

# El juego del diálogo de educación de requisitos

## Requirements elicitation dialog game

Carlos Mario Zapata Jaramillo, Ph.D y Gloria Lucia Giraldo G., Ph.D

Grupo de Investigación en Lenguajes Computacionales - Universidad Nacional de Colombia, Sede Medellín  
{cmzapta, glgiraldog} @unalmed.edu.co

Recibido para revisión 16 de marzo de 2009, aceptado 20 de mayo de 2009, versión final 9 de julio de 2009

**Resumen**—El desarrollo de software suele comenzar con un diálogo entre analistas e interesados, que permite definir las especificaciones de la aplicación por construir. En ese diálogo, es de vital importancia que ambas partes comprendan la relevancia de su papel para lograr buenas especificaciones. Si bien la enseñanza tradicional de la Ingeniería de Software define estos roles, por lo general esa relevancia sólo se adquiere con la práctica. Por ello, en este artículo se emplea un juego como estrategia didáctica para generar conciencia sobre la importancia de esos roles durante las etapas iniciales del ciclo de vida de una aplicación y se discuten los resultados obtenidos al practicar el juego con diferentes grupos de estudiantes.

**Palabras clave**—Educación de Requisitos, Juego, Diálogo, Especificaciones de Software.

**Abstract**—Commonly, software development begins with a stakeholder-analyst dialog, in order to define the specs of the application to be developed. In this dialog, understanding about the importance of dialog roles is crucial for obtaining good specs. Traditional software engineering teaching defines these roles, but only by practicing this process such roles can be adequately understood. By this reason, in this paper we employ a game as a didactic strategy for generating consciousness about the importance of dialog roles in the first phases of software development lifecycle. We also discuss the obtained results in playing the game with several groups of students.

**Keywords**—Requirements Elicitation, Gaming, Dialog, Software Specs.

### I. INTRODUCCIÓN

La elaboración de software de calidad exige que los requisitos de los interesados en la aplicación se recopilen, procesen y conviertan en buenas especificaciones. La educación de requisitos, como se conoce este conjunto de procesos, es una de las tareas más importantes del desarrollo de software [1].

Pressman [2] y Sommerville [3], entre otros, presentan algunas técnicas que permiten llevar a cabo una adecuada educación de requisitos. Por lo general, este proceso se suele realizar mediante entrevistas y otras técnicas que emplean el diálogo interpersonal, tales como las tormentas de ideas, las técnicas para facilitar la especificación de aplicaciones, el análisis de escenarios y el prototipado, entre otros.

Por ser un proceso que se desarrolla entre humanos, la educación de requisitos suele presentar problemas, especialmente por las diferencias de énfasis y conocimiento que suelen tener los interesados y los analistas [4]. El hecho de que se realice en forma de diálogo, por ejemplo, hace que los interesados tiendan a explayarse en sus explicaciones, entregando grandes cantidades de información que, en general, no se incorporan adecuadamente en las especificaciones del software. En especial, los interesados no se percatan de la importancia de entregar respuestas concisas e informativas sobre el dominio de su aplicación.

La enseñanza tradicional de la Ingeniería de Software define los actores del proceso de educación y sus funciones dentro del diálogo. Para ello, se suelen emplear como estrategias didácticas las clases expositivas y la realización de pequeños proyectos prácticos (“de juguete”) [5], en los cuales los alumnos sólo adquieren competencias básicas pero no practican situaciones reales del proceso de educación. Además, esta enseñanza se dirige a personal técnico y no propiamente a interesados, que suelen desconocer el lenguaje técnico (los diagramas y modelos) que se emplean en la Ingeniería de Software [6]. En particular, los interesados suelen desconocer el hecho de que el diálogo que realizan para la educación de

requisitos es el que sirve para elaborar los diferentes diagramas que se emplean luego para especificar la aplicación por desarrollar.

Por ello, es común que los interesados tiendan a aportar mucha información dentro del proceso, pero con los problemas propios del lenguaje natural: ambigüedades, manejo indiscriminado de sinónimos y entrega de gran cantidad de información que luego difícilmente se podrá traducir en diagramas [6].

Como alternativa a la enseñanza tradicional de la Ingeniería de Software, se vienen empleando juegos didácticos que buscan afianzar los conocimientos impartidos en las sesiones teóricas, simulando situaciones de común ocurrencia en el ciclo de vida del software. Algunos de esos juegos son: problemas y programadores [5], el juego de los requisitos [7], el juego de la consistencia [8] y el juego del desarrollo de software [9], en los cuales se alerta sobre los peligros que puede generar un inadecuado proceso de educación de requisitos, pero no se practica, en sí, el proceso mismo, ni se presenta a los participantes un resultado concreto de la traducción del diálogo en diagramas.

Buscando superar estas limitaciones, en este artículo se propone “el juego del diálogo de educación de requisitos” como una estrategia didáctica para conscientizar a los actores del diálogo—dado que éste origina los requisitos de las aplicaciones de software—sobre la importancia de suministrar respuestas concisas relacionadas con temas fundamentales para el desarrollo de su aplicación. Además, se busca que los participantes practiquen de forma entretenida la elaboración de un esquema preconceptual [4], que es un diagrama previo para la realización de otros diagramas que sirven para especificar la aplicación, tomando como punto de partida un diálogo controlado entre un analista y un interesado.

Este artículo posee la siguiente estructura: en la sección 2 se presentan algunos conceptos teóricos necesarios para fundamentar el resto del artículo; en la sección 3 se discute la importancia de los juegos en la ingeniería de software; en la sección 4 se presenta el juego del diálogo de educación de requisitos; en la sección 5 se discuten los resultados de la aplicación del juego a varios grupos de estudiantes; finalmente, en las secciones 6 y 7 se presentan las conclusiones y el trabajo futuro que se puede derivar de este juego.

