

# Preparando a los futuros desarrolladores de software: una metodología que apoya en la elaboración de la Especificación de Requerimientos.

María C. GÓMEZ

Departamento de Matemáticas Aplicadas y Sistemas,  
Universidad Autónoma Metropolitana  
Artificios 40 Col. Hidalgo, Álvaro Obregón D.F., 01120 México.

## RESUMEN

La enseñanza del Proceso de Análisis de Requerimientos a los estudiantes de licenciaturas relacionadas con la Ingeniería de Software es, aparentemente, algo sencillo. Por lo general se imparten los principios básicos de la Ingeniería de Requerimientos en clases teóricas y, se solicita a los alumnos que elaboren los requerimientos de un proyecto en particular para que apliquen los conocimientos adquiridos. Si bien lo anterior es una metodología bastante razonable, en la práctica se presenta un problema fundamental, hay muy pocos ejemplos de requerimientos para sistemas de software en los libros especializados. Este trabajo parte de la hipótesis de que las clases teóricas y los ejemplos de los libros no son suficientes para que los alumnos puedan elaborar correctamente un documento de Especificación y sugiere una metodología que pretende contrarrestar este problema. Los resultados que se presentan indican que la estrategia funciona.

**Palabras Claves:** Especificación de Requerimientos, enseñanza de ingeniería de software.

## 1. INTRODUCCIÓN

Preparar ingenieros capaces de integrarse eficientemente a la industria del software es un reto para las universidades, ya que se requiere de una formación multidisciplinaria. El éxito del desarrollo de un sistema de software no depende solamente de las capacidades técnicas, también juegan un papel muy importante la capacidad para aplicar metodologías, para apegarse a los estándares y para trabajar en equipo. Debido a lo anterior, el mundo académico se ha preocupado por estudiar las competencias y habilidades que se deben fomentar en los futuros desarrolladores de software y las estrategias pedagógicas que pueden utilizarse para que sus experiencias de aprendizaje se apeguen a las técnicas, prácticas y modos de trabajo que exige el desarrollo de software de calidad a nivel industrial [1, 2, 3, 4, 5]. Este trabajo presenta una nueva metodología que contribuye a la mejor preparación de los futuros ingenieros de software para que lleven a cabo correctamente una de las tareas fundamentales en el proceso de desarrollo de software: la especificación de los requerimientos.

El proceso de desarrollo de software se conoce como ciclo de vida del software porque describe la vida de un producto de software; primero nace con los requerimientos, luego se lleva a cabo su implantación, que consiste en su diseño, codificación y pruebas, posteriormente el producto se entrega, y sigue

viviendo durante su utilización y mantenimiento, la vida del sistema de software termina cuando se deja de utilizar. Un modelo de desarrollo de software es una representación abstracta de un proceso del software. Cada modelo de desarrollo es llamado a veces paradigma del proceso. Hay una gran variedad de paradigmas o modelos de desarrollo de software y es difícil describir todos, cada libro que trata este tema [6, 7, 8, 9, 10], entre otros, elige los que considera más importantes y desafortunadamente las opiniones son muy diversas. Sin embargo, Sommerville [9], clasifica todos los procesos de desarrollo de software en tres modelos generales que no son descripciones definitivas de los procesos del software sino más bien, son abstracciones de los modelos que se pueden utilizar para desarrollar software:

1. *El modelo en cascada.* Considera las actividades fundamentales que son: especificación, desarrollo, validación y evolución y las representa como fases del proceso separadas y consecutivas.
2. *Modelo evolutivo.* Entrelaza las actividades de especificación, desarrollo y validación. Un sistema inicial se desarrolla rápidamente a partir de especificaciones abstractas. Éste se refina basándose en las peticiones del cliente para producir un sistema que satisfaga sus necesidades.
3. *Modelo de componentes reutilizables.* Se basa en la existencia de un número significativo de componentes reutilizables. El proceso de desarrollo del sistema se enfoca en integrar estos componentes en el sistema en lugar de desarrollarlos desde cero.

