

## La ingeniería de requisitos en el desarrollo de aplicaciones informáticas

### The requirements engineering in the development of informatics applications

Ing. Jorge Reyes Estévez<sup>1\*</sup> 0000-0003-0619-6465

<sup>1</sup>Empresa Comercializadora y Distribuidora de Medicamentos (EMCOMED). OSDE BioCubaFarma.

\*Correspondencia: [yreyese@infomed.sld.cu](mailto:yreyese@infomed.sld.cu)

#### RESUMEN

**Introducción:** La ingeniería de software abarca la obtención de los requerimientos o requisitos del software, el diseño del sistema, la implementación, las pruebas, la instalación, el mantenimiento y la actualización del sistema. La ingeniería de requisitos se enfoca en la definición de lo que se desea producir.

**Objetivo:** Describir las principales características de la ingeniería de requisitos y resaltar su importancia dentro del proceso de desarrollo de software.

**Método:** Revisión documental clásica con ayuda de las palabras clave en sitios de Internet. Análisis crítico de los artículos, estructurando el desarrollo del trabajo en tres secciones: requisitos de software, ingeniería de requisitos y técnicas principales para obtener los requisitos.

**Desarrollo:** La ingeniería de requisitos consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, las necesidades de los clientes para la producción de una nueva aplicación informática o la modificación de una existente; su fin es minimizar los problemas relacionados con la deficiente gestión de los requerimientos, lo que puede aumentar los costos y en muchos casos conducir al fracaso de los proyectos. Se exponen y discuten la generación de especificaciones de requisitos, la Ingeniería de requisitos en sí, así como las técnicas principales para obtener requisitos.

**Conclusiones:** La ingeniería de requisitos abarca las actividades de obtención, análisis, validación y documentación de las especificaciones de requisitos, su adecuada gestión permite optimizar el inicio de la fase de diseño, con vistas a que el producto final refleje las reales necesidades del usuario.

**Palabras clave:** requisitos; requerimientos; ingeniería de requisitos; desarrollo de software.

---

<http://scielo.sld.cu>



Este documento está bajo [Licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/).

## ABSTRACT

**Introduction:** Software engineering consists in obtaining software requirements, system designing and implementation, tests, installing, maintaining and actualizing the system. Requirement engineering focus on defining the customer needed specifications to perform by the software system.

**Objective:** Describe the main characteristics of requirements engineering and highlight its importance within the software development process.

**Methods:** A classical documentary review was carried out by selecting publications with the help of keywords on Internet sites. A critical analysis of the articles was carried out. The development of the exposition was structured in three sections.

**Development:** The execution of the process of generating correct specifications that clearly, unambiguously, consistently and compactly describe the needs of users or customers to minimize the problems related to mismanagement of requirements in systems development, it is the center of requirements engineering. There are exposed as well as discussed the concepts related with the subject. From the point of view of the documental analysis the issue is presented by the description and discussion of the process of generating requirements specifications, requirements engineering and the main techniques used to obtain requirements.

**Conclusions:** Requirements engineering involves all the activities related to obtaining, analyzing, validating and documenting the requirements specifications, so its proper management allows reaching an optimal state at the beginning of the software design phase so that the final product reflects the real needs of the user.

**Keywords:** Requirements; requirement's engineering; user software development.

Recibido: 7/4/2020

Aprobado: 3/7/2020

## Introducción

La producción de software es un proceso intrínsecamente creativo y la ingeniería de software trata de sistematizarlo con el fin de reducir el riesgo del fracaso en la consecución del objetivo, por medio de diversas técnicas que se han demostrado efectivas sobre la base de la experiencia precedente.

El proceso de producción de software puede describirse sintéticamente como la obtención de los requisitos o requerimientos del software, el diseño del sistema de software, la implementación, las pruebas, la instalación, el mantenimiento y la actualización del sistema.

---

<http://scielo.sld.cu>



Este documento está bajo [Licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/).

La ingeniería de requisitos tiene una función básica en el proceso de producción de un software, ya que se enfoca en la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones de requisitos correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, las necesidades de los usuarios finales y los clientes.

A través de los años se ha podido constatar que los requisitos son una pieza fundamental en un proyecto de desarrollo de software, ya que marcan el punto de partida para actividades como la planeación, básicamente en lo que se refiere a las estimaciones de tiempos y costos, así como la definición de recursos necesarios y la elaboración de cronogramas que será uno de los principales mecanismos de control durante la etapa de desarrollo. Además, la especificación de requisitos es la base que permite verificar si se alcanzan o no los objetivos establecidos en el proyecto, ya que son un reflejo detallado de las necesidades de los clientes y usuarios finales del sistema y es contra lo que se va a estar verificando si se están cumpliendo las metas trazadas.