## II. MARCO TEÓRICO

### A. Educación de Requisitos de Software

El término *requirements elicitation* se emplea en la literatura técnica para denotar el proceso mediante el cual se capturan, analizan, procesan y convierten los requisitos de una aplicación de software [10] desde las necesidades y expectativas de los interesados hasta las especificaciones formales o semiformales. En general, la educación de requisitos la realiza un equipo compuesto por personal técnico (analistas, diseñadores y programadores) y personal perteneciente al área de negocios que requiere la aplicación (en adelante denominados “interesados”, como una traducción del término inglés *stakeholder*).

La técnica más común para iniciar el proceso de educación de requisitos es la entrevista, en la cual se procura realizar un diálogo, por lo general dirigido, entre analistas e interesados para determinar los conceptos fundamentales que rodean el dominio de una aplicación [2]. Otras técnicas como JAD (Joint Application Development), TFEA (Técnicas para Facilitar la Especificación de Aplicaciones), el prototipado, el análisis de escenarios o IBIS (Issue-based Information System) se basan, al igual que las entrevistas, en el diálogo estructurado o semiestructurado entre los analistas y los interesados en búsqueda de las necesidades y expectativas que rodean la aplicación por construir, con el fin de especificar los requisitos [2], [3] [10], pero presentan algunas variaciones a esta técnica.

Christel y Kang [10] señalan que el proceso de educación de requisitos es altamente complejo, pues presenta tres tipos de problemas: alcance (pues el interesado puede entregar mucha información, pero con poca utilidad para la aplicación por desarrollar), comprensión (pues los interesados pueden comprender poco sus necesidades y expectativas, mientras los analistas poco conocen sobre el dominio de la aplicación) y volatilidad (pues los requisitos frecuentemente cambian con el tiempo). Estos problemas justifican la necesidad de encontrar estrategias que ayuden a analistas e interesados a solucionar los inconvenientes propios del diálogo que da origen a la especificación de los requisitos de la aplicación

### B. Modelos de Diálogo

El habla es el mecanismo mediante el cual dos o más actores intercambian información; por su importancia dentro de los mecanismos de comunicación, las tecnologías del habla ocupan un lugar preponderante dentro de las tecnologías lingüísticas, pues permiten el estudio de las interacciones entre los diferentes actores (humanos o máquinas) mediante el habla [11].

Dentro de las tecnologías del habla se estudian los fenómenos lingüísticos que ocurren en palabras o frases. Sin embargo, el lenguaje no consta sólo de estos elementos, pues las frases se pueden agrupar en unidades mayores con una coherencia propia, en lo que se suele denominar “discurso” [12]. Existen varios tipos de discurso, entre los que se destacan los monólogos y los diálogos. Los diálogos se diferencian de otros tipos de discursos por tres características principales: (1) los actores implícita o explícitamente definen una manera de alternar los turnos para hablar, (2) los actores deben acordar un tema común y (3) de acuerdo con el tema, algunas respuestas se pueden sobreentender, lo que define cierto grado de implicación [12].

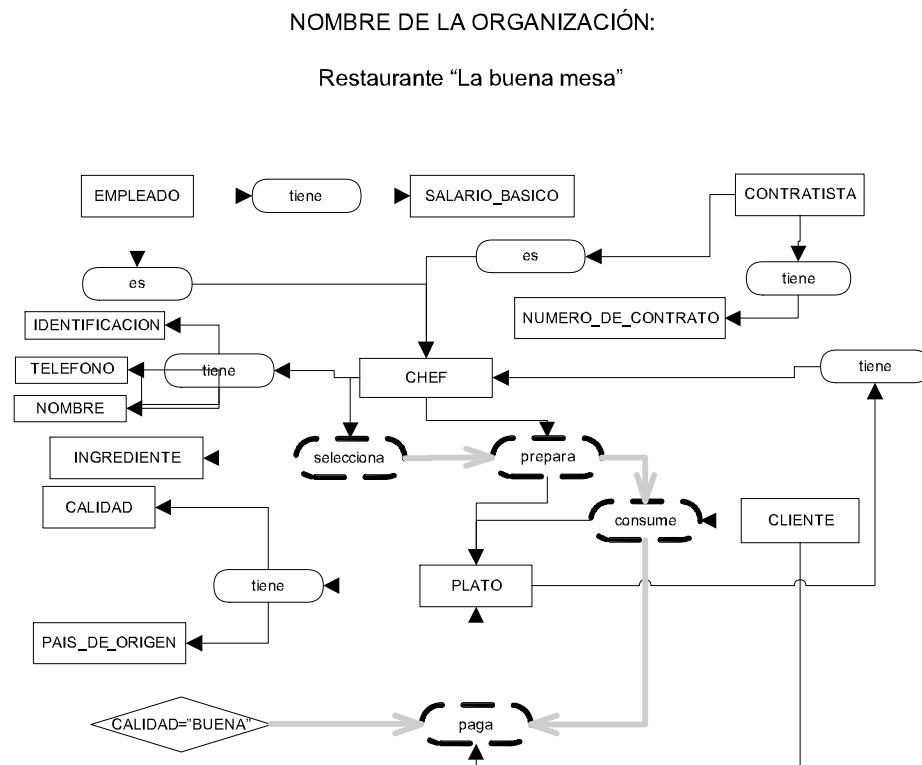
Un modelo de diálogo procura representar de manera gráfica o textual las interacciones que realizan dos actores durante el diálogo que sostienen [13]. Una gran cantidad de los modelos de diálogo existentes se dedican al análisis de las interacciones entre actores humanos, especialmente en temas como la reserva de trenes o tiquetes aéreos [12]. Una primera aproximación a modelos de diálogo para la educación de requisitos se puede consultar en [14].

