla proyecto, se muestra un ejemplo del manejo de una máquina de reciclar latas, botellas y cajas de botellas.

Figura 3. Ejemplo plantilla modelo verbal.



Fuente. Elaboración propia

Observación - Es importante destacar, que hay muchas maneras en las cuales se puede concebir el modelo verbal propuesto para la realización del diagrama que se solicita. Sin embargo, por efectos de las limitaciones de tiempo de juego, se decidió predefinir la solución. En este caso es importante la intervención del profesor para dar una evaluación final acerca del diagrama desarrollado.

- Escenario modelar Casos de uso: plantilla que contiene caja de diálogo con la simbología UML, una ficha de requerimientos predefinidos de acuerdo al modelo verbal seleccionado, adaptados de forma que se puedan colocar en la realización de los diagramas como se puede apreciar en la figura 4. La plantilla con los artefactos UML

Figura 4. Plantilla para realizar diagramas.



Fuente. Elaboración propia

4.6. Evaluación

4.6.1. Procedimiento Experimental

La evaluación del juego se llevó a cabo con 32 estudiantes. La realización de la prueba consistió en dividir el número de estudiantes en dos grupos de 16: un grupo que utilizó el juego y un segundo grupo que trabajó con la guía de un docente.

El procedimiento consistió en lo siguiente: Primero, se dio una breve explicación del contexto en el que iban a trabajar, tanto al grupo que iba a utilizar el juego como el que no; a continuación un grupo de estudiantes pasó a utilizar el juego: el profesor le asigna un problema para modelar un diagrama de casos de uso y el otro grupo pasa a resolver el mismo problema con la guía de un profesor. Al finalizar la prueba cada estudiante que uso el juego, mediante un formulario integrado en el juego pudo enviar su puntuación obtenida conjuntamente con una valoración personal sobre la experiencia. Por último, se entregó a los estudiantes que utilizaron el juego como a los que no un cuestionario, para evaluar la actividad realizada. En este se preguntó aspectos como: si creían haber aprendido algo, si les había resultado fácil llegar a la solución correcta, si para llegar a la solución habían reflexionado previamente o simplemente lo habían hecho por prueba y error, si la ayuda les había sido útil o les hubiera gustado otro tipo de ayuda para resolver el juego, y por último que describieran los aspectos positivos y negativos del juego.

Finalmente, a partir de las respuestas al cuestionario realizado, se tabularon los datos y se extrajo los resultados y las conclusiones pertinentes para el proyecto.

5. RESULTADOS DE LA APLICACIÓN DEL JUEGO

Para el análisis de la evaluación se ha tenido en cuenta las diferentes respuestas de los participantes en los cuestionarios.

Para el análisis de los resultados, se tabularon los datos cuantitativos y cualitativos (respuestas a preguntas abiertas) obtenidos de los diferentes

cuestionarios, estos sirvieron para inferir: si los estudiantes que utilizaron el juego dedujeron algunos aspectos relativos a la ingeniería del software, clarificaron los conceptos en relación a diagramas de casos de uso y pudieron realizar diagramas UML de forma correcta y consistente. A continuación se presentan los resultados en lo que tiene que ver con: la experiencia de aprendizaje, la ayuda utilizada en el juego como facilitadora para encontrar la solución, facilidad en la solución del problema, encontrar la solución por reflexión o prueba y error, tipo de ayuda (sugerida por los estudiantes), y aspectos positivos y negativos que los estudiantes percibieron del juego.

Experiencia de aprendizaje: En esta sección se valoran los resultados obtenidos a la pregunta: ¿El juego Modela para modelar requerimientos con casos de uso fue una experiencia de aprendizaje?, los resultados obtenidos han sido los siguientes: un 87% de los estudiantes con el juego creen haber aprendido conceptos nuevos durante el juego mientras que un 13% creen no haber aprendido ningún concepto nuevo. Por otra parte, un 94% de alumnos sin el juego creen haber aprendido conceptos nuevos mientras que un 6% cree que no ha aprendido nada nuevo. Como se aprecia los resultados, en ambos casos, son muy similares, con esto se puede afirmar que el juego es un complemento que sirve como apoyo al docente en la enseñanza de la ingeniería del software.