“El tratamiento de requisitos es el proceso mediante el cual se especifican y validan los servicios que debe proporcionar el sistema, así como las restricciones sobre las que se deberá operar. Consiste en un proceso iterativo y cooperativo de análisis del problema, documentando los resultados en una variedad de formatos y probando la exactitud del conocimiento adquirido” <sup>(1)</sup>. La importancia de esta fase es esencial puesto que los errores más comunes y más costosos de reparar, así como los que más tiempo consumen se deben a una inadecuada ingeniería de requisitos.

Es muy frecuente escuchar de la existencia de muchos problemas en el desarrollo de software, como el inadecuado entendimiento de las necesidades de los usuarios finales e insatisfacciones de los clientes, por inaceptable o bajo desempeño del software. Muchas de las causas se pueden hallar en inadecuada definición, especificación, y administración de los requisitos.

El presente trabajo tiene como objetivo describir las principales características de la ingeniería de requisitos resaltando su importancia dentro del proceso de desarrollo de software.

## Método

Se realizó una revisión documental clásica que recuperó información acerca del papel de la ingeniería de requisitos en el proceso de desarrollo de software. Por la naturaleza del tema, se utilizaron las palabras claves definidas en esta investigación para realizar la búsqueda y selección de publicaciones y documentos disponibles en Internet, fundamentalmente en las siguientes fuentes: Repositorios de Universidades de Latinoamérica y España, Revistas de Congresos y Biblioteca Electrónica en Línea Scielo.

---

<http://scielo.sld.cu>



Este documento está bajo [Licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/).

Se analizó de forma crítica el material seleccionado y a partir de esta información y a criterio del autor, se sustentaron de manera teórica los factores que componen la estructura del trabajo al considerar las siguientes secciones: requisitos de software, ingeniería de requisitos y técnicas principales para obtener requisitos.

## Desarrollo

Los requisitos representan una parte fundamental en la consecución de la calidad del software, razón por la cual se justifica la existencia de la denominada Ingeniería de Requisitos (IR).

“La IR trata de establecer lo que el sistema debe hacer, sus propiedades emergentes deseadas y esenciales y las restricciones en el funcionamiento del sistema y los procesos de desarrollo del software. No es simplemente un proceso técnico sino también social, pues los requerimientos del sistema están influenciados por las preferencias, aversiones y prejuicios de los usuarios y por cuestiones políticas y organizacionales”<sup>(2)</sup>.

Si no se realiza un estudio previo de los requisitos del usuario, no se hace una definición completa del alcance del proyecto y no se realiza el modelado del negocio antes de desarrollar el software, el equipo desarrollador o analista no se involucra en el problema y aunque tenga claro que el sistema debe desarrollarse para dar soporte a los procesos de la organización, se corre el riesgo de que los requisitos identificados no correspondan a las necesidades para lo que se debe crear.

Ayala et al.<sup>(3)</sup> refieren que, como disciplina, la IR establece el proceso de definición de requisitos en una sucesión de actividades mediante las cuales lo que debe hacerse se educa, se modela y también se analiza.

Así pues, al entender que la IR es la fase de un proyecto software donde se definen las propiedades y la estructura del mismo; y que a la vez comprende el desarrollo y gestión de requisitos, se entiende también que debe realizarse de manera adecuada, pues omitir información en esta actividad ha provocado que muchos proyectos de software fracasen.

“Año tras año, el software se ha convertido en componente vital de nuestra sociedad. No existen sectores de negocio e iniciativas económicas, infraestructura industrial y civil, política, educación y entretenimiento, por nombrar unas pocas, que no estén profundamente permeadas y gobernadas por programas de aplicación y sistemas o aspectos de la vida diaria que no estén afectadas por el software”<sup>(4)</sup>. Consecuentemente, el desarrollo de software se ha convertido en una actividad crítica que necesita ser cuidadosamente estudiada, entendida y mejorada.

“Muchos de los proyectos de software fracasan al menos en forma parcial ya que son pocos los que cumplen con sus presupuestos de costo, planificación, criterios de

---

<http://scielo.sld.cu>



Este documento está bajo [Licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/).

calidad o una definición correcta y objetiva de especificaciones de requerimientos. Las causas del fracaso generalmente se detectan en autopsias que se realizan a los proyectos, cuando ya es muy tarde para cambiar de dirección”<sup>(5)</sup>.