Estos tres paradigmas o modelos de procesos genéricos se utilizan ampliamente en la práctica actual de la ingeniería del software. No se excluyen mutuamente y a menudo se utilizan juntos, especialmente para el desarrollo de sistemas grandes. Independientemente del modelo que se elija, siempre se presentará un reto fundamental: *el análisis de los requerimientos y la elaboración de la Especificación del sistema de software a desarrollar.*

## 2. LA IMPORTANCIA DE LOS REQUERIMIENTOS.

Los requerimientos especifican qué es lo que un sistema de software debe hacer, sus propiedades básicas y las deseables. La tarea de recolectar los requerimientos tiene como objetivo principal comprender lo que los clientes y los usuarios esperan

que haga el sistema. Norris y Rigby [10] plantean en su libro lo siguiente: “La captura y el análisis de los requerimientos del sistema es una de las fases más importantes para que un proyecto de software tenga éxito. Como regla de modo empírico, el costo de reparar un error se incrementa en un factor de diez de una fase de desarrollo a la siguiente, por lo tanto la preparación de una especificación adecuada de requerimientos reduce los costos y el riesgo general asociado con el desarrollo”. Este planteamiento pone en claro la gravedad de hacer un análisis de requerimientos defectuoso o un mal documento de Especificación.

Ya que el análisis de requerimientos es una tarea esencial en cualquier proceso de desarrollo de software, debe prestarse especial importancia a su enseñanza, para que así los futuros ingenieros de software estén mejor capacitados para afrontar los retos de su trabajo.

### 3. DESCRIPCIÓN DE LA METODOLOGÍA PROPUESTA

La metodología expuesta en este trabajo tiene dos propósitos, el primero, es generar una dinámica en la que los alumnos se vean en la necesidad de investigar lo que el cliente requiere; y el segundo es proveerlos con ejemplos de Especificaciones de Requerimientos de tal forma que tengan la oportunidad de practicar, y que lo visto en la clase teórica no quede tan abstracto.

La metodología consta de tres fases, la primera fase, llamada **Entrevistas**, consiste en que los alumnos tengan una entrevista con el profesor para que de esta manera extraigan la mayor cantidad de requerimientos posibles de un sistema en particular. Se propone utilizar proyectos de software ya elaborados por alumnos de generaciones anteriores para trabajar únicamente en sus requisitos. Se forman equipos, cada uno tiene una cita con el profesor a una hora diferente durante el horario de clase. Los alumnos deben preparar su entrevista basándose en los conocimientos adquiridos en clase teórica. Se puede generar expectación si no se les menciona cual será el sistema del que deberán extraer los requerimientos. La idea es que el profesor no dé más información de la que le solicite cada equipo.

El objetivo que se persigue con esta entrevista es lograr que los alumnos se den cuenta del grado de dificultad que representa la extracción de requerimientos en la vida real, ya que a lo largo de su vida académica han estado condicionados a hacer tareas bien definidas por un profesor y la tendencia es a seguir instrucciones en lugar de descubrir necesidades. Normalmente este tipo de entrevistas motiva a los alumnos a poner más atención a las clases teóricas. Es una situación común el que los alumnos den poca importancia a los requerimientos, ya que muchos piensan que esta fase debe ser superada rápidamente para comenzar cuanto antes con la implementación del sistema. El inicio temprano de las entrevistas durante el curso permite a los alumnos visualizar que en un futuro cercano se tendrán que enfrentar a este tipo de situaciones y, darse cuenta de que la extracción de requerimientos no es un paso que deba apresurarse o tomarse a la ligera. Las entrevistas también hacen salir a la luz ciertos aspectos que si solo se mencionan en clase no son tan evidentes, por ejemplo: seguir el protocolo de presentación y despedida, pedir permiso para grabar la entrevista, controlar los

nervios cuando la personalidad del entrevistador lo requiera, preparar las preguntas lo mejor posible para obtener la mayor cantidad de información, etc.