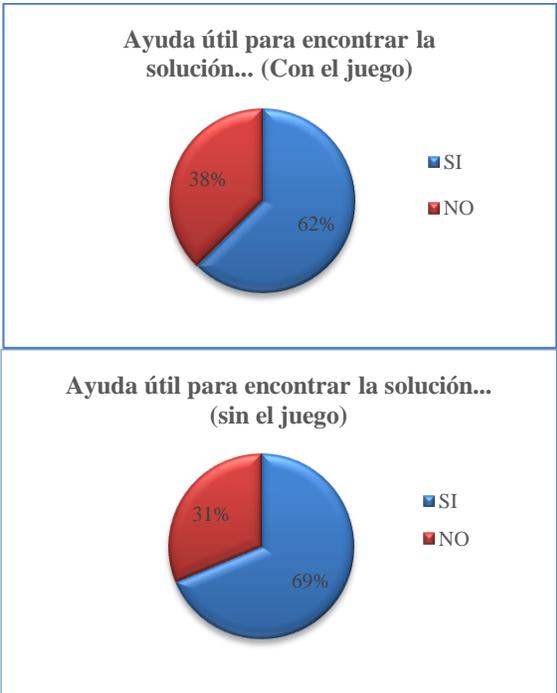
Figura 8. Resultados obtenidos en la pregunta número 1.



Fuente. Elaboración propia

La ayuda utilizada en el juego facilita encontrar la solución: En esta sección se valoran los resultados obtenidos a la pregunta: ¿El tipo de ayuda utilizada en el juego te sirvió durante el desarrollo de este para encontrar la solución correcta?, los resultados obtenidos han sido los siguientes: un 62% de los estudiantes afirman que la ayuda les sirvió, mientras que un 38% dicen que no. Por otra parte, un 69% de alumnos sin el juego afirman que la ayuda les sirvió, mientras que un 31% dice que no. Como se aprecia los resultados, en ambos casos, son muy similares, lo que corrobora el planteamiento de Wankat y Oreovicz [12]: la clase magistral como estrategia didáctica requiere ser complementada por otras estrategias para alcanzar objetivos cognitivos de más alto nivel

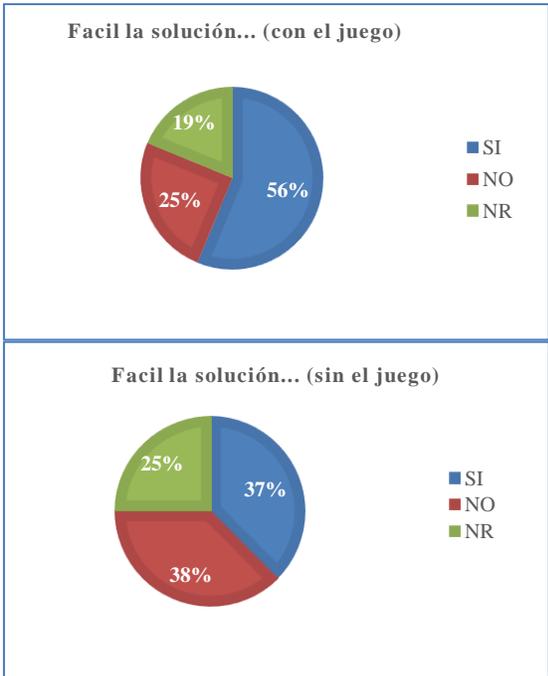
Figura 9. Resultados obtenidos en la pregunta número 5.



Fuente. Elaboración propia

Facilidad en la solución del problema: En este apartado se estudia los resultados obtenidos a la pregunta: ¿Te fue fácil, a través del juego, llegar a la solución correcta?

Figura 10. Resultados obtenidos en la número 6.