### C. Esquemas Preconceptuales

Son diagramas que se emplean para representar el conocimiento de manera cercana a como se expresarían en lenguaje natural y que sirven para obtener algunos de los diagramas típicos que se utilizan para modelar las especificaciones del software [4]. Para su construcción, los interesados deben aportar su conocimiento del dominio y los analistas sus conocimientos sobre modelado; así, empleando el esquema preconceptual como elemento de validación del dominio, tanto analistas como interesados deben construir una representación, lo más completa y precisa del dominio de la aplicación para generar, a partir de allí, los principales diagramas que especifican los requisitos del software.

Los esquemas preconceptuales poseen una sintaxis cercana a un subconjunto controlado del lenguaje natural, lo que hace que cualquier interesado los pueda leer y comprender. Un ejemplo de esquema preconceptual se puede apreciar en la Fig. 1.

En dicha figura, los rectángulos se denominan “conceptos” y se asocian con sustantivos y frases nominales del dominio; los óvalos en línea continua se denominan “relaciones estructurales” y contienen los verbos conjugados “es” y “tiene”; los óvalos en línea discontinua se denominan “relaciones dinámicas” y contienen verbos asociados con acciones; los rombos se denominan “condicionales” y tienen expresiones que se deben cumplir para poder realizar la relación dinámica con la cual se conectan; finalmente, las flechas delgadas unen conceptos con relaciones estructurales o dinámicas y viceversa; finalmente, las flechas gruesas unen relaciones dinámicas entre sí o condicionales con relaciones dinámicas, conformando situaciones causa-efecto que se leen, por ejemplo, así: “cuando chef selecciona ingrediente, entonces chef prepara plato” o “cuando calidad='buena', entonces cliente paga plato”. Con estos elementos, se puede construir un conjunto de diagramas para especificar requisitos, según se muestra en [4].



**Figura 1.** Un ejemplo de esquema preconceptual

### III. LOS JUEGOS EN LA INGENIERÍA DE SOFTWARE

Baker *et al.* [5] realizan una reflexión de la forma como se vienen enseñando los conceptos correspondientes a la Ingeniería de software desde sus inicios a finales de los años sesenta. Así, concluyen que dicha enseñanza utiliza una mezcla de clases expositivas y pequeños proyectos prácticos. Estas estrategias no suelen preparar adecuadamente a los ingenieros de software para enfrentar las condiciones que van a encontrar

en ámbitos reales de trabajo, puesto que las simulaciones a pequeña escala y con las limitaciones de tamaño y realismo que se pueden lograr con los proyectos “de juguete” que estos autores nombran, no permiten prever condiciones reales de trabajo práctico.

En esas estrategias, además, los contenidos se suelen concentrar en habilidades de programación y matemáticas, lo cual, según Kitchenham *et al.* [15], constituye un error, pues los ingenieros de software también requieren algunas

habilidades que poco se pueden desarrollar con la manera tradicional de enseñanza de esta profesión, como el adecuado manejo de los grupos de trabajo, la planeación de los proyectos a gran escala, las respuestas a los riesgos del entorno y la recolección adecuada de las necesidades y expectativas relacionadas con un dominio en particular. A este respecto, Boehm [16]

en su reflexión sobre el pasado, presente y futuro de la Ingeniería de Software y su forma de enseñarla, concluye que es importante, a futuro, combinar la enseñanza tradicional con nuevas corrientes que confronten los estudiantes con las condiciones reales en que se desempeñarán profesionalmente en el futuro y sugiere los juegos instructivos en clase como estrategias interesantes para tomar en consideración.

En esta línea de trabajo, se vienen realizando diferentes tipos de experiencias conducentes a complementar la enseñanza tradicional de los conceptos en Ingeniería de Software con juegos. En esta línea se pueden mencionar: “problemas y programadores” [5], que es un juego de cartas para simular el trabajo de los programadores a lo largo del ciclo de vida de software y experimentar las respuestas que podrían entregar ante situaciones desfavorables; “el juego de los requisitos” [7], que simula en una sesión de dos o cuatro horas de duración el desarrollo de una aplicación completa, incluyendo la elaboración de la documentación técnica que servirá de soporte; “el juego de la consistencia” [8], un rompecabezas para armar cuatro diagramas típicos del desarrollo de software a partir de un discurso preestablecido; finalmente, “el juego del desarrollo de software” [9], un juego que busca generar conciencia, en los actores del proceso de desarrollo de software, sobre la importancia de respetar los requisitos definidos para una aplicación.

En estas experiencias, no se toma en consideración el origen de los requisitos del software, proceso que se realiza por lo general a partir de técnicas basadas en entrevistas o diálogos.

Esas experiencias se suelen dirigir a ingenieros de software, que son personas con entrenamiento técnico, pero poco se emplean con interesados reales, quienes, en general, poseen características diferentes y son de vital importancia en el proceso de educación de requisitos. En efecto, como lo señalan Britton y Jones [6] los lenguajes de especificación de requisitos se deberían modificar para poder representar el dominio de una aplicación en una forma entendible para los interesados, puesto que la imposibilidad de validar tales especificaciones constituye uno de los problemas fundamentales durante todo el proceso de educación de requisitos.