Después de la entrevista, se solicita a los alumnos una especificación de los requerimientos obtenidos durante ésta y se procede a la segunda fase llamada **Revisión de los requerimientos**. En esta etapa los equipos intercambian sus trabajos durante la clase y anotan sus observaciones. El objetivo principal de esta etapa es promover y poner en práctica el trabajo de revisión en pares, el cual es indispensable en el proceso de Aseguramiento de la Calidad [11, 12], los estudiantes deben aprender a hacer y a aceptar críticas constructivas que ayuden a mejorar la calidad de su trabajo [4].

La lectura de las especificaciones de los diferentes equipos es interesante tanto para los alumnos como para el profesor, ya que todos aprenden de los aspectos que unos tuvieron en cuenta y que a otros les faltó contemplar. En las carreras universitarias relacionadas con la ingeniería de software es muy común que existan algunos alumnos con algo de experiencia profesional, ellos pueden aportar mucho al grupo. Si se repiten estas entrevistas a lo largo del curso, los alumnos podrán ir poniendo en práctica lo aprendido en sus experiencias anteriores y constatar su avance.

Para cerrar la segunda fase, se regresa a los alumnos sus trabajos con los comentarios tanto de los demás equipos como los del profesor. Puede ser de gran ayuda proporcionar a los alumnos un resultado global del grupo, indicando cuales fueron las fallas generales y también sus aciertos.

En la tercera fase, llamada **Uso del Estándar**, el profesor proporciona a los alumnos un documento de Especificación incompleto elaborado con el Estándar IEEE-830 [13], dicho documento corresponde al sistema que previamente trabajaron en la fase de Entrevistas y de Revisión de Requerimientos. Esta tercera fase tiene dos propósitos, el primero es la familiarización con el Estándar IEEE-830 y el segundo es proveer ejemplos concretos de un documento de Especificación. Al comenzar esta fase, los alumnos ya tienen bastante idea de lo que requiere el sistema que se está trabajando, sin embargo, se recomienda dejar las secciones *1. Introducción* y *2. Descripción Global de la Especificación* con la mayor parte de su contenido final para que los alumnos comprendan mejor el sistema en cuya Especificación están trabajando. Se recomienda que la fase de *Uso del Estándar* se lleve a cabo en el laboratorio, de esta manera se garantiza que cada equipo esta haciendo su trabajo sin copiar. La versión del documento de especificación que reciben los equipos contiene los requerimientos específicos únicamente listados pero sin indicador de la sección que les corresponde. Cada equipo trabaja colocando los requerimientos en la sección donde ellos creen más conveniente. Para las primeras prácticas se define una de las 8 plantillas de la IEEE-830 y los alumnos tendrán que identificar cual fue la plantilla que se usó, en prácticas posteriores ellos mismos deberán elegir la plantilla que mejor describa al sistema en particular. Al final de la práctica, una vez que los alumnos ya enviaron su trabajo al profesor, éste les envía sus observaciones y les proporciona el documento de Especificación completo en el Estándar IEEE-830 para que verifiquen su resultado.

Si bien la bibliografía donde se puede consultar el proceso de análisis de requerimientos es excelente [6, 7, 9], hay una

carencia de ejemplos de documentos completos de Especificación de Requerimientos. Al enviar el documento de Especificación se provee a los alumnos con un ejemplo de requerimientos redactados de tal forma que cumplan con las características deseables de un requerimiento, es decir, que sea correcto, completo, consistente, realista, verificable, modificable y rastreable [6, 9, 13].

Los alumnos pueden hacer uso del lenguaje UML [14] para modelar los requerimientos, los ejemplos proporcionados en la tercera etapa también tienen la finalidad de ser ilustrativos al respecto.