Fuente. Elaboración propia

Como se puede ver en la figura 10 un 56% de los alumnos que utilizó el juego le pareció fácil, mientras que un 25% le resultó difícil. El 19% no responde. Por otro lado, un 37% de los alumnos que no utilizaron el juego les pareció fácil, a diferencia de un 38% que le resulta difícil, y un 25% que no responde. De aquí se puede extraer que hay una ligera tendencia a que con el juego es más fácil realizar un diagrama de casos de uso que con la enseñanza tradicional. Los estudiantes argumentan que con el juego no hay presión en el proceso de aprendizaje.

Encontrar la solución por reflexión o prueba y error: En este apartado se estudia los resultados obtenidos a la pregunta que indaga si llegar a la solución correcta se hizo reflexionando previamente el tipo de elemento que iba a colocar o simplemente se llegó a la solución por prueba y error.

Figura 11. Resultados obtenidos en la pregunta número 7



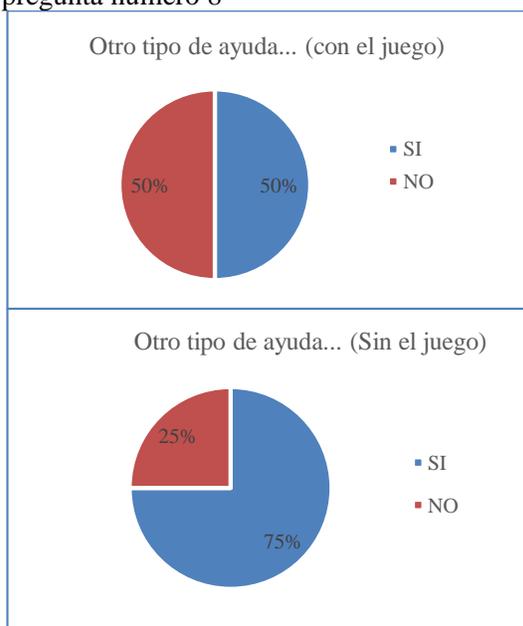
Fuente. Elaboración propia

Como se puede observar en la figura 11 un 75% de los alumnos que utilizaron el juego

llegaron a la solución a través de la reflexión y un 25% por prueba y error. Por otro lado, un 63% de los alumnos que no utilizaron el juego llegaron a la solución por prueba y error y un 37% por reflexión. De aquí se puede extraer que hay una ligera tendencia a reflexionar más en aquellos casos en los que se provee a los estudiantes con técnicas que guían en el proceso de aprendizaje.

Tipo de ayuda (sugerida por los estudiantes): En la Figura 12 se puede observar la respuesta de los alumnos a la pregunta: ¿Te hubiera gustado que el juego te proporcionara otro tipo de ayuda diferente?

Figura 12. Resultados obtenidos en la pregunta número 8



Fuente. Elaboración propia

De la figura 12 podemos extraer que 50% de los estudiantes que utilizaron el juego les hubiera gustado que este les proporcionara otro tipo de ayuda en tanto que el otro 50% dice que no. Por el contrario el 75% de los alumnos que no utilizaron el juego resaltan la necesidad de que hubiera sido útil algún tipo de ayuda.

Aspectos positivos y negativos: En general las opiniones respecto al juego son bastante positivas, como se puede observar en la tabla

3, los estudiantes valoran el juego en lo concerniente a su interfaz de usuario, la navegabilidad, los colores y su facilidad de uso.

Tabla 2. Aspectos positivos y negativos del juego.

Me gusto del juego	Nro	No me gusto del juego	Nro
Su facilidad de uso	14	Su facilidad de uso	2
Su interfaz de usuario	15	Su interfaz de usuario	1
Su animación	13	Su animación	3
El menú de navegabilidad	15	El menú de navegabilidad	1
El sistema de ayuda	9	El sistema de ayuda	7
Los colores usados en el juego	15	Los colores usados en el juego	0

Fuente. Elaboración propia

Con respecto a la metodología tradicional se puede notar que los estudiantes valoran el manejo y orden en el desarrollo de la clase, en tanto que no están a gusto con los ejemplos propuesto en clase, la forma como se responden las preguntas y como se utiliza el tablero, como se puede ver en la tabla 3.