En este sentido, el grupo Standish<sup>(6)</sup> emite en forma anual un “Reporte del Caos” desde el año 1995 en el cual sus últimas investigaciones se centran en identificar el alcance de los fracasos de proyectos de software, los factores que más inciden y los elementos claves a considerar para reducir estos fracasos. Estima que las compañías americanas y agencias del gobierno gastarán 81 billones de dólares por proyectos de software cancelados.

El Reporte del Caos 2015<sup>(7)</sup>, refiere que se estudiaron un total de 50.000 proyectos donde el 71% de los proyectos realizados fueron cancelados antes de ser implementados o finalizaron con problemas. A partir de una encuesta realizada, este grupo determinó que las principales causas del fracaso son requerimientos y especificaciones incompletas; cambios frecuentes en los requerimientos y especificaciones; expectativas no realistas y objetivos poco claros. Es decir, que una de las principales causas del fracaso está asociada al mal manejo de los requerimientos en aproximadamente el 41,7% de los proyectos, según esta misma encuesta.

Se estima que, “en un ciclo tradicional de desarrollo, el costo de corregir un error en los requerimientos crece en forma exponencial en relación con el crecimiento del tiempo de detección”<sup>(8)</sup>, por lo cual es primordial “realizar una buena gestión de los requerimientos y anticiparse a cualquier cambio en los requerimientos a fin de minimizar su impacto perjudicial sobre el proyecto”<sup>(9)</sup>.

## Requisitos del software

Según el Standard Glossary of Software Engineering Terminology de IEEE, un requisito es: “una condición o una capacidad necesitada por un usuario para solucionar un problema o lograr un objetivo” También considera que: “Una condición o una capacidad que debe tener o poseer un sistema o un componente de sistema para satisfacer un contrato, estándar, especificación, u otros documentos formalmente impuestos”<sup>(10)</sup>.

Por su parte, Ian Sommerville plantea que “los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema... El proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se denomina ingeniería de requerimientos”<sup>(2)</sup>.

“Un requerimiento es algo que el producto debe hacer o una cualidad que el producto debe tener. Un requerimiento existe, ya sea porque el tipo de producto demanda ciertas necesidades o cualidades, o porque el cliente desea que ese requerimiento sea parte del producto entregado”<sup>(11)</sup>.

Considerando las definiciones anteriores, un requisito es una descripción de una condición o capacidad que debe cumplir un sistema, derivada de una necesidad de

---

<http://scielo.sld.cu>



usuario identificada, o bien, estipulada en un contrato, procedimiento u otro tipo de documento formal establecido al inicio del proceso.

Cuando un requisito es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste, se considera un requisito de usuario. Por otro lado, cuando es una definición detallada y formal de una función del sistema, se denomina requisito de sistema.

De la misma forma, Sommerville refiere que: “Los requerimientos del usuario son declaraciones, en lenguaje natural y en diagramas, de los servicios que se espera que el sistema proporcione y de las restricciones bajo las cuales debe funcionar. Los requerimientos del sistema establecen con detalle las funciones, servicios y restricciones operativas del sistema” <sup>(2)</sup>.

Es común clasificar los requisitos en funcionales y no funcionales.

“Los requisitos funcionales son los que definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas” <sup>(12)</sup>. Es importante que se describa el ¿Qué? y no el ¿Cómo? se deben hacer esas transformaciones. Estos requerimientos al tiempo que avanza el proyecto de software se convierten en los algoritmos, la lógica y gran parte del código del sistema. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer.

Asimismo, Fernández <sup>(12)</sup> especifica que los requisitos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc. Son restricciones de los servicios o funciones ofrecidos por el sistema.

Coincidentemente, Sonia I. Marino et al. plantean que “los requerimientos funcionales describen lo que el sistema debe hacer y dependen del tipo de software a desarrollarse y los no funcionales, es decir las propiedades emergentes del sistema como la fiabilidad, el tiempo de respuesta, entre otros” <sup>(13)</sup>.

Wieggers & Beatty <sup>(9)</sup> refieren que un requisito funcional define una función del sistema de software o sus componentes y pueden ser: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone, un sistema debe cumplir. Y los no funcionales, especifican criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento.