En la siguiente sección se muestra un ejemplo del material que se utiliza para aplicar esta metodología.

Anaya [1] señala que el enfoque constructivista del conocimiento visualiza el aprendizaje como un proceso en espiral, en el que se van trabajando periódicamente los mismos contenidos, cada vez con mayor profundidad, con el fin de que el estudiante continuamente modifique las representaciones mentales que se ha ido construyendo. La aplicación repetida de esta metodología al análisis de los requerimientos de diferentes sistemas de software, permite reforzar y profundizar el conocimiento de la elaboración de documentos de Especificación de Requerimientos conforme al estándar IEEE-830.

## 4. MATERIAL DE EJEMPLO

A continuación se muestra la Especificación de Requerimientos para el caso de estudio de un “Sistema Visualizador de Moléculas”. Los alumnos parten de lo siguiente:

**Planteamiento del problema:** a continuación se presentan los primeros puntos de un ERS (Especificación de Requerimientos de Software) para un Visualizador de Moléculas. Algunos puntos de la IEEE-830 se excluyeron de este ejercicio (1.4, 1.5, 2.4, 2.5).

### 1.- Introducción

#### 1.1 Propósito

Una empresa biotecnológica le ha solicitado al grupo de ingenieros de software de la universidad desarrollar un visualizador molecular similar a Ras Mol (<http://www.openrasmol.org/>) para que sus científicos puedan trabajar en el diseño de moléculas.

Se espera que la aplicación desarrolladora pueda leer archivos en el formato XYZ (<http://openbabel.sourceforge.net/wiki/XYZ>) y muestre la molécula representada, permitiendo cambiar la vista, es decir, que permita observarla desde distintas direcciones, se tiene contemplado que en versiones futuras también sea posible dibujar una molécula en tercera dimensión.

#### 1.2 Alcance

- El sistema a desarrollar se llama “Visualizador Molecular” y tiene una interfaz gráfica que permite observar moléculas tridimensionales desde diferentes ángulos y también dibujarlas.
- Será posible convertir una vista de la imagen tridimensional a una imagen plana.

- También deberá ser posible observar una imagen bidimensional.
- El “Visualizador Molecular” sólo servirá para visualizar las moléculas.
- Solamente se podrá abrir un archivo en formato XYZ a la vez.

### 1.3.- Definiciones, siglas y abreviaturas

**Archivo XYZ:** Es un archivo que se forma de cuatro columnas en la primera columna representa el nombre del átomo y las otras tres columnas son las coordenadas las cuales especifican la posición del átomo en el espacio. Para representar una molécula se usan varios átomos.

**Renderización:** En términos de visualización en una computadora en 3D, es un proceso de cálculo complejo, desarrollado por la computadora para generar una imagen 2D a partir de una escena 3D. En otras palabras, la computadora interpreta la escena en 3 dimensiones y la plasma en una imagen bidimensional.

**Vista seleccionada:** es la forma en la que se ve la molécula una vez que el usuario la rotó y/o la trasladó.

## 2 Descripción Global

### 2.1 Perspectiva del Producto

El “Visualizador Molecular” es un sistema autónomo que consta de una interfaz gráfica en la que se pueden hacer diversas operaciones con la imagen de una molécula. Perspectivas del “Visualizador molecular”:

Interfaces del Sistema: la Interfaz Gráfica debe contener cuatro pestañas que indican los 4 modos de operación del sistema: (i) visualizar una molécula en 3D, (ii) visualizar una molécula en 2D, (iii) ver un archivo XYZ (iv) dibujar molécula.

### 2.2 Funciones del Producto

El “Visualizador Molecular” debe ser capaz de:

**1.- Visualizar una molécula en 3D:** Interpretar un archivo XYZ para mostrar en pantalla un modelo 3D de una molécula.

**2.- Rotar y trasladar la imagen de la molécula:** Una vez que la molécula se dibuja en la pantalla deberá ser posible manipular la imagen de tal forma que esta se pueda rotar y mover de un lado a otro.