Tabla 3. Aspectos positivos y negativos de la metodología tradicional.

Me gusto de la enseñanza tradicional	Nro	No me gusto de la enseñanza tradicional	Nro
Como explica los temas en clase	10	Como explica los temas en clase	5
Manejo del discurso en clase	11	Manejo del discurso en clase	0
La forma como utiliza el tablero	10	La forma como utiliza el tablero	6
Como responde las preguntas	10	Como responde las preguntas	6
Orden en el desarrollo de	12	Orden en el desarrollo de la	4

la clase		clase	
Los ejemplos propuestos en clase	9	Los ejemplos propuestos en clase	7

Fuente. Elaboración propia

6. CONCLUSIONES

Entre las principales conclusiones de este trabajo se cuentan las siguientes:

- Para la enseñanza de la Ingeniería del Software se han empleado diversas estrategias tradicionales, que poco se han complementado con otras experiencias que no se centren en el docente como responsable único de la enseñanza. Esto ha generado la búsqueda de alternativas de enseñanza como los juegos serios; es por ello que hoy en día, estos están tomando cada vez mayor fuerza en la educación.
- El Juego Modela es una herramienta didáctica para usar en las aulas de clase que le permite al estudiante afianzar conocimientos sobre modelado de requerimientos con casos de uso, métodos de desarrollo de software, trabajo en equipo, comunicación y, sobre todo, análisis en la concepción del problema propuesto.
- Es de anotar que tanto los estudiantes que utilizaron el juego como los que no valoran las ayudas que se les puedan proporcionar para adquirir un buen aprendizaje en el modelado de diagramas de casos de uso.
- Los resultados de la experiencia muestran que los estudiantes, en mayor porcentaje, manifiestan que les resulta más fácil desarrollar un modelo de caso de uso correcto utilizando un juego serio, en este caso el juego MODELA, que con la enseñanza tradicional. Esto corrobora la utilización de los juegos serios como herramienta didáctica en el proceso enseñanza aprendizaje.
- La experiencia también muestra que los estudiantes, específicamente los que no utilizaron el juego, en un alto porcentaje consideran la importancia de utilizar una

metodología de enseñanza diferente para modelar diagramas de casos de uso, que según los resultados tendría que basarse en la filosofía de las metodologías activas para el proceso enseñanza-aprendizaje

- Finalmente, los resultados muestran, que los juegos en clase no reemplazan la enseñanza tradicional de la Ingeniería del Software, más bien la complementan. Los conocimientos previos en relación con los temas que se abordan en el juego ha demostrado ser fundamental para las conclusiones a que llegan sus participantes. Sin embargo, algunos conceptos del manejo de proyectos, tales como el trabajo en equipo y la importancia de la comunicación entre los integrantes del grupo son difíciles de enseñar mediante métodos tradicionales; la experimentación de situaciones como las que aborda el juego permite que esos conceptos se refuercen, de forma que se puedan incorporar en el aprendizaje significativo de los estudiantes.

7. RECOMENDACIONES Y TRABAJO FUTURO

Con base en esta experiencia, los puntos a tener en cuenta en el futuro son:

- Modelar otros artefactos UML como diagrama de secuencia, de clase, de estado, de actividad, entre otros, en el juego MODELA.
- Aplicación de “El Juego Modela” en grupos de personas con diferente perfil. Sería especialmente importante conocer las reacciones frente al juego de profesionales de la Ingeniería del Software y de Sistemas en general. También se podría utilizar el juego con personas con pocos conocimientos en modelado de requerimientos con artefactos UML.
- Elaboración de otros juegos que permitan mejorar o complementar “El Juego Modela”, u otros temas como: Aspectos, Puntos de Vista o Refinamiento, de gran actualidad en la Ingeniería del Software.