Por los motivos discutidos en las secciones 2 y 3, en la siguiente sección se presenta un juego que combina su facilidad de adaptación (a públicos entrenados y no entrenados en los términos técnicos de la educación de requisitos) con la claridad que se logra en el paso entre un diálogo controlado y su especificación en un esquema gráfico entendible por el interesado.

#### IV. EL JUEGO DEL DIÁLOGO DE EDUCACIÓN DE REQUISITOS

##### A. *Objetivo del juego*

Lograr que los participantes se conscienticen sobre la importancia del diálogo assertivo y preciso en la educación de requisitos, además de interiorizar de forma práctica la transición entre el diálogo y un lenguaje gráfico para especificación de requisitos, representado por los esquemas preconceptuales. Además, el juego se puede emplear para recordar conceptos fundamentales o conocimientos previos de algún área en particular, puesto que incluye un conjunto de preguntas que se deben resolver para capturar los elementos del tablero de juego.

##### B. *Materiales*

Para la realización completa del juego, se debe contar con los siguientes materiales:

- Un tablero de juego como el que se muestra en la Fig. 2. Cada celda se debe cubrir en forma secuencial con un número del 1 al 56. Se sugiere que el tablero se haga en una hoja electrónica, colocando el número que cubre cada celda de forma que se pueda descubrir el contenido fácilmente al iniciar el juego. Se debe notar que en dicha figura existen varios tipos de contenido de celdas: sustantivos (escritos en mayúsculas), verbos de acción (escritos en minúsculas), comodines (las palabras “sustantivo” y “verbo de acción”), restricciones (las celdas “DIA=VIERNES” y “PLATO.CALIDAD=BUENA”), nombres de dominio (las celdas “Restaurante la buena mesa” y “Comercializadora le vendo todo”) y órdenes especiales del juego (las celdas “intercambie palabras”, “ pierde palabra” y “ pierde turno”).
- Un cuestionario con 56 preguntas de conocimientos orientados al público con el cual se va a practicar el juego. Por ejemplo, si el juego se dirige a público entrenado en Ingeniería de Software, las preguntas pueden ser correspondientes a lenguajes de programación, gestión de proyectos de software, lenguajes de modelado, etc. Si se trata de público no entrenado con alguna afinidad (por ejemplo, como se verá en los resultados, el juego ya se practicó con estudiantes de pregrado y postgrado que tenían como afinidad la lingüística computacional), el cuestionario se deberá elaborar con preguntas relativas a esa actividad.
- Una hoja para diligenciar el diálogo y otra para diligenciar el esquema preconceptual, con la apariencia que se muestra en las Fig. 3 y 4, respectivamente.
- Una hoja de solución correspondiente a cada uno de los dos dominios en que se divide el juego. El esquema preconceptual correspondiente a uno de los dominios es el que se mostró previamente en la Fig. 1. El diálogo que se muestra en la Fig. 5 forma parte de la solución para el otro dominio.
- Un computador con la hoja electrónica del tablero de juego y un proyector.

CHEF	CONTRATISTA	SALARIO_BASICO	Confirma	Entrega	Intercambia palabras	NUMERO	Pierde palabra
ORDEN	Realiza	SALARIO_BASICO	Sustantivo	Reporta	Prepara	Pierde turno	PAIS_DE_ORIGEN
NUMERO_DE CONTRATO	INGREDIENTE	Pierde turno	Selecciona	VENDEDOR	Verbo de acción	Sustantivo	Pierde palabra
NOMBRE	IDENTIFICACION	Pierde palabra	Sustantivo	TELEFONO	Verbo de acción	Sustantivo	PLATO CALIDAD= "BUENA"
Paga	Intercambia palabras	EMPLEADO	NUMERO_DE CONTRATO	PLATO	Sustantivo	VENTA	Verbo de acción
Sustantivo	Restaurante "La buena mesa"	ALMACENISTA	CLIENTE	COMISION	CONTRATISTA	Intercambia palabras	NOMBRE
DIA="VIERNES"	DESTINO	IDENTIFICACION	EMPLEADO	Consumo	Comercializadora "Le vendo todo"	CALIDAD	Pierde turno

**Figura 2.** Tablero de juego

### C. Procedimiento

Previamente a la iniciación del juego, el director del mismo debe conservar el cuestionario con las respuestas y la hoja de

solución para cada uno de los dominios. Esta información sólo la conocerá el director y le servirá para determinar la asignación de las casillas y, al final, el equipo ganador. Además, deberá

El siguiente es un diálogo entre un analista y un interesado para capturar los requisitos de una aplicación de software.

ANALISTA: Buenos días. El objetivo de esta entrevista es aclarar la información concerniente al dominio del problema que vamos a trabajar. Por favor responda de la forma más concreta posible.

INTERESADO: De acuerdo. Comencemos.

ANALISTA: ¿Cuál es el nombre de esta organización?

INTERESADO:

ANALISTA: Elabore, por favor, una lista de los actores (internos o externos que participan en el proceso).

INTERESADO:

ANALISTA: ¿Quiénes pueden ser ?

INTERESADO:

ANALISTA: ¿Puede mencionar características del ?

INTERESADO:

ANALISTA: ¿Puede mencionar características del ?

INTERESADO:

ANALISTA: ¿Puede mencionar características del ?

INTERESADO:

ANALISTA: Elabore una lista de las funciones del

INTERESADO:

ANALISTA: Elabore una lista de las funciones del

INTERESADO:

ANALISTA: ¿Puede mencionar características de ?

INTERESADO:

ANALISTA: ¿Qué se requiere para que se dé que   ?