**3.- Hacer zoom en la molécula presentada en 3D** Se requiere desarrollar una herramienta que permita hacer más grande la imagen en 3D y restaurarla.

**4.- Guardar la imagen 2D de la molécula a partir de la vista seleccionada por el usuario:** se requiere de una herramienta que convierta a 2D la vista de la imagen 3D que seleccionó el usuario.

**5.- Visualizar una molécula en 2D:** mostrar una imagen plana, que fue previamente renderizada, esto puede hacerse cuando se elige el modo (ii).

**6.- Ver un archivo XYZ:** ver el contenido de un archivo que está en este formato, esto puede hacerse cuando se elige el modo (iii).

**7.- Dibujar molécula:** Hacer el dibujo de una molécula en 3D a partir de un dibujo anterior o bien comenzando desde el principio y guardarlo en un archivo.

### 2.3.- Características del Usuario (completar)

## 2.6.- Prorratear los requisitos

(completar)

### Ejercicio:

- 1.- ¿Cual es la plantilla de la IEEE830 que debe usarse para especificar este sistema?
- 2.- Organizar los requerimientos que se presentan a continuación conforme a la plantilla correspondiente.

Nota 1: se omiten los siguientes puntos: 3.3 Requisitos de desarrollo, 3.4 Restricciones de diseño y 3.5 Atributos del software del sistema.

Nota 2: Omitir los incisos del estándar IEEE-830 que no se mencionen en los requerimientos de este ejercicio.

3.- En la sección "1.2 Alcance" hay un requerimiento que no es consistente. ¿Cuál es y como lo solucionarías?

### Requerimientos

- ❖ Este producto estará destinado a científicos con conocimientos de química, en específico, con conocimientos sobre el estudio y desarrollo de moléculas.
- ❖ El requerimiento de "dibujar una molécula" podrá ser pospuesto para una versión futura del sistema. Para la primera versión solo se requiere poder visualizar y renderizar imágenes 3D a 2D, así como poder ver un archivo en formato XYZ.
- ❖ El usuario podrá elegir las siguientes opciones dentro de cada uno de los modos:
  - (i) Visualizar una molécula en 3D
    - Cargar archivo XYZ
    - Renderizar la vista
    - Zoom
  - (ii) Visualizar una molécula en 2D,
    - Cargar archivo renderizado
  - (iii) Ver un archivo XYZ
    - Cargar archivo XYZ
  - (iv) Dibujar molécula
    - Cargar dibujo
    - Guardar dibujo.
- ❖ Mediante el uso del ratón el usuario podrá interactuar con el "Visualizador Molecular" para rotar y trasladar la imagen. También podrá usar el teclado para nombrar el archivo JPG con la imagen renderizada. Para acceder a las diferentes opciones y características que ofrece el sistema podrá usar tanto el ratón como el teclado.
- ❖ El "Visualizador Molecular" trabajará sobre un sistema operativo Windows XP, un Windows Vista, o sobre Windows 7. Para desarrollarlo se utilizará Java y la librería 3D de java.
- ❖ El sistema opera con los siguientes modos:

**Modo de visualización 3D**

  - 1 Debe existir por lo menos un archivo XYZ en la maquina donde opera el sistema.
  - 2 El "Visualizador Molecular" debe poder interpretar un archivo XYZ y desplegar la molécula en pantalla.
  - 3 La imagen debe poder ser renderizada en una posición elegida por el usuario.
  - 4 Una vez que la imagen ha sido renderizada deberá guardarse en un archivo como imagen JPG.

**Modo de Visualización 2D**

  - 1 Mostrar en la pantalla la imagen de un archivo JPG que debe ser la imagen renderizada de una figura 3D.

**Ver un archivo XYZ**

  - 1 Mostrar en la pantalla el formato de un archivo XYZ seleccionado por el usuario.