- Realizar la experiencia con profesores para determinar la adaptación del producto por parte del profesorado ya que en este TFM, (Trabajo final de Máster), únicamente se centró en las opiniones de los estudiantes.
- Como es conocido en este juego se ha utilizado la técnica de pistas textuales. Otra alternativa de trabajo futuro sería la de utilizar otro tipo de pistas visuales, técnica de scaffolding basada en pistas, preguntas, etc.
- Respecto al juego realizado en el TFM, se deberían pulir algunos aspectos, como mejorar la interfaz y programación para que sea más atractivo para el usuario.

REFERENCIAS

- [1] Pressman R.: “Ingeniería del Software: un Enfoque Práctico”, McGraw-Hill, Madrid, 2002.
- [2] Bauer, F.L.: “Software Engineering”. In: Proceedings of the IFIP Congress 71, ed. Friedman, C.V., North Holland (1972), 530–538.
- [3] Rumbaugh J., Blaha M., Premerlani W., Eddy F., and Lorenzen W.: “Object-Oriented Modeling and design”, Prentice Hall, New Jersey. 1991
- [4] OMG. OMG Unified Modeling Language Specification. Object Management Group. Disponible: <http://www.omg.org/UML/>. [Citado 30 de Septiembre de 2014]
- [5] Zowghi D. and Gervasi V.: “The Three Cs of Requirements: Consistency, Completeness and Correctness”. In: Proceedings of 8th International Workshop on Requirements Engineering: Foundation for Software Quality, (REFSQ’02), Essen, Germany (2002).
- [6] Zapata C. M. y Awad G.: “El Juego de los Requisitos: Enseñanza de la Gestión de Proyectos de Software”. En: Memorias del XIII Congreso Iberoamericano de Educación Superior en Computación CIESC2005, Cali (2005), 33-43.
- [7] Zambrano A., W. R. (2011). Modelo de enseñanza-aprendizaje para la educación superior basado en redes sociales. Didáctica, Revista de Investigación, págs. 26-48.
- [8] Wankat, P.C. y Oreovicz, F.S.: “Teaching Engineering”, McGraw-Hill, Nueva York, 1993.
- [9] Rugarcia, A., Felder, R., Woods, D. y Stice, J.: “The Future of Engineering Education I: The vision for a new century”. Chemical Engineering Education, Vol. 34, No. 1, (2000), 16-25.
- [10] Kober R., and Tarca A.: “For fun or profit? An evaluation of a business simulation game”. Accounting Research Journal, Vol. 15 (2000) 98-111.
- [11] Klassen K. y Willoughby K.: “In-Class Simulation Games: Assessing Student Learning”. Journal of Information Technology Education, Vol. 2, (2003), 1-13.
- [12] Strategy Dynamics: “Beefeater restaurants microworld”. Available <http://www.strategydynamics.com/products/beefspec1.asp> [Citado 23 de Mayo. de 2014]
- [13] Baker A., Navarro E., and Van der Hoek A.: “An experimental card game for teaching software engineering processes”. The Journal of Systems and Software, No. 75, (2005), 3–16.
- [14] Zapata C. M. y Awad G.: “El Juego de los Requisitos: Enseñanza de la Gestión de Proyectos de Software”. En: Memorias del XIII Congreso Iberoamericano de Educación Superior en Computación CIESC2005, Cali (2005), 33-43.
- [15] Zapata Jaramillo, Carlos Mario (2007). “*Los juegos de clase no tecnológicos como una estrategia didáctica para la enseñanza de*

la ingeniería de software". Recuperado el 29 de octubre de 2014, de <http://www.bdigital.unal.edu.co/1621/#sthash.kIZe0r6d.dpuf>

- [16] Booch G.: "El proceso Unificado de Desarrollo de Software", Addison Wesley, Madrid, 2001.
- [17] Aurum, Aybüke; Wohlin, Claes. "*Engineering and Managing Software Requirements*". Editorial Springer. 2005.
- [18] Wiegers, Karl. "*Software Requirements*". Segunda edición. Microsoft Press. 2003.
- [19] Jacobson, I., Booch, G., & Rumbaugh, J. (2006). El Lenguaje Unificado de Modelado. México: 2da Edición. Pearson Addison-Wesley.