INTERESADO:

ANALISTA: ¿Puede establecer una secuencia en las funciones que enunció?

INTERESADO: 1.   
2.   
3.   
4.

ANALISTA: Muchas gracias por su valiosa información. Lo estaremos contactando después para aclarar las dudas que surjan en el proceso.

INTERESADO: A usted. Seguimos en contacto.

**Figura 3.** Hoja para diligenciar el modelo de diálogo

ceriorarse que el computador esté listo para proyectar la imagen del tablero de juego. Posteriormente, dividirá los participantes en un número par de equipos. Se recomiendan equipos de 3 a 6 personas. Por ejemplo, para un grupo de 30 personas, se recomienda organizar 6 grupos de cinco integrantes cada uno. Finalmente, en una sesión de aproximadamente quince minutos deberá explicar la dinámica del juego, según las reglas que se enuncian seguidamente, y la forma de leer los esquemas preconceptuales.

El juego se divide en dos partes: asignación de colores a las casillas del tablero y diligenciamiento de las hojas de modelo de diálogo y esquema preconceptual.

#### Reglas para la primera parte

- El director asigna un orden numérico a los grupos y un color de juego que será igual para los grupos pares y otro para los impares. En la segunda parte, cuando las casillas se destapen completamente, cada grupo podrá utilizar sólo las casillas de su color más aquellas correspondientes a comodines.
- Iniciando el orden numérico, cada grupo selecciona un número del tablero que corresponderá a una pregunta del cuestionario. Si el grupo en su turno responde acertadamente la pregunta, el director destapa la casilla correspondiente, pudiendo ocurrir una de varias situaciones: (1) si se trata de una casilla que contiene un sustantivo, un verbo de acción, una restricción o un título o de las casillas “sustantivo” o “verbo de acción”, el director asigna a la casilla el color del grupo que respondió; (2) si se trata de la casilla “intercambie palabras”, el grupo que respondió acertadamente debe cambiar una palabra de su propio color por una palabra del color contrario; (3) si se trata de la casilla “ pierde palabra”, el grupo numérico siguiente decide cuál de las palabras del grupo que respondió acertadamente se le debe quitar el color; (4) si se trata de la casilla “ pierde turno” la única acción que se realiza es impedir que en el próximo turno que le toque al grupo que respondió acertadamente pueda seleccionar un número del tablero. Este proceso se repite hasta que se destapen todas las casillas del tablero.
- En ocasiones, y dependiendo de la complejidad de las preguntas del cuestionario, pueden existir varias preguntas que ninguno de los grupos respondan adecuadamente. En este caso, se recomienda realizar rondas completas de “amnistía”, en las cuales simplemente con seleccionar el número, el director suministra la respuesta y sigue las instrucciones dadas por la regla anterior.

NOMBRE DE LA ORGANIZACIÓN:

---

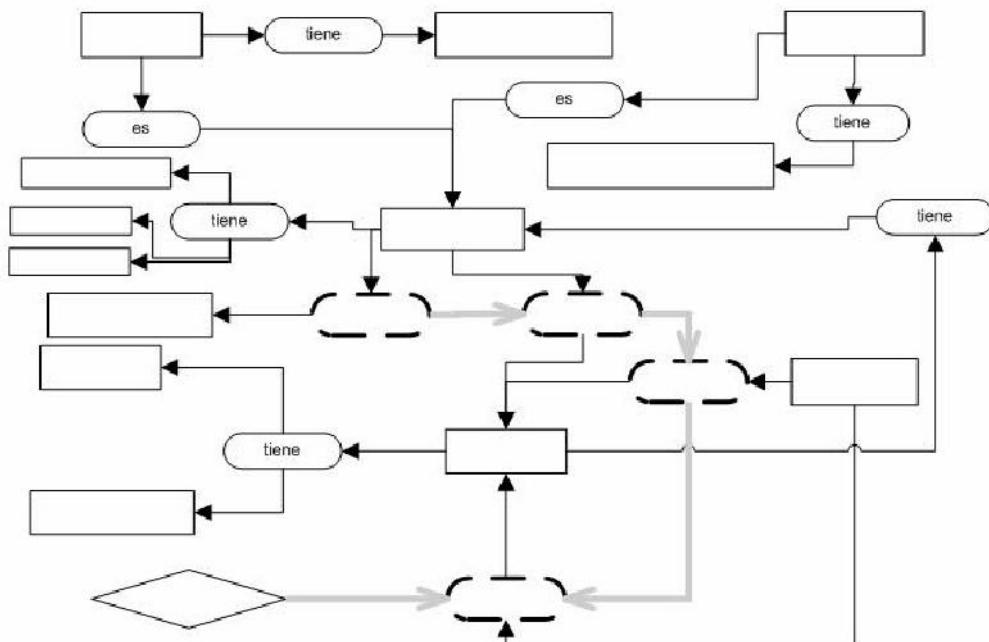


Figura 4. Hoja para diligenciar el esquema preconceptual

### **Reglas para la segunda parte**

- Con el tablero completamente destapado y algunas de las casillas asignadas a uno u otro color, cada uno de los grupos debe proceder a diligenciar las hojas correspondientes a las Fig. 3 y 4. Para este proceso, los grupos deben tomar en consideración varias situaciones: (1) no todas las casillas con un color definido pueden servir para diligenciar la información (por ejemplo, es muy difícil que la palabra “chef” pueda caber en la “comercializadora le vendo todo”; (2) las casillas “sustantivo” son comodines que pueden tomar el valor de cualquier casilla de sustantivo, título o restricción que se encuentre en el tablero, ya sea en blanco o con un color propio o ajeno; (3) las casillas “verbo de acción” se pueden tomar el valor de cualquier casilla de verbo de acción que se encuentre en el tablero, ya sea en blanco o con un color propio o ajeno.
- Un asistente del director del juego realiza la valoración de las soluciones de los diferentes grupos con las siguientes calificaciones: (1) por cada palabra colocada correctamente en el diálogo se asignan dos puntos; (2) por cada palabra colocada correctamente en el esquema preconceptual se asignan dos puntos; (3) por cada palabra colocada incorrectamente pero perteneciente al dominio, ya sea en el diálogo o en el esquema preconceptual se asigna un punto; (4) las palabras que no pertenezcan al dominio o los comodines mal empleados no generan puntos.
- Simultáneamente, el director del juego realiza una sesión de cierre para discutir la dinámica del juego y sus resultados. En este caso, el director cumple sólo el papel de moderador, con el fin de que sean los participantes los que conjuntamente lleguen a las conclusiones adecuadas para el juego.

El siguiente es un diálogo entre un analista y un interesado para capturar los requisitos de una aplicación de software.

ANALISTA: Buenos días. El objetivo de esta entrevista es aclarar la información concerniente al dominio del problema que vamos a trabajar. Por favor responda de la forma más concreta posible.

INTERESADO: De acuerdo. Comencemos.

ANALISTA: ¿Cuál es el nombre de esta organización?

INTERESADO: Comercializadora “Le vendo todo”

ANALISTA: Elabore, por favor, una lista de los actores (internos o externos que participan en el proceso).

INTERESADO:	EMPLEADO CONTRATISTA VENDEDOR ALMACENISTA
-------------	--

ANALISTA: ¿Quiénes pueden ser VENDEDOR?

INTERESADO:	EMPLEADO CONTRATISTA
-------------	-------------------------

ANALISTA: ¿Puede mencionar características del EMPLEADO?

INTERESADO:	SALARIO_BASICO
-------------	----------------

ANALISTA: ¿Puede mencionar características del CONTRATISTA?

INTERESADO:	NUMERO_DE_CONTRATO
-------------	--------------------

ANALISTA: ¿Puede mencionar características del VENDEDOR?

INTERESADO:	IDENTIFICACION COMISION NOMBRE
-------------	--------------------------------------

ANALISTA: Elabore una lista de las funciones del VENDEDOR

INTERESADO:	Realiza VENTA Reporta ORDENES
-------------	----------------------------------

ANALISTA: Elabore una lista de las funciones del ALMACENISTA

INTERESADO:	Confirma ORDENES Entrega ORDENES
-------------	-------------------------------------

ANALISTA: ¿Puede mencionar características de la VENTA?

INTERESADO:	NUMERO DESTINO
-------------	-------------------

ANALISTA: ¿Qué se requiere para que se de que ALMACENISTA entrega ORDENES?

INTERESADO:	DIA="VIERNES"
-------------	---------------

ANALISTA: ¿Puede establecer una secuencia en las funciones que enunció?

INTERESADO:	1. VENDEDOR Realiza VENTA 2. VENDEDOR Reporta ORDENES 3. ALMACENISTA Confirma ORDENES 4. ALMACENISTA Entrega ORDENES
-------------	---

ANALISTA: Muchas gracias por su valiosa información. Lo estaremos contactando después para aclarar las dudas que surjan en el proceso.

INTERESADO: A usted. Seguimos en contacto.

**Figura 5.** Solución al modelo de diálogo para el segundo dominio

## V. RESULTADOS DE LA APLICACIÓN DEL JUEGO

El juego del diálogo de educación de requisitos se practicó en cuatro grupos con características disímiles. Las características relevantes de estos grupos se listan en la tabla I.

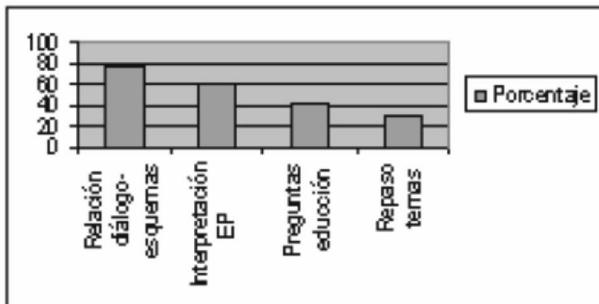
La validación se realizó mediante una encuesta que contenía tres preguntas: (1) ¿Qué aprendió del juego? (2) ¿Qué cree que se necesitaba para ganar el juego? (3) ¿Qué le cambiaría al juego?

Las preguntas eran abiertas para no sesgar las respuestas de los participantes. Posteriormente, se agruparon dichas respuestas en categorías comunes.

Las respuestas a la primera pregunta se pueden consultar en la Fig. 6. La respuesta mayoritaria (mencionada por el 76,9% de los participantes) fue la relación que existe entre el diálogo y los esquemas preconceptuales. En efecto, se notó en la segunda parte del juego una tendencia a complementar las respuestas comparando el modelo de diálogo y el esquema preconceptual y completando la información de una y otra fuente. Esto permite validar la completitud de la descripción del dominio, una de las principales características de calidad de la documentación. Otra respuesta mencionada por el 60,3% de los participantes se relaciona con la interpretación de los esquemas preconceptuales. A este respecto, los participantes manifestaron satisfacción con la forma de representación de los esquemas, por su simplicidad de lectura. Las otras dos respuestas fueron la claridad en los elementos para preguntar en un diálogo de educación (42,3%) y el repaso a preguntas pertenecientes a su área de conocimientos (29,5%).

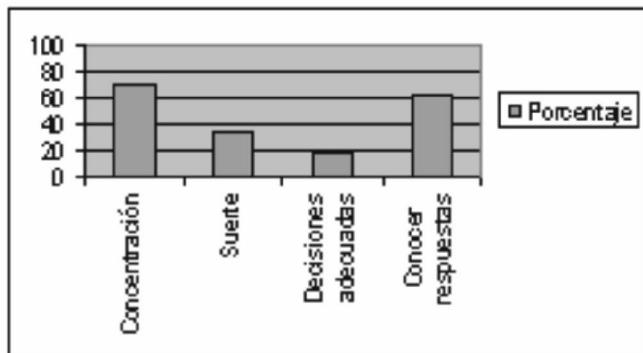
**TABLA I.** CARACTERÍSTICAS DE LOS GRUPOS EN QUE SE PRACTICÓ EL JUEGO.

# jug	Formación	Especialidad	País
8	Estudiantes de pregrado y postgrado de distintas carreras	Lingüística Computacional	Colombia
23	Estudiantes de Maestría en Sistemas con diferentes títulos de pregrado	Ingeniería de Software	Colombia
15	Participantes en un congreso sobre ingeniería de software e ingeniería del conocimiento	Ingeniería de software	Ecuador
32	Analistas y desarrolladores de una empresa de servicios	Programación	Colombia



**Figura 6.** Respuestas a la pregunta sobre el aprendizaje del juego.

En relación con la segunda pregunta, las respuestas se pueden apreciar en la Fig. 7. Una gran mayoría de los participantes respondió que se requería concentración (76,9%) y conocer las respuestas del cuestionario (61,5%). Estos elementos también se pudieron observar en el desarrollo del juego, puesto que algunos grupos llevaban registros de aquellas preguntas que no podían contestar, incrementando la probabilidad de ganar la celda correspondiente. Por el contrario, aquellos grupos que poca atención prestaron al juego caían repetidamente en preguntas que ningún grupo lograba responder. Otras respuestas minoritarias a lo que se necesitaba para ganar el juego fueron suerte (34,6%) y toma de decisiones adecuadas (17,9%), que se manifestaron en el desarrollo del juego en las casillas de órdenes especiales del juego. Si bien es una experiencia traumática en el juego obtener un castigo por la respuesta acertada a una pregunta (tal como la pérdida del turno o la pérdida de una palabra), es importante que los practicantes tengan conciencia al respecto, porque en el entorno real ese tipo de cosas ocurren: no siempre quien tiene el conocimiento es el que gana mayor reconocimiento. Además, la toma de decisiones inadecuadas suele afectar negativamente el desempeño del juego, tal como puede ocurrir también en un proceso real de educación de requisitos.



**Figura 7.** Respuestas a la pregunta sobre lo necesario para ganar el juego.

En cuanto a la tercera pregunta, cuyas respuestas se pueden apreciar en la Fig. 8, no hubo respuestas mayoritarias, aunque el 44,9% de los participantes manifestaron su conformidad con el juego expresando que no le cambiarían nada. Un 35,9% de los participantes manifestó que cambiarían la interfaz gráfica del juego, puesto que la hoja electrónica que se empleó en todas las ediciones en ocasiones dificultaba el normal desempeño del juego. El 29,5% de los participantes solicitó más tiempo para la capacitación en la elaboración de los esquemas preconceptuales, pero esta sugerencia (a pesar de surgir de los grupos iniciales en que se aplicó el juego) no se atendió y se siguió con la misma dinámica, pues en el fondo lo que se pretende es que los participantes descubran las relaciones entre el diálogo y los esquemas y no tanto que se convierta en una capacitación en elaboración de los esquemas. Otras respuestas fueron la inclusión de más comodines en el tablero de juego (19,2%) y la adición de puntaje a las respuestas acertadas del cuestionario.

(14,1%). Cabe aclarar que el tablero de juego definitivo (que se presenta en la Fig. 2) tiene más comodines de los que se utilizaron en las dos prácticas iniciales del juego, por lo cual es lógico que en las últimas encuestas no se presentara mayoritariamente esta respuesta. En cuanto al puntaje por las respuestas acertadas, no se incluyó aún porque se considera que se daría relevancia a un proceso del juego que es meramente selectivo, pues el cuestionario sólo agrega un elemento de competencia al premiar (o incluso castigar, como se vio previamente) las personas que más conocimientos exhiben. A este respecto, una edición del juego con mayores limitaciones de tiempo podría obviar el cuestionario y simplemente asignar las casillas por selección de los números, tal como se hace en las rondas de amnistía que se discutieron en la sección anterior.

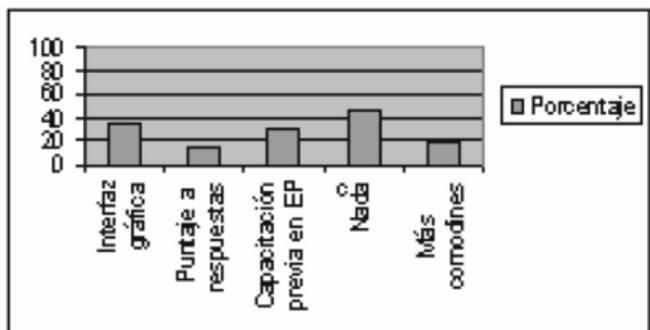


Figura 8. Sugerencias de cambio de los participantes del juego

## VI. CONCLUSIONES

Como una estrategia didáctica que pretende generar conciencia sobre la importancia del diálogo en las fases iniciales del desarrollo de software, más precisamente en la educación de requisitos de una aplicación, se presentó en este artículo el juego del diálogo de educación de requisitos. El juego, practicado hasta ahora en cuatro grupos con características diferentes, se validó mediante encuestas que muestran una tendencia clara al cumplimiento de su objetivo principal, pues sus practicantes manifestaron que adquirieron ciertas habilidades como las pretendidas en el objetivo del juego.

El cuestionario inicial que acompaña al tablero se constituye en un elemento que puede hacer versátil el juego, especialmente cuando se trata de personal no técnico en ingeniería de software o especialidades afines. Es importante señalar que se recomienda el juego como medio de entrenamiento en las empresas, pudiendo conformar grupos de participantes entrenados o no en desarrollo de software y con niveles de conocimiento que podrían ser disímiles. Esta flexibilidad hace que este juego se convierta en una alternativa diferente para complementar el aprendizaje de elementos técnicos en personal entrenado o para concientizar al personal no entrenado sobre la importancia de la precisión en las respuestas que suministren en un proceso real de educación de requisitos.

## VII. TRABAJO FUTURO

Según las respuestas de los participantes y el análisis de la aplicación del juego en los diferentes grupos, existen varias líneas de acción en trabajo futuro que pueden complementar el desarrollo que hasta ahora tiene el juego. Se pueden mencionar las siguientes:

- Es importante mejorar la interfaz gráfica del juego para que no sea un factor que reduzca el tiempo dedicado a la práctica real.
- Se debe trabajar en la generación de cuestionarios para diferentes ámbitos, pues por los grupos que practicaron el juego hasta ahora sólo se cuenta con tres tipos de cuestionarios. Sería importante hacer una proyección sobre los grupos que potencialmente podrían recibir una capacitación de este tipo para definir las temáticas que se podrían abordar con el juego, lo que le suministraría dinamismo a la preparación del juego mismo.
- Se podrían explorar otras temáticas de la Ingeniería de Software que se pudieran practicar con personal no entrenado, para facilitar los procesos de desarrollo de software. Algunos ejemplos podrían ser el manejo de los diferentes puntos de vista y las ontologías del dominio. Con estas temáticas se podrían diseñar nuevos juegos complementarios a este y que puedan servir en un futuro para ofrecer un curso de Ingeniería de Software para personal no técnico en esta área. Una visión tal permitiría agilizar los procesos de desarrollo de software por una vía poco explorada hasta ahora: la capacitación de los interesados para que colaboren más asertivamente en el proceso de desarrollo de software.

## AGRADECIMIENTO

Este artículo se realizó en el marco del proyecto de investigación “Un modelo de diálogo para la generación automática de especificaciones en UN-Lencep”, financiado por la DIME—Dirección de Investigación de la sede Medellín de la Universidad Nacional de Colombia

## REFERENCIAS

- [1] J. C. Leite. 1987. A survey on requirements analysis, Advanced Software Engineering Project. Technical Report RTP-071, Department of Information and Computer Science, University of California at Irvine.
- [2] R. Pressman, 2005. Software Engineering: A Practitioner's Approach, 6th ed. McGraw Hill (New York).
- [3] I. Sommerville, 2001. Software Engineering (6th. Ed). Addison-Wesley Longman Publishing Co., Inc., (Boston).
- [4] C. M. Zapata, A Gelbukh and F. Arango. 2006. Pre-conceptual Schema: A Conceptual-Graph-Like Knowledge Representation for

- Requirements Elicitation. Lecture Notes in Computer Science, 4293, 17–27.
- [5] A. Baker, E. Navarro and A. Van der Hoek. 2005. An experimental card game for teaching software engineering processes. *The Journal of Systems and Software*, (75), 3–16.
  - [6] C. Britton y S. Jones. 1999. The untrained eye: How languages for Software Specification support understanding in untrained users. *Human Computer Interaction*, 14, 191–244.
  - [7] C. M. Zapata y G. Awad. 2007. Requirements Game: Teaching Software Project Management. *CLEI Electronic Journal*, 10(1). Retrieved from <http://www.clei.cl/cleiej/paper.php?id=133>.
  - [8] C. M. Zapata y M. Duarte. 2008. El juego de la consistencia: una estrategia didáctica para la Ingeniería de Software. *Revista Técnica de Ingeniería de la Universidad del Zulia*, 31(1), 1–10.
  - [9] C. M. Zapata. 2009. Teaching Software development by means of a classroom game: the software development game. In *Proceedings of the 36th annual ABSEL conference*, (Seattle, March 18-20), Bernie Keys CD ROM Library.
  - [10] M. Christel y K. Kang. 1992. Issues in Requirements elicitation. Technical Report CMU/SEI-92-TR-012 ESC-TR-92-012, Software Engineering Institute.
  - [11] R. Sproat. 2000. *A Computational Theory of Writing Systems*. Cambridge University Press (Cambridge).
  - [12] D. Jurafsky y J. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall (New Jersey).
  - [13] B. Grosz y C. Sidner. 1986. Attention, intention, and the structure of discourse. *Computational Linguistics*, (12), 175–204.
  - [14] C. M. Zapata y N. Carmona, “Un modelo de diálogo para la educación de requisitos de software”, *Dyna*, 2009, por aparecer.
  - [15] B. Kitchenham, D. Budgen., P. Brereton y Ph. Woodall. 2003. An investigation of software engineering curricula. *The Journal of Systems and Software*, 74, 325–335.
  - [16] B. Boehm. 2006. A View of 20th and 21st Century Software Engineering. Proceeding of the 28th international conference on Software engineering (Shanghai, May 20–28), ACM Press, New York, 12–29.