## Dibujar molécula en 3D

- 1 Se debe poder cargar la imagen de un dibujo previo.
- 2 También se puede comenzar a hacer un dibujo desde el principio.
- 3 El dibujo de la molécula se traducirá a formato XYZ para poder guardarlo en un archivo.

A partir de lo anterior los alumnos deberían haber resuelto lo siguiente.

### Respuesta al ejercicio

1.- La plantilla A1 del estándar IEEE-830, estructurada por el modo, se adapta muy bien a este caso.

2.- La primera parte "1.-Introducción" no tiene cambios, excepto el señalado en la respuesta 3. La parte "2. Descripción Global" tampoco, excepto 2.2 y 2.3 que en el ejercicio estaban vacíos, su contenido debe ser:

### 2.3.- Características del Usuario

Este producto estará destinado a científicos con conocimientos de química, en específico sobre el estudio y desarrollo de moléculas.

### 2.6.- Prorratear los requisitos

El requerimiento de "dibujar una molécula" podrá ser pospuesto para una versión futura del sistema. Para la primera versión solo se requiere poder visualizar y renderizar imágenes 3D a 2D, así como poder ver un archivo en formato XYZ.

3.- En la sección "1.2 Alcance" hay un requerimiento que no es consistente. ¿Cuál es y como lo solucionarías?

El requerimiento:

- ❖ El "Visualizador Molecular" solo servirá para visualizar las moléculas.
- No es consistente con el requerimiento de que también se podrá dibujar una molécula. Para eliminar la inconsistencia se puede redactar como:
- ❖ En su primera fase, el "Visualizador Molecular" solo servirá para visualizar las moléculas.

La sección 3 de la Especificación es en donde se definen los requerimientos específicos. La respuesta que se presenta a continuación muestra la aplicación de la plantilla A1 para ordenar los requerimientos proporcionados en el ejercicio.

### 3.- Requerimientos Específicos:

#### 3.1 Requisitos de la Interfaz Externa:

El sistema consta de 4 tareas básicas:

- Visualización de una molécula en 3D en pantalla.
- Visualización de una molécula en 2D en pantalla.
- Mostrar un archivo en formato XYZ.
- Dibujar molécula en 3D.

#### 3.1.1 Interfaz con el Usuario

El usuario podrá elegir las siguientes opciones dentro de cada uno de los modos:

- (i) Visualizar una molécula en 3D
  - Cargar archivo XYZ
  - Renderizar la vista
  - Zoom
- (ii) Visualizar una molécula en 2D
  - Cargar archivo renderizado

- (iii) Ver un archivo XYZ
  - Cargar archivo XYZ
- (iv) Dibujar molécula
  - Cargar dibujo.
  - Guardar dibujo.

### 3.1.2 Requisitos de la interfaz con el hardware

Mediante el uso del ratón el usuario podrá interactuar con el “Visualizador Molecular” para rotar y trasladar la imagen. También podrá usar el teclado para nombrar el archivo JPG con la imagen renderizada. Para acceder a las diferentes opciones y características que ofrece el sistema podrá usar tanto el ratón como el teclado.

### 3.1.3 Requisitos de la interfaz con el software

El “Visualizador Molecular” trabajará sobre el sistema operativo Windows XP, Windows Vista y Windows 7 para desarrollarlo se utilizará Java con la librería 3D de java.