En realidad, la distinción entre diferentes tipos de requerimientos no es tan clara como sugieren estas definiciones. Por ejemplo, un requerimiento del usuario sobre seguridad podría parecer un requerimiento no funcional. Sin embargo, cuando se desarrolla en detalle, puede generar otros requerimientos que son claramente funcionales, como la necesidad de incluir en el sistema recursos para la autenticación del usuario.

---

<http://scielo.sld.cu>



En la formulación de un requisito se deben cumplir un grupo de características básicas, entre las que están: <sup>(14)</sup>

1. Debe ser especificado por escrito, quedando como una evidencia de acuerdo entre las partes.
2. Necesario: Lo que solicite un requisito debe ser necesario para el producto.
3. No ambiguo: El texto debe ser claro, preciso y tener una única interpretación posible.
4. Posible de probar o verificar. Se debe poder verificar con absoluta certeza, si el requisito fue satisfecho o no. Esta verificación puede lograrse mediante inspección, análisis, demostración o testeo.
5. Conciso: Un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.
6. Consistente: Ningún requisito debe entrar en conflicto con otro requisito diferente, ni con parte de otro. Asimismo, el lenguaje empleado entre los distintos requisitos debe ser consistente también.
7. Completo: Los requisitos deben contener en sí mismos toda la información necesaria, y no remitir a otras fuentes externas que los expliquen con más detalle.
8. Alcanzable: Un requisito debe ser un objetivo realista, posible de ser alcanzado con el dinero, el tiempo y los recursos disponibles.

Según Boehm <sup>(15)</sup>, la especificación de requisitos debe ser completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en dónde se describen las funciones que realizará el sistema.

Estas características suelen ser subjetivas, es decir, no pueden ser calculadas de forma automática por ningún sistema. Por ello, se tiende a utilizar métricas o indicadores que sí pueden ser calculados de forma automática y que, de algún modo, pueden contribuir a ponderar las anteriores características.

## Ingeniería de requisitos

El proceso de recopilar, analizar y verificar las necesidades del cliente o usuario para un sistema es llamado ingeniería de requisitos. Según Sommerville <sup>(2)</sup>, la meta de la ingeniería de requisitos es crear, mantener y entregar una especificación de requisitos de software correcta y completa.

Por otra parte, Pressman <sup>(16)</sup> refiere que el espectro amplio de tareas y técnicas que llevan a entender los requerimientos se denomina ingeniería de requerimientos. Desde la perspectiva del proceso del software, la ingeniería de requerimientos es una de las acciones importantes de la ingeniería de software que comienza durante la actividad de comunicación y continúa en la de modelado. Debe adaptarse a las necesidades del proceso, del proyecto, del producto y de las personas que hacen el trabajo.

La ingeniería de requerimientos tiende un puente para el diseño y la construcción.

---

<http://scielo.sld.cu>



Este documento está bajo [Licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/).

“La ingeniería de requerimientos proporciona el mecanismo apropiado para entender lo que desea el cliente, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación y administrar los requerimientos a medida que se transforman en un sistema funcional”<sup>(16)</sup>.

El proceso general se puede desglosar en cuatro subprocesos de alto nivel.

Estudio de viabilidad. Trata de una evaluación de la posible utilidad del sistema para la organización, de la posibilidad de que se pueda implementar con las tecnologías disponibles y que pueda integrarse con otros sistemas existentes. Puede parecer obvio, pero en ocasiones un sistema no aporta nada al negocio pues no se consiguen definir correctamente los requisitos del negocio para el sistema o las propias entidades no tienen bien declarados sus objetivos.

Obtención y análisis de requisitos. En este subproceso, los ingenieros de software trabajan con los clientes y los usuarios finales del sistema para determinar el dominio de la aplicación, qué servicios deben proporcionar el sistema, el rendimiento requerido del sistema, las restricciones hardware, entre otros elementos. Se involucra a todas las personas que puedan afectarse de alguna manera con el sistema, a todos los interesados (stakeholder).

Según Arias<sup>(17)</sup>, en el trabajo con los stakeholders se pueden presentar inconvenientes para obtener y comprender los requisitos, los que son importantes de identificar y prevenir. A continuación, se presenta un listado con los más comunes en este proceso:

1. Los requerimientos no son obvios y vienen de muchas fuentes.
2. Son difíciles de expresar en palabras (el lenguaje es ambiguo).
3. La cantidad de requerimientos en un proyecto puede ser difícil de manejar.
4. Un requerimiento puede cambiar a lo largo del ciclo de desarrollo.
5. Diferentes stakeholders tienen diferentes requisitos, generándose conflictos.
6. El usuario no puede explicar lo que hace.
7. Tiende a recordar lo excepcional y olvidar lo rutinario.
8. Hablan de lo que no funciona
9. Los usuarios tienen distinto vocabulario que los desarrolladores.

Sobre este tema, Herrera<sup>(18)</sup> refiere que muchos sistemas tienen diferentes tipos de clientes. Cada grupo de clientes tiene necesidades diferentes, y diferentes requerimientos tienen diferentes grados de importancia para ellos. Por otro lado, escasas veces tenemos que los clientes son los mismos usuarios; trayendo como consecuencia que los clientes soliciten procesos que causan conflictos con los solicitados por el usuario.

En este subproceso se realizan un grupo de actividades cíclicas que son:

1. Descubrimiento de los requisitos en interacción con los stakeholders.
2. Clasificación y organización de los requisitos en grupos coherentes.

---

<http://scielo.sld.cu>





3. Ordenación de requisitos por prioridades y negociación de posibles conflictos entre requisitos provocados por diferentes stakeholders.
4. Documentar los requisitos ordenados y revisados.

Validación de requisitos. Se trata de mostrar que los requisitos obtenidos reflejan y definen el sistema que el usuario o cliente desea. Errores que persistan en los requisitos implican costos superiores en el desarrollo del proyecto. Lo que se verifica son precisamente las características que definen a los requisitos. Para la validación se pueden utilizar diferentes técnicas como el uso de prototipos, casos de prueba, etc.

Gestión de requisitos. Habitualmente los requisitos iniciales son incompletos, por lo que es normal que sean de naturaleza cambiante. Los requisitos van evolucionando según los stakeholders profundizan la comprensión del problema. También es un hecho cierto que siempre aparecen nuevas necesidades y prioridades y es necesario que los especialistas puedan gestionar estos cambios.

En síntesis, el proceso de ingeniería de requisitos se utiliza para definir todas las actividades involucradas en el descubrimiento, documentación y mantenimiento de los requisitos para un producto de software determinado, donde es muy importante tomar en cuenta que el aporte de la ingeniería de requisitos vendrá a ayudar a determinar la viabilidad de llevar a cabo el software (si es factible desarrollarlo o no), pasando posteriormente por un subproceso de obtención y análisis de requerimientos, su especificación formal, para finalizar con el subproceso de validación donde se verifica que los requerimientos realmente definen el sistema que quiere el cliente.

Los principales beneficios que se obtienen de la Ingeniería de Requerimientos son:

1. Permite gestionar las necesidades del proyecto en forma estructurada: Cada actividad de la IR consiste de una serie de pasos organizados y bien definidos.
2. Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados: La IR proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios.
3. Disminuye los costos y retrasos del proyecto: es sabido que reparar errores por un mal desarrollo no descubierto a tiempo, es sumamente caro; especialmente aquellas decisiones tomadas durante la IR, ya que es una de las etapas de mayor importancia en el ciclo de desarrollo de software y de las primeras en llevarse a cabo.
4. Mejora la calidad del software: La calidad en el software tiene que ver con cumplir un conjunto de requerimientos (funcionalidad, facilidad de uso, confiabilidad, desempeño, etc.).
5. Mejora la comunicación entre equipos: La especificación de requerimientos representa una forma de consenso entre clientes y desarrolladores. Si este consenso no ocurre, el proyecto no será exitoso.
6. Evita rechazos de usuarios finales: La ingeniería de requerimientos obliga al cliente a considerar sus requerimientos cuidadosamente y revisarlos dentro del

---

<http://scielo.sld.cu>



marco del problema, por lo que se le involucra durante todo el desarrollo del proyecto.<sup>(17)</sup>

Es necesario apuntar que no existe un proceso único que sea válido de aplicar en todas las organizaciones. Cada organización debe desarrollar su propio proceso de acuerdo al tipo de producto que se esté desarrollando, a la cultura organizacional, y al nivel de experiencia y habilidad de las personas involucradas en la ingeniería de requisitos. Hay muchas maneras de organizar el proceso de ingeniería de requisitos y en otras ocasiones se tiene la oportunidad de recurrir a consultores, ya que ellos tienen una perspectiva más objetiva que las personas involucradas en el proceso.

## Técnicas principales para obtener requisitos

En opinión de Herrera<sup>(18)</sup>, Guerra<sup>(19)</sup> y según se refiere en la WEB PMOinformatica.com<sup>(20)</sup>, existen varias técnicas que se utilizan para obtener los requisitos.

Es importante resaltar que estas técnicas pueden ser aplicables a las distintas fases del proceso de la ingeniería de requisitos, haciendo la salvedad de que hay que tomar en cuenta las características propias del proyecto en particular que se esté desarrollándose para aprovechar al máximo su utilidad.

Entrevistas y cuestionarios: El equipo de la ingeniería de requisitos hace preguntas a los stakeholders sobre el sistema que utilizan y sobre el sistema a desarrollar. Los requisitos provienen de las respuestas a estas preguntas. Las entrevistas pueden ser abiertas o cerradas. Generalmente al inicio de la entrevista se utiliza un cuestionario introductorio. Las entrevistas no están consideradas una técnica muy eficaz, pero son útiles para complementar otras técnicas de obtención de requisitos. Por lo común, los encuestados son usuarios de los sistemas existentes o usuarios en potencia del sistema propuesto. En algunos casos, son gerentes o empleados que proporcionan datos para el sistema propuesto o que serán afectados por él.

Las preguntas que deben realizarse en esta técnica, deben ser preguntas de alto nivel y abstractas que pueden realizarse al inicio del proyecto para obtener información sobre aspectos globales del problema del usuario y soluciones potenciales. Con frecuencia, se utilizan preguntas abiertas para descubrir sentimientos, opiniones y experiencias generales, o para explorar un proceso o problema. Este tipo de preguntas son siempre apropiadas, además que ayudan a entender la perspectiva del afectado y no están influenciadas por el conocimiento de la solución. El éxito de esta técnica combinada, depende de la habilidad del entrevistador y de su preparación para la misma. Los analistas necesitan ser sensibles a las dificultades que algunos entrevistados crean durante la entrevista y saber cómo tratar con problemas potenciales. Asimismo, necesitan considerar no sólo la información que adquieren a través del cuestionario y la entrevista, sino también, su significación. Por otro lado, Manies & Nikual<sup>(21)</sup> recomiendan que durante la preparación de los cuestionarios se tener en cuenta:

---

<http://scielo.sld.cu>



1. Mantener el cuestionario lo más pequeño posible. En lugar de utilizar un cuestionario grande, es mejor aplicar varios pequeños. En el caso de los cuestionarios grandes, por lo general, el usuario tiende a cansarse después de contestar las primeras 15-20 preguntas y no será muy objetivo al responder el resto. Como regla común un cuestionario no deberá contener más de 10-15 preguntas.
2. Estimar el tiempo necesario para responder las preguntas y un ambiente propicio que facilite la objetividad del cuestionario.
3. Estar seguro de que las preguntas están en un contexto libre de ambigüedades. Para asegurarlo, aplicar el prototipo a alguien cercano a la comunidad de los entrevistados y confirmar su comprensión de las preguntas estructuradas.
4. Antes de formular preguntas, estar seguro de que se necesitan las respuestas. En un contexto específico, las respuestas a muchas preguntas se convierten en información sin sentido.
5. Listar todas las posibles preguntas separadamente. Una vez que todos los requisitos y preguntas están listos, crear un plano cartesiano, con los requisitos en el eje X y las preguntas sobre el eje Y; para cada pregunta, identificar cuáles de los requisitos se están cumpliendo. Al final de este ejercicio se descartan las preguntas que no estén asociadas a uno de los requisitos.

Sistemas existentes: Esta técnica consiste en analizar distintos sistemas ya desarrollados que estén relacionados con el sistema a ser construido. Por un lado, podemos analizar las interfaces de usuario, observando el tipo de información que se maneja y cómo es manejada, por otro lado, también es útil analizar las distintas salidas que los sistemas producen (listados, consultas, etc.), porque siempre pueden surgir nuevas ideas sobre la base de estas.

Casos de uso basados en escenarios: Se han convertido en una característica fundamental de la notación de Lenguaje de Modelado Unificado (UML), que se utiliza para describir modelos de sistemas orientados a objetos. Los casos de uso permiten describir la posible secuencia de interacciones entre el sistema y uno o más actores, en respuesta a un estímulo inicial proveniente de un actor, es una descripción de un conjunto de escenarios, cada uno de ellos comenzado con un evento inicial desde un actor hacia el sistema. La mayoría de los requerimientos funcionales, sino todos, