

Evolución de un Juego Educativo de Ingeniería de Software a través de Técnicas de Elicitación de Requisitos

Elizabeth Suescún Monsalve¹, Vera Maria B. Werneck², Julio Cesar Sampaio do Prado Leite¹

¹Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brasil

²Universidade Estadual do Rio de Janeiro (UERJ), Rio de Janeiro, Brasil

emonsalve@inf.puc-rio.br, vera@ime.uerj.br, www.inf.puc-rio.br/~julio

Resumen

La ingeniería de requisitos tiene como tarea principal representar las necesidades de un contexto estudiado. El entendimiento y el uso de técnicas apropiadas para representar estas necesidades pueden asegurar la satisfacción de los interesados. Para ello la ingeniería de requisitos se vale de diferentes estrategias para la captura de esta información. Basados en estos conceptos queremos mostrar cómo la elección y la aplicación oportuna de estas estrategias pueden llevar a la evolución y mejora de los sistemas. Nuestro objetivo es mostrar como a través de la aplicación de técnicas de elicitación de requisitos fue concebida la evolución de un juego de cartas educativo. Este juego contaba con un modelo situacional como fuente de información y derivó para un modelo intencional, este último como parte de la representación evolucionada del juego.

Abstract

The principal task of requirements engineering is the definition of requirements that reflects the understanding of the problem. In this process different approaches for information elicitation may be used. In our case we show that the application of elicitation strategies in the right form can improve the evolution of an existing artifact. In this paper we show the evolution of an education game.

The principal task of requirements engineering is the definition of requirements that reflects the understanding of the problem. In this process different approaches for information elicitation may be used. In our case we show that the application of elicitation strategies in the right form can improve the evolution of an existing artifact. In this paper we show the evolution of an education game. The modeling of the

game evolved as well, we moved from a situational model to an intentional model.

1. Introducción

La ingeniería de requisitos puede considerarse como un proceso de descubrimiento y comunicación de las necesidades del usuario y la gestión de los cambios o evoluciones que surjan de dichas necesidades [15]. Es en este contexto que queremos presentar la evolución del juego SimulES [1], un juego de cartas educativo que simula el proceso de desarrollo de software.

Para su evolución dos técnicas de elicitación, *observación* y *cuestionario* fueron utilizadas permitiéndonos identificar y comprender las necesidades de los usuarios. Para el cumplimiento de éste objetivo fue realizada una actividad en la UERJ (*Universidade do Estado do Rio de Janeiro*) donde las dos técnicas fueron aplicadas, y como resultado fue evaluada la aceptación del juego por parte de los participantes, así como también las oportunidades de mejoras e implementación del juego.

Como parte de este proceso nos contextualizaremos en juegos educativos, observando los objetivos de cada uno y la forma como fueron concebidos, resaltaremos como ellos se están haciendo importantes como una tendencia que permite que la experiencia educativa sea cada vez más diversificada, principalmente en aquellos casos donde están involucradas diferentes tareas o procesos.

Dentro de los métodos tradicionales de enseñanza tenemos las clases magistrales y trabajos prácticos. Existen otros métodos menos explorados como son los juegos, que pueden llegar a complementar la enseñanza tradicional. Explorar métodos alternativos de aprendizaje se hace necesario para que, con otros enfoques, los estudiantes asimilen el conocimiento que les permita desarrollar sus competencias.

Entendiendo que estas metodologías de enseñanza deben ser cada día mejoradas con la ayuda y retroalimentación de los involucrados en el proceso de aprendizaje (profesores y alumnos), vislumbramos la evolución de SimulES utilizando técnicas que ayuden en la obtención de la información, la cual permitirá proponer elementos de perfeccionamiento.

El artículo está organizando en 6 Secciones: en la Sección 2 presentaremos algunos juegos educativos empleados para la enseñanza en la ingeniería de software. En la Sección 3 presentamos el juego SimulES. En la Sección 4 mostraremos las técnicas de elicitación utilizadas en este trabajo y como fueron aplicadas. En la Sección 5 se discuten los aspectos generales sobre la evolución de SimulES. Finalmente en la Sección 6 se presentan las conclusiones y trabajos futuros.

2. Enseñando 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 percibimos 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 se ha utilizado con frecuencia el desarrollo de pequeños proyectos [4], [23], [24], [25], pero aún así no simulan situaciones dadas en sistemas más grandes y complejos. De esta manera, los juegos educativos pueden ser utilizados de forma eficiente para suplir o simular estos procesos reales.

Recientemente, juegos (de tablero, cartas y computador), simuladores, entre otros, están siendo explorados para apoyar la enseñanza en diferentes áreas del conocimiento [8],[9],[10], así como también la ingeniería de software se ha sumado en la exploración de esta alternativa [5],[6],[7]. En este artículo concentraremos nuestra contribución en el juego educativo para la enseñanza de la ingeniería de software SimulES, centrados en la perspectiva de su evolución, identificada y representada, utilizando técnicas de elicitación de requisitos.

Como ya fue mencionado, los juegos pueden llegar a complementar la metodología de enseñanza tradicional en muchas áreas. La tecnología de juegos es utilizada como una herramienta que complementa las aulas tradicionales, pero aún es una práctica poco común en la enseñanza de ingeniería de software.

Fueron explorados algunos simuladores para la enseñanza en ingeniería de software tales como: SESAM, [12], SimVBSE [13], SIMSE[14]. A continuación describiremos brevemente cada uno de estos.

SESAM (Software Engineering Simulation by Animated Models) [12], está enfocado a la enseñanza de la gestión de proyectos a través de simulación. Su objetivo es demostrar cómo los recursos utilizados en un proyecto pueden influenciar resultados globales y cómo un proyecto es impactado como consecuencia de los cambios de procesos y recursos.

SimVBSE [13] está enfocado a la enseñanza de un nuevo principio en ingeniería de software llamado “*ingeniería de software basado en valor*”, donde los interesados de un sistema son identificados y también su factor crítico de éxito y su valor de preferencia, o sea, la simulación se basa en identificar qué es lo que los interesados consideran como factor crítico de éxito y determinar una estrategia que permita equilibrar estos valores definidos.

SIMSE[14] es un software educativo interactivo para un solo jugador, que simula un proceso de ingeniería de software. Guía a los estudiantes por los diferentes procesos de software, en los cuales los estudiantes tienen que lidiar con presupuesto, tiempo de proyecto y con dificultades que se presentan al ir ejecutando la simulación. Para ello los estudiantes deben tomar decisiones que pueden afectar positivamente o negativamente el proyecto. La idea es terminar el proyecto en el tiempo y el presupuesto estipulado.

Estos juegos presentan similitudes con SimulES ya que tienen como origen el mismo juego, “*Problems and Programmers*” (PnP) [2]. Ellos fueron modelados así: el primero no tiene una metodología concreta pero se tienen documentos de especificación, diseño y arquitectura creados en las diferentes evoluciones. El segundo fue modelado basado en prototipos y el tercero y último fue modelado usando diferentes métodos, cada uno dependiendo de la versión del juego.

3. SimulES

SimulES fue concebido a partir de la evolución del juego “*Problems and Programmers*” (PnP) [2], que también es la base de los juegos mencionados en la sección anterior. SimulES adiciona conceptos de evolución de software e interacción entre jugadores al ser un juego multiusuario. En SimulES el estudiante asume el rol de gerente de proyecto y debe lidiar con el presupuesto del proyecto, contratar ingenieros,

despedir ingenieros, construir los diferentes artefactos requeridos para el proyecto, entre muchas otras actividades. El jugador debe crear una estrategia que le permita mejorar su juego, debe realizar jugadas que desestabilicen los juegos de sus adversarios, debe defenderse utilizando conceptos de ingeniería de software y debe bloquear el ataque del adversario si posee cartas que se lo permitan. El jugador que primero construya todos los módulos con los artefactos requeridos para finalizar el proyecto y, si durante el mismo, el jugador realizó acciones de inspección y no posee ningún error, ganará el juego. Si no ha hecho inspección puede solicitarla y si no son encontrados errores en los artefactos construidos podrá ganar el juego también.

En la Figura 1 podemos ver el proceso que es seguido durante el juego. En *jugada de inicio* se elige el proyecto que deberá ser construido durante el juego, en *jugada de acciones* deben ser tratados los artefactos propios del producto software, en *jugada de conceptos* y *tratamiento de problemas* serán usadas cartas relacionadas con conceptos y problemas típicos en la ingeniería de software. El jugador podrá utilizar *cartas concepto* en su beneficio y utilizar *cartas problemas* como una forma de atacar a los adversarios, y por último, tenemos *enviar producto*. El primer jugador que llegue a esta instancia sin tener errores en su producto de software y no tenga ninguna penalidad pendiente ganará el juego.

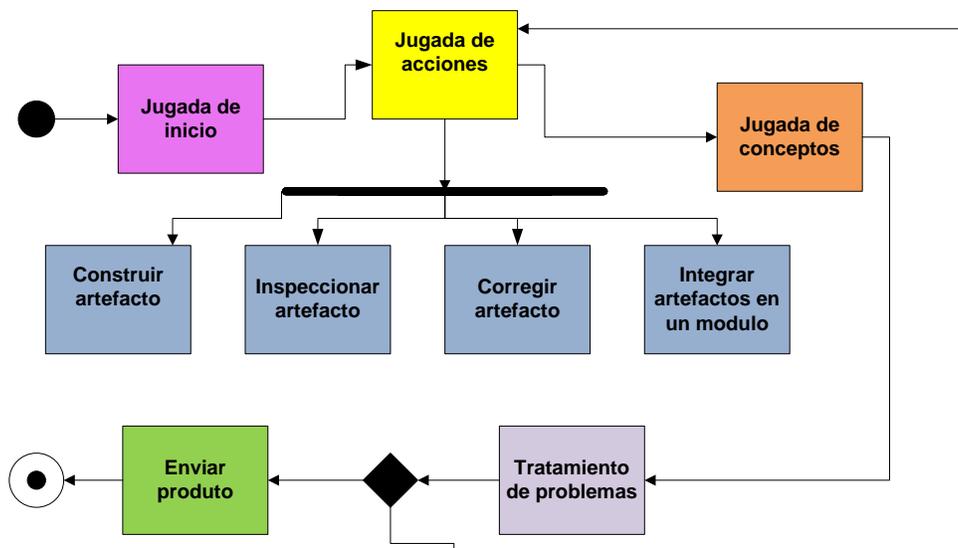


Figura 1. Dinámica de Simules

En la Figura 2 se ilustra el tablero individual que es gestionado por cada uno de los jugadores durante el juego. En él son ejecutadas las acciones de *contratar ingeniero* y *despedir ingeniero*, además las acciones de *construir*, *inspeccionar*, *corregir* y *enviar producto*. Algunas penalidades impuestas por otros jugadores también pueden verse reflejadas en el tablero individual.

En la Figura 3 presentamos los elementos del tablero principal, el cual es visible para todos los jugadores y donde están disponibles todos los recursos del juego. De la pila *cartones de proyectos* es escogido aleatoriamente el proyecto a ser tratado durante el juego, de la pila *Cartones de ingenieros de software* los jugadores van escogiendo los ingenieros que luego serán colocados en el tablero individual cuando el jugador ejecute la acción de *contratar ingeniero*, las

cartas blancas y *cartas grises* son usadas para la construcción de los diferentes artefactos presentes en el tablero individual, la cantidad de artefactos esta especificada en el *cartón del proyecto* en el ítem *módulos*. Es importante anotar que estos elementos (*cartas blancas* y *cartas grises*) pueden contener errores o no, esto es conocido en el momento de hacer *inspección* o cuando el jugador ejecuta *enviar el producto*.

4. Técnicas de Elicitación Utilizadas

La ingeniería de requisitos es un proceso sistemático de desarrollar requisitos mediante un proceso iterativo y cooperativo de analizar el problema, documentar las observaciones resultantes en varios formatos de representación y comprobar la precisión del

conocimiento obtenido [17]. Las estrategias escogidas para encontrar elementos de evolución en el juego SimuleS fueron: Observación y Cuestionario.

Tablero Individual				
	Ingeniero 1	Ingeniero 2	Ingeniero 3	
Requisitos	Carta blanca	Carta gris		
Dinero		Carta gris		
Trazabilidad	Carta blanca			
Cargo		Carta gris		
Ayuda				

Figura 2. Tablero individual de SimuleS

Figura 3. Elementos del tablero principal de SimuleS

La *Observación* es una técnica ya que permite que el ingeniero de requisitos tenga una posición pasiva del contexto observado [16]. O sea, el ingeniero mira lo que pasa en el ambiente y hace anotaciones acerca de lo que observa. En nuestro caso el foco de observación fue la práctica del juego SimuleS. Además, esta técnica posibilita que el ingeniero tenga un contacto suficientemente cercano con el fenómeno que está investigando. En nuestra experiencia con SimuleS y los estudiantes, la técnica ayudó a identificar elementos importantes de evolución y a validar y refinar el vocabulario del juego.

Los *Cuestionarios* son otra técnica que permite tener una idea bien definida de ciertos aspectos del sistema o del dominio, permiten abarcar un mayor número de personas, además de permitir análisis de estadísticas [16]. Los cuestionarios son utilizados para evaluar sistemas de una forma eficiente, tal es el caso de cuestionarios para evaluar sitios web [19] donde una serie de preguntas permite valorar qué tan transparentes resultan ser estos sitios para los usuarios.

Conforme a lo mencionado en [18] los cuestionarios son usados de forma útil cuando es difícil planear reuniones iniciales o entrevistas, las cuales son necesarias para hacer captura de requisitos. Nuestra situación era similar ya que solamente contábamos para realizar la actividad con la programación de la clase.

Estas dos técnicas fueron escogidas para identificar y definir elementos de evolución en SimuleS, permitiéndonos un mejor entendimiento y comunicación con los alumnos involucrados en la actividad, además nos llevo a entender y profundizar en el dominio analizado.

4.1. Utilizando la Técnica de Cuestionarios

Se consideró adecuado realizar un laboratorio didáctico-pedagógico para los alumnos, sobre todo para aquellos que están iniciando estudios en el área de la informática.

Esta actividad tenía básicamente dos objetivos: el primero era encontrar elementos de evolución en SimuleS, el cual ya ha sido mencionado en este trabajo, y el segundo era identificar la aceptación del juego por parte de los estudiantes.

Se escogió el grupo de Geomática¹ de la UERJ que cuenta con estudiantes de áreas como la estadística, cartografía, biología, matemática y alumnos de ciencias de la computación. La idea de esta experiencia era transmitir a los estudiantes conceptos de ingeniería de software a través del juego SimuleS e identificar fortalezas y debilidades del mismo.

En un formato impreso fue distribuida la documentación explicativa del juego, principalmente documentación en forma de escenarios, los cuales son descripciones de situaciones en lenguaje natural semi-estructurado que permiten entender, unificar, analizar y rastrear relaciones de un sistema, y también permiten describir interacciones entre sus componentes [11].

Título	Inicio de juego
Objetivo	Describir los preparativos para inicio del juego.
Contexto	<ul style="list-style-type: none"> - Tablero individual colocado en la mesa de juego - Cartas barajadas sobre la mesa - Cada jugador debe tener una carta de ingeniero de software
Actores	Jugador
Recursos	Dado, cartas, informaciones del proyecto, tablero individual, tablero principal
excepción	
Episódios	<ul style="list-style-type: none"> - Jugador lanza el dado, restricción: jugador que al tirar obtiene el mayor numero, inicia el juego. - Jugador escoge aleatoriamente de la pila de cartones de proyecto. Restricción: Los demás jugadores deben estar de acuerdo con la carta de proyecto escogida por el jugador a la vez este comienza el juego. - Cada jugador compra una carta de ingeniero de software y la coloca en su tablero. Restricción: el juego debe ir en sentido horario.

Figura 4. Escenario *Inicio de Juego* en SimuleS

¹ <http://www.geomatica.eng.uerj.br/>

La Figura 4 ilustra el escenario *inicio de juego*. La ejecución de este escenario comienza el juego, recordando lo mostrado en la Figura 1. Como podemos ver todos los detalles que deben ser llevados a cabo para el cumplimiento de esta jugada están descritos, como también cada una de las restricciones.

Los demás escenarios que se muestran en la Figura 1 también se encuentran detallados de esta forma, su especificación fue hecha por un grupo de estudiantes de la PUC-Rio como parte de la materia *evolución de software* [12] y fueron parte del material didáctico que utilizamos en nuestra actividad.

Otros recursos utilizados fueron *tablero general*, *tablero individual*, *cartas de proyecto*, *cartas blancas*, *cartas grises*, *cartones de ingenieros de software* y un *dado*, información que también se encuentra disponible en [13].

Hicimos un entrenamiento básico al distribuir todos los recursos y dar las explicaciones generales, luego nos dispusimos a jugar. Los estudiantes fueron guiados a través de las diferentes etapas del proceso de desarrollo de software para la creación de artefactos de requisitos, diseño, código, trazabilidad y ayudas, esto para la construcción del producto de software, principal objetivo del juego. Durante la construcción de los diferentes artefactos los jugadores fueron atacados con *cartas de problemas*, las cuales son una recopilación de problemas típicos de la ingeniería de software y son usadas para dañar el juego de los adversarios. Para defenderse de esta acción los jugadores utilizaron *cartas concepto*, permitiendo poner a prueba conocimientos relacionados con la ingeniería de software de los jugadores, con ello fueron neutralizados los problemas o penalidades.

Al terminar la actividad se realizó un cuestionario. Esto permitió recibir retroalimentación por parte de los estudiantes, lo que derivó en la evolución del juego SimulES.

La tabla 1 nos ilustra el cuestionario usado y su contenido. Con la selección de estas preguntas queríamos que por un lado el estudiante expresara abiertamente su opinión o impresión con respecto al juego y las sugerencias para mejorarlo. Y por otro lado queríamos identificar cuestiones concretas del juego, como se ilustra en la Figura 5.

El objetivo específico de cada una de las preguntas fue el siguiente:

Con la primera pregunta queríamos identificar la formación académica de los jugadores, ya que el juego involucra conocimientos y procesos de ingeniería de software, conceptos que quizás no eran familiares para algunos de los participantes.

La segunda pregunta nos permitió conocer si los jugadores involucrados en la actividad, aunque no tuvieran formación en informática, trabajaban o participaban en áreas afines a esta.

Con la tercera pregunta queríamos identificar si tanto la actividad como el juego despertaron interés entre los jugadores y si las expectativas que ellos se habían formado frente al juego habían sido cumplidas, pudiendo concluir que al ser aceptado podría generar beneficio entre los participantes.

Tabla 1. Cuestionario sobre SimulES aplicado a los estudiantes de Geomática de la UERJ

Cuestionario SimulES (Simulador de Ingeniería de Software)	
1.	Formación académica: <input type="radio"/> Pregrado en: <input type="radio"/> Posgrado en: <input type="radio"/> Otros
2.	Participa en áreas relacionadas con la ingeniería de Software o desarrollo de software? <input type="radio"/> Si <input type="radio"/> No <input type="radio"/> Cuales
3.	El juego SimulES cumplió con las expectativas? <input type="radio"/> Si <input type="radio"/> No <input type="radio"/> Cuales
4.	El juego SimulES aportó algún conocimiento nuevo? <input type="radio"/> Si <input type="radio"/> No <input type="radio"/> Cuales
5.	Puntos fuertes del Juego
6.	Puntos débiles de Juego
7.	Sugerencias de mejora
8.	Calificación del juego (de 0 a 5) <input type="radio"/> 5 optimo <input type="radio"/> 4 bueno <input type="radio"/> 3 débil <input type="radio"/> 1 malo <input type="radio"/> 0 sin opinión

Con la cuarta pregunta nuestro objetivo era identificar si un enfoque práctico e interactivo realmente podría apoyar la enseñanza tradicional, esto con respecto al aporte a cada jugador al interactuar con el juego.

En la quinta pregunta queríamos conocer por parte de los estudiantes o jugadores qué fortalezas identificaban dentro del juego, pregunta que complementaba las expectativas de los estudiantes frente al juego en la pregunta 4.

En la sexta pregunta los estudiantes nos reportarían aquello que ellos identificaban como debilidades del juego, y así, descubrir, proponer, plantear mejoras.

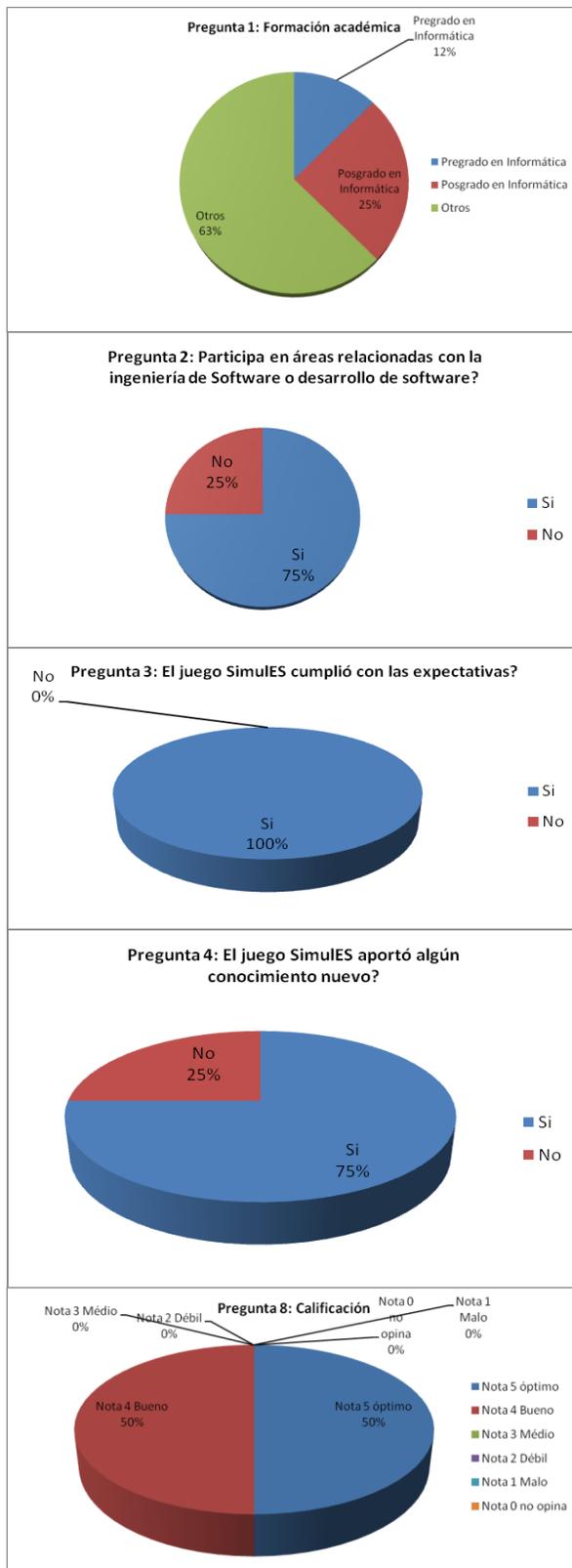


Figura 5. Representación gráfica de los resultados de las preguntas cerradas

En la séptima pregunta queríamos conocer desde la perspectiva de los estudiantes aquello que en su opinión consideraban como elementos a ser mejorados dentro del juego, como también en las actividades que se desarrollan en torno a este. Esta pregunta es un complemento a la pregunta sobre puntos débiles (pregunta 6). Estas dos preguntas fueron claves para identificar elementos de evolución del juego.

Y por último, la pregunta 8 sería un indicador del valor que los estudiantes darían al juego. O sea, su apreciación o juicio nos indicaría el impacto en los estudiantes.

4.2. Resultados Obtenidos de la Aplicación del Cuestionario

A continuación presentaremos una visión de las preguntas objetivas y el análisis general de las respuestas suministradas por los participantes a través del cuestionario.

La primera pregunta sobre la formación académica de los participantes nos dio un indicio de qué tan familiarizados estaban ellos con los conceptos de ingeniería de software. Al respecto pudimos concluir que la mayoría de los estudiantes que participaron de la experiencia no tenían una **formación** en áreas de la informática, pero al ser estudiantes de Geomática generalmente sí aplican sus conceptos a través de procesamiento de imágenes, banco de datos, registros técnicos, sistemas de información geográfica, mapeo digital y/o sistemas de posicionamiento.

Del resultado de la pregunta dos, que es un complemento de la primera pregunta, pudimos deducir que el gran porcentaje de los estudiantes **participan o trabajan** en áreas relacionadas con la informática. Los estudiantes manifestaron que el concepto de ingeniería de software y gestión de proyectos eran nuevos para ellos y que dentro de las áreas en las cuales actúan estos conceptos no habían sido aplicados. Concluimos que la información suministrada fue importante como parte integral de su formación.

Con la pregunta 3 determinamos que hubo una buena aceptación del juego SimulES por parte de los estudiantes. Ellos aprobaron la actividad por ser tanto entretenida como proveedora de información de interés. El resultado de esta pregunta es alentadora para el proyecto de SimulES porque en la medida de que éste sea aceptado, aún con fallas, nos motiva para continuar con su mejora y evolución.

Con la pregunta 4 la mayoría de los alumnos (75%) anotaron que el juego aportó nuevos conocimientos de ingeniería de software. Con esta actividad los

estudiantes quedaron muy motivados por el tema, ya que esta práctica podía ser útil en su futuro profesional.

En la evaluación final del juego SimulES todos los alumnos encontraron que el juego era bueno, siendo que la mitad asignó la nota máxima, considerando el juego óptimo. El juicio de los estudiantes nos da un indicio de la importancia y aceptación que pueden tener los juegos educativos en la enseñanza.

Los puntos positivos estuvieron entre las apreciaciones más destacadas suministradas por los estudiantes, y que de hecho, el juego estimula la habilidad para el desarrollo de artefactos, atendiendo siempre criterios de requisitos de proyecto, supervisión y pruebas (*test*). Éste también incentiva la interacción entre los jugadores y el trabajo en equipo, generando una competencia saludable.

Los estudiantes reportaron que el juego fue confuso en su inicio, solo después de la segunda jugada lograron el entendimiento de su dinámica, también reportaron que la organización de la actividad debería tener una mejor planeación, suministrando información anticipada sobre el juego. Esta es una oportunidad de mejora identificada que permite que los estudiantes sean más proactivos y ayuda para que el entrenamiento previo sea minimizado.

4.3. Utilizando la Técnica de Observación

La técnica de observación nos ayudó a descubrir y asimilar información sobre el dominio del juego en su ambiente natural, además de visualizar cómo los participantes se sumergían en la actividad al interactuar con el juego y sus adversarios.

A continuación detallaremos los aspectos que fueron identificados utilizando ésta técnica. En el caso de la aplicación que aquí describimos las dos primeras coautoras desempeñaron el rol de observadoras.

La observación fue realizada durante el juego y la información organizada posteriormente.

Interacción de los estudiantes con el juego:

- *Observado:* los estudiantes estuvieron muy atentos a las instrucciones iniciales del juego, fue distribuida una información impresa con instrucciones y detalles del juego que no fue lo suficientemente atractiva.

- *Aspecto a mejorar:* las personas participantes de la actividad deben recibir algún tipo de información previa, el material impreso a ser distribuido debe ser de fácil lectura y lo suficientemente atractivo para captar la atención de los estudiantes. Al ser distribuido

material con anterioridad a los involucrados, permitirá que lleguen a la actividad con un conocimiento básico sobre lo que será realizado y el entrenamiento será más fluido.

Los conceptos que contiene el juego:

- *Observado:* los estudiantes fueron guiados durante toda la actividad, el entendimiento general del juego vino después de haber ejecutado la primera ronda de jugadas. En la ejecución de la *jugada de conceptos y problemas* los estudiantes se sintieron inseguros de realizar sus jugadas, sobre todo aquellos que tenían poco conocimiento de los conceptos de ingeniería de software.

- *Aspecto a mejorar:* identificamos que un instructor es importante para que guíe a los estudiantes en esta actividad y refuerce los conceptos que se quieren transmitir.

Tipificación de cartas problema y concepto:

- *Observado:* se identificó que *las cartas problemas y cartas concepto* pueden ser clasificadas según el tipo de alumno que estará involucrado en la actividad.

- *Aspecto a mejorar:* las cartas pueden ser tipificadas para reforzar un aspecto en especial como *pruebas, desarrollo de software, ingeniería de software etc.*, dependiendo de las necesidades del grupo en el cual se aplica la actividad.