## 3.2 Requisitos funcionales

### 3.2.1 Modo de visualización 3D

**3.2.1.1** Debe existir por lo menos un archivo XYZ en la maquina donde opera el sistema.

**3.2.1.2** El “Visualizador Molecular” debe poder interpretar un archivo XYZ y desplegar la molécula en pantalla.

**3.2.1.3** La imagen debe poder ser renderizada en una posición elegida por el usuario.

**3.2.1.4** Una vez que la imagen ha sido renderizada deberá guardarse en un archivo como imagen JPG.

### 3.2.2 Modo de Visualización 2D

**3.2.2.1** Mostrar en la pantalla la imagen de un archivo JPG que debe ser la imagen renderizada de una figura 3D.

### 3.2.3 Ver un archivo XYZ

**3.2.3.1** Mostrar en la pantalla el formato de un archivo XYZ seleccionado por el usuario.

### 3.2.3 Ver un archivo XYZ

### 3.2.4 Dibujar molécula en 3D

**3.2.4.1** Se debe poder cargar la imagen de un dibujo previo.

**3.2.4.2** También se puede comenzar a hacer un dibujo desde el principio.

**3.2.4.3** El dibujo de la molécula se traducirá a formato XYZ para poder guardarlo en un archivo.

En los grupos de 2010 y 2011 se observó que los primeros requerimientos de los equipos tuvieron muchas carencias. Al terminar la fase de *Revisión de los requerimientos* los alumnos hicieron observaciones interesantes a los trabajos de los demás equipos y también identificaron los puntos incluidos en otros trabajos que no se habían tomado en cuenta en el trabajo propio.

Las carencias típicas de los primeros trabajos fueron: requerimientos específicos muy detallados con falta de una descripción global del sistema, o viceversa, una buena descripción global descuidando los requerimientos específicos, algunos incluyeron información no técnica propia de otro tipo de documentos, como la justificación o la planeación del proyecto. Y prácticamente todos incluyeron requerimientos no verificables. Posteriormente, al realizar un segundo ciclo de la metodología, se notó un cambio sustancial en los trabajos, la mayoría con secciones bien definidas, con una visión global del sistema y otra sección con los requerimientos específicos. En la etapa de *Revisión de los Requerimientos* los alumnos lograron identificar la información que estaba fuera de lugar en los trabajos de sus compañeros.

Cuando los alumnos llevaron a cabo la tercera etapa: *Uso del Estándar* por primera vez, el 100% de los trabajos estuvo incorrecto, lo que comprueba la hipótesis de que la clase teórica no es suficiente para que los alumnos puedan elaborar correctamente el documento de Especificación de Requerimientos, sin embargo, después de que los alumnos lo intentaron una vez y pudieron ver el ejemplo correcto proporcionado por el profesor, se obtuvieron los siguientes resultados en el grupo 2010: el 20% del grupo logró elegir la plantilla correcta y colocar todos los requerimientos en la sección correspondiente, el 60% restante no lo logró por completo pero tuvo una buena cantidad de aciertos, y solo el 20% no lo logró. En el grupo 2011: 20% logró elegir la plantilla correcta y colocar todos los requerimientos en la sección correspondiente, el 70% no lo logró pero tuvo una cantidad de aciertos razonable y solo el 10% no lo logró.

Al realizar todo el proceso en repetidas ocasiones pudo observarse que tanto los alumnos del grupo 2010 como los del 2011 no solo se familiarizaron con la elaboración de requerimientos conforme al estándar IEEE-830, sino que también estuvieron en contacto con ejemplos de lo que debe ser el contenido de un documento de especificación y con las características que deben tener los requerimientos, de tal forma que no tuvieron problemas para elaborar sus trabajos finales.

## 5. RESULTADOS

Se evaluó la calidad de los trabajos finales de Especificación de Requerimientos de un sistema de software presentados por los alumnos de los grupos 2008 y 2009 y se compararon los resultados con los trabajos finales presentados por los alumnos de los grupos 2010 y 2011 con los cuales se trabajó la nueva metodología propuesta. La diferencia en la calidad de los trabajos debida al nivel académico promedio de cada uno de los grupos no fue decisiva, pues se observó una mejora notable en las Especificaciones presentadas por alumnos “promedio” de los grupos de 2010 y 2011 con respecto a las Especificaciones presentadas por alumnos de alto rendimiento de los grupos anteriores (2008 y 2009).