Contenido de las cartas problema y concepto:

- *Observado:* se identificó que *las cartas problemas y cartas concepto* tienen descripciones que pueden ser consideradas muy técnicas lo que dificultó el entendimiento de las mismas.

- *Aspecto a mejorar:* mecanismos de evaluación deben ser utilizados para determinar la claridad, entendimiento y legibilidad de las cartas. Conceptos de transparencia pueden ser útiles para esta evaluación. Según [22] dentro de las características de la transparencia encontramos que la calidad del contenido y entendimiento son algunos de sus atributos, lo que puede ser útil para la valoración de las cartas y así buscar un refinamiento de las mismas.

Secuencia y relación entre las jugadas de SimulES:

- *Observado:* la jugada de conceptos está compuesta por dos actividades principales. En la primera está involucrado el tablero individual y los artefactos, allí se puede *construir artefacto, inspeccionar artefacto, corregir artefacto y/o integrar artefactos en un módulo*. En la segunda actividad se hace *lanzamiento de dado* para la compra de *ingenieros de software*, compra de *cartas conceptos y cartas problemas*, todo esto dentro de una misma jugada. Como son dos actividades que no tienen una relación aparente, generó confusión entre los jugadores.

- *Aspecto a mejorar:* debe ser planteada una desagregación mayor de los escenarios, ya que existen casos en los cuales se tienen actividades que no están relacionadas entre sí. Como en este caso en particular, es posible plantear un sub-escenario para lanzamiento del dado y sus diferentes acciones y otro para las demás actividades o acciones sobre el tablero individual, indicar a los jugadores de la existencia de estos dos sub-escenarios mejoraría el entendimiento por parte de ellos.

Valores y reglas dentro del juego:

- *Observado:* la complejidad del proyecto se relaciona directamente con la habilidad del ingeniero de software para la construcción de artefactos dentro del juego, por ejemplo, si la complejidad del proyecto es de 2 esto indica que cada carta blanca cuesta 2 “puntos de tiempo” y consecuentemente cada carta gris cuesta 1 punto, entonces, si el ingeniero de software tiene una habilidad de 4 solo podrá obtener por jugada 2 cartas blancas o 4 cartas grises, este cálculo debe ser hecho con cada una de los ingenieros de software que posee el jugador, cálculo que en algunas ocasiones involucra penalidades impuestas al jugador y que puede afectar la habilidad del ingeniero.

Se identificó que cada vez que los jugadores ejecutaban el escenario *jugada de acciones* debían realizar mentalmente este cálculo, lo que generó demoras en el juego y confusión entre los jugadores.

- *Aspecto a mejorar:* durante el entrenamiento de los jugadores las explicaciones sobre cálculos y validaciones deben ser claras y dentro del material de apoyo suministrado deben ser incluidos ejemplos que ayuden a los jugadores a entenderlo. Los jugadores deben tener muy claro estas actividades de control para que el juego sea más fluido.

5. Aspectos generales sobre la evolución de SimulES

Analizando las fuentes de información consultadas, tales como: modelos, documentos en general del juego, los juegos educativos y la información recopilada de la actividad reportada en este artículo, identificamos algunos puntos importantes que se detallan a continuación.

Existe una tendencia creciente por la sistematización de los juegos educativos. Aplicando esto a SimulES consideramos que esta solución mejoraría la dinámica del juego y muchas actividades de control serían delegadas al sistema haciendo que los jugadores solo se preocupen por aquello relacionado con el juego, o sea, existen situaciones y controles que no forman parte del *CORE* del juego, pero hacen parte de las reglas del mismo y que no son competencia del estudiante o jugador el lidiar con ellas. Por eso, delegar al sistema para que ejecute estas tareas y validaciones haría que los jugadores se centraran en las actividades concernientes al juego y al aprendizaje en sí de la ingeniería de software.

Analizamos la factibilidad de la implementación del juego, tomamos los modelos con los cuales contábamos y examinamos si estos eran suficientes y consecuentes con lo observado en la actividad.

Los modelos situacionales son, como su nombre lo indica, aquellos encargados de describir situaciones de un contexto o de un sistema, las situaciones se producen en un momento específico y están determinadas por la existencia y ejecución de los actores, una forma de describir estas situaciones es a través de escenarios [11].

Con SimulES el modelado situacional permitió reunir o agrupar en conjuntos realidades o circunstancias que compartían acciones en común. La técnica de escenarios fue la utilizada para el modelado situacional como es descrito en [12].

Tomando estos modelos y lo recopilado en la experiencia descrita en este artículo, identificamos que la interacción entre jugadores se tenía en cuenta, como tampoco lo era en los juegos educativos consultados [12],[13],[14] en los cuales la descripción de modelado es bastante escasa.

Identificamos que el modelaje intencional era adecuado si queríamos modelar o representar interacción entre jugadores, o sea, íbamos a pasar de un modelado situacional [12] a un modelado intencional.

Según [20] el modelado intencional es habitualmente usado para modelar contextos organizacionales basados en las relaciones entre los actores. Los actores en el modelado intencional son

participantes activos de ese contexto organizacional, ellos son entidades que deben efectuar **acciones y alcanzar metas a través del ejercicio de sus habilidades y conocimientos**. Estas metas pueden tener dependencias intencionales entre sí, las cuales ocurren cuando existe una relación entre los actores.

Si llevamos esta definición al contexto de SimULES vemos que los actores, o sea, los jugadores poseen metas dentro del juego, entre ellas *ganar el juego, realizar las mejores jugadas, atacar a sus adversarios* y realizar tareas como *comprar cartas, contratar y despedir ingeniero*. A la vez los ingenieros de software tienen sus propias metas como *completar el producto de software* y para ello ejecutan tareas como *construir artefactos, inspeccionar artefactos y corregir artefactos* entre otras.

Como resultado de este análisis, los modelos intencionales del juego SimULES ya están siendo desarrollados utilizando el método propuesto en [21]. Uno de ellos es mostrado en la Figura 6, las principales metas de los actores son representadas, *Dado sea lanzado, Recursos estén disponibles* y *artefactos sean comprados*. Además de tareas internas para el cumplimiento de los objetivos como es el caso de *Disponer recursos* por parte de SimULES y *Construir software* por el *Ingeniero de Software*.

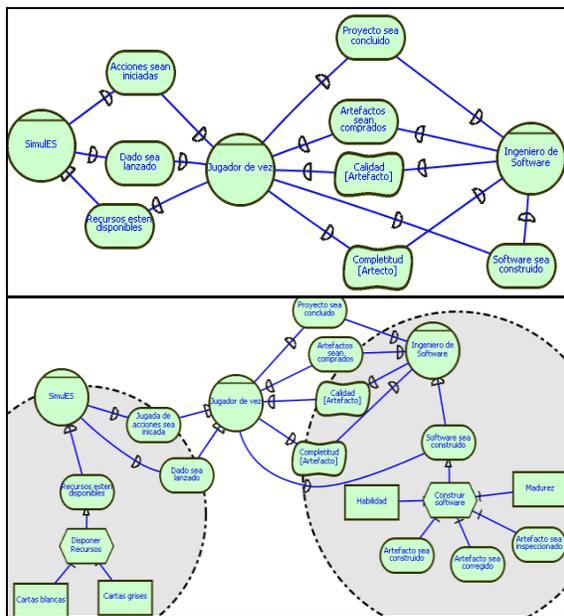


Figura 6. Modelos SD y SR de Jugada de Acciones

En modelos anteriores SimULES no había sido modelado como un actor, en nuestro caso él es un proveedor de recursos como vemos en la Figura 7 y como vemos en la figura 6 al encargarse de disponer las *Cartas blancas* y *Cartas grises*.

Heurísticas similares a la efectuada con SimULES fueron utilizadas para los demás actores mostrados en la Figura 7, estas heurísticas están propuestas en [21] y permiten llegar al modelo de actores o modelo SA.

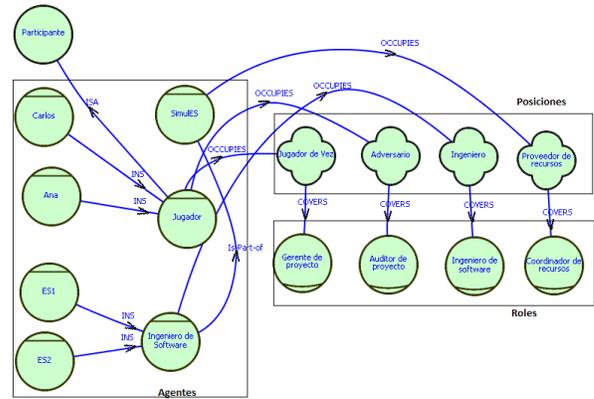


Figura 7. Modelo SA SimULES

Otra propuesta en [21] es catalogar escenarios como *SDsituations*, ellas son una forma de representación estructurada de una situación, compuesta por uno o más elementos de dependencia en la cual los actores participan a través de la colaboración. Las *SDsituations* hacen parte de un modelo intencional mayor que, a través de relaciones recíprocas de dependencia entre actores, tienen la capacidad de concluir una meta situacional. Llevando esta definición a nuestro contexto en SimULES notamos una *SDsituation* en cada jugada dentro del juego, o sea, aquellas que vimos en la Figura 1 cumplen una meta situacional que debe ser realizada para ejecutar la próxima. La red de situaciones que envuelven el juego está representada en el diagrama *SDsituations* y es presentada al lector en la Figura 8, en ella notamos la jerarquía y el factor tiempo como elementos principales para la ejecución de cada una de ellas.

LEGENDA	
Simbolo	Nombre
	SDSituation
	Bifurcación
	Decisión
	Continuación
	Unión
	Inicio
	Fin

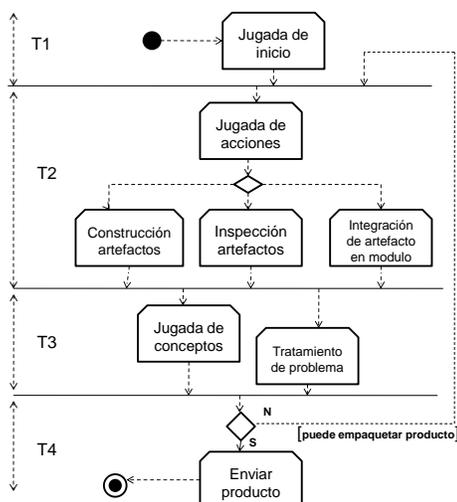


Figura 8. SDSituation SimulES

6. Conclusión

Creemos haber demostrado que las técnicas de elicitación son importantes no solo para empezar un nuevo software, si no también son útiles cada vez que hay necesidad de capturar un nuevo conocimiento y en nuestro caso este conocimiento fue importante para evolucionar el juego tanto en su estructura, modelo e implementación.

Creemos que para elicitar y modelar juegos es necesario tener en cuenta la interacción entre jugadores y los elementos de su ambiente. El modelado de juegos usando el abordaje intencional es novedoso y adecuado para representar las responsabilidades de actores dentro del juego.

Se considera que el software educativo como ambientes de aprendizaje en ciertas áreas se están tornando en una tendencia en crecimiento, la cual es incentivada por recursos tecnológicos cada vez más sofisticados. Como respuesta a esta tendencia consideramos una oportunidad la implementación del juego SimulES, esto permitirá compartir nuestra experiencia y utilizar experiencias previas en modelado intencional y juegos educativos.

A partir de los modelos intencionales creados ya se está trabajando en la implementación del juego, estamos explorando formas de hacer la implementación más próxima al modelado y creemos que este modelado puede ser adecuado para este objetivo.

Los juegos educativos han sido utilizados para ayudar en la enseñanza en diversas áreas del conocimiento y muchas veces despiertan mayor interés por parte del alumno [14]. Consideramos que son una poderosa herramienta para el aprendizaje en la medida

que el estudiante sea estimulado y se capture su interés a través de estos medios. Los juegos educativos llevan a que los profesores sean conductores, estimuladores y evaluadores del conocimiento. De esta forma, el uso de juegos para simular ambientes reales puede mejorar el desempeño de los estudiantes y estimular la generación de vivencias individuales y grupales que ayudarán en su formación como profesionales.

7. Agradecimientos

Julio Cesar Sampaio do Prado Leite agradece la financiación recibida por parte de los órganos de financiamiento CNPq y Faperj.

Elizabeth Suescún Monsalve agradece al órgano de financiamiento CAPES por la beca recibida que viabilizó esta investigación.

Los autores agradecen las contribuciones de los revisores y de los editores que ayudaron a mejorar el presente artículo.

8. Referencias

- [1] E. Figueiredo, C. Lobato, K. Dias, J.C.S.P. Leite, C. Lucena, "Um Jogo para o Ensino de Engenharia de Software Centrado na Perspectiva de Evolução", XV Workshop sobre Educação em Computação (WEI), Rio de Janeiro. co-allocado ao XXVII Congresso da SBC, 2007. pp. 37-46;
- [2] <http://www.problemsandprogrammers.com/> ultimo acceso el 9 de abril de 2009;

- [3] Gramigma, M. R. M. Jogos de empresa. São Paulo: Makron Books, 1994;
- [4] PRESSMAN, Roger. Software Engineering: A Practitioner's Approach, 7ªedición, Mc Graw Hill, 2006;
- [5] http://www.iste.uni-stuttgart.de/se/research/sesam/overview/index_e.html ultimo acceso el 7 de abril de 2009;
- [6] A. Jain, B. Boehm, "SimVBSE: Developing a Game for Value-Based Software Engineering", Proceedings 19th Conference on Software Engineering Education and Training, 2006, pp. 103 -114;
- [7] T. Birkhoelzer, E. Navarro, A. van der Hoek, "Teaching by Modeling instead of by Models", Proceedings of the 6th International Workshop on Software Process Simulation and Modeling, St. Louis, MO, 2005, pp. 4;
- [8] M. Ebner, A. Holzinger "Successful implementation of user-centered game based learning in higher education: An example from civil engineering", Elsevier, Volume 49, Issue 3, November 2007, pp. 873-890
- [9] M. A. Roubidoux, C. M. Chapman, M. E. Piontek "Development and Evaluation of an Interactive Web based Breast Imaging Game for Medical Students", Elsevier, Volume 9, Issue 10, October 2002, pp 1169-1178
- [10] "Avaliação de software educativo para o ensino de Matemática" WIE, Florianópolis, Brasil, 2002.
- [11] J.C.S.P. Leite, G. Rossi, F. Balaguer, V. Maiorana, G. Kaplan, G. Hadad, A. Oliveros, "Enhancing a Requirements Baseline with Scenarios", Proceedings of the Third IEEE International Symposium on Requirements Engineering, IEEE, Annapolis, MD, USA, 1997, pp. 44-53;
- [12] M. Serrano, M. Serrano, F. Napolitano, B. Soares, "Evolução do SimulES Versão 2.0", Monografia Final, Pontifícia Universidad Católica de Rio de Janeiro, Rio de Janeiro, Brasil, 2007;
- [13] <http://www.teccomm.les.inf.puc-rio.br/emagno/simules/> ultimo acceso 16 de noviembre de 2009;
- [14] Gramigma, M. R. M. Jogos de empresa. São Paulo: Makron Books, 1994.
- [15] Durán, Amador. Un Entorno metodológico de Ingeniería de requisitos para Sistemas de Información. Tesis Doctoral. Sept. 2000.
- [16] <http://livrodeengenhariaderequisitos.blogspot.com/2007/09/entrevistas.html> ultimo acceso el 2 de abril de 2009;
- [17] M. G. Christel y K. C. Kang. Issues in Requirements Elicitation. Technical Report CMU/SEI-92-TR-12, Software Engineering Institute, Carnegie Mellon University, 1992.
- [18] Olivera, pablo. "Elicitação de requisitos de software através da utilização de questionários", Tesis de maestría, Pontifícia Universidad Católica de Rio de Janeiro, Rio de Janeiro, Brasil, 2005;
- [19] <http://pes.inf.puc-rio.br/questionario/> ultimo acceso en 30 de noviembre de 2009;
- [20] E. Yu, "Modelling Strategic Relationships for Process Reengineering", PhD Tesis, Graduate Department of Computer Science, University of Toronto, Toronto, Canada, 1995, pp. 124.
- [21] A. P. Oliveira, "Engenharia de Requisitos, um Método de Elicitação, Modelagem e Analise de Requisito", PhD Tesis, PUC Rio, Brasil, 2007, pp. 54.
- [22] <http://transparencia.les.inf.puc-rio.br/> ultimo acceso 15 de noviembre de 2009;
- [23] Claypool, Kaja y Claypool, Mark. Teaching software engineering through game design, ACM SIGCSE Bulletin, v.37 n.3, September 2005.
- [24] Sweedyk, Elizabeth y . Keller, Robert M. Fun and games: a new software engineering course, Proceedings of the 10th annual SIGCSE, 2005.
- [25] Barros, M. O. y Araújo, Renata Mendes. Ensinando Construção de Software Aplicada a Sistemas de Informação do Mundo Real. In: I Fórum de Educação em Engenharia de Software, 2008, Campinas. Anais do I Fórum de Educação em Engenharia de Software, 2008.