## 6. CONCLUSIONES

El propósito de este trabajo es contribuir en la innovación y actualización de metodologías para la conducción del proceso de enseñanza-aprendizaje de la Ingeniería de Software, en específico del proceso de análisis y especificación de requerimientos. El crecimiento exponencial de los desarrollos de sistemas de software que ha caracterizado a estas últimas décadas, debe apoyarse con una formación sólida de los nuevos ingenieros en computación, quienes se enfrentarán al reto de extraer e interpretar correctamente los requerimientos de los sistemas que desarrollarán y plasmarlos correctamente en un documento de Especificación.

El uso de la metodología aquí propuesta es prometedor y es aplicable a todas las licenciaturas en las que se estudie el tema

del Desarrollo de Software. Su correcta aplicación provee a los estudiantes con ejemplos concretos de Especificación de Sistemas de Software y les proporciona retroalimentación a su trabajo.

## 7. TRABAJO FUTURO

En el futuro se pueden agregar entrevistas con otros profesores o incluso, cuando sea posible, promover que los alumnos se enfrenten con usuarios reales aplicando las fases de *Entrevistas* y de *Revisión de los Requerimientos*. También se puede promover que los alumnos vayan cambiando de equipos a lo largo del curso para propiciar que trabajen todos los participantes.

## 8. REFERENCIAS

- [1] Anaya, R.: “Una visión de la enseñanza de la ingeniería de software como apoyo al mejoramiento de las empresas de software”. Red de Revistas Científicas de América Latina y el Caribe, España y Portugal (Redalyc) Universidad Eafit, Vol. 42, No. 141, pp. 60-76, Colombia, 2006.
- [2] Ahumada, E., Flores, M. y Zárate R.: “Simulación Empresarial aplicada a la Enseñanza de Ingeniería de Software y de Mercadotecnia como Actividades Multidisciplinarias”, Simposium Iberoamericano en Educación, Cibernética e Informática, Orlando Florida, 2009.
- [3] Liu, J., Marsaglia J, and Olson, D.: “Teaching software engineering to make students ready to real world”. Journal of Computing Science in College, Vol. 18, No. 2, pp. 43-50, 2002.
- [4] Bracken, B.: “Progressing from student to professional: The importance and challenges of teaching software engineering”. Journal of Computing Sciences in College, Vol. 19, No. 2, pp.358-368, 2003.
- [5] Cockbaine J. y Álvarez J., “La Enseñanza y Aprendizaje de la Formulación y Evaluación de Proyectos Tecnológicos en Carreras de Ingeniería Informática: Una Experiencia Práctica” SIECI 2010, 29 de junio al 2 de julio, Orlando, Florida 2010.
- [6] Pfleeger, Shari L.: “Ingeniería de software. Teoría y práctica” 1ª edición. Editorial Pearson Education, Buenos Aires, 2002.
- [7] Pressman, R. S.: “Ingeniería del Software: Un enfoque práctico”, 5a edición. Editorial McGraw Hill, España, 2002.
- [8] Weitzenfeld, A.: “Ingeniería de Software Orientada a Objetos con UML, Java e Internet”. Editorial Thomson, 2004.
- [9] Sommerville, I.: “Ingeniería del Software”, 7ª Ed., Pearson Addison Wesley, Madrid, 2005.
- [10] Norris and Rigby: “Ingeniería de software explicada”, 1ª edición. Editorial Megabyte-Noriega editores, México, 1994.
- [11] IEEE STD-730 “Standard for Software Quality Assurance Plans”, IEEE Computer Society, 2002.
- [12] Gilb T. and Graham D.: “Software Inspection”, Ed. Addison-Wesley, 4ª ed., England, 1998.
- [13] IEEE-STD-830-1998: Recommended Practice for Software Requirements Specifications.
- [14] Cheesman, J. and Daniels J.: “UML Components. A simple process for Specifying Component-Based Software”, Addison Wesley, 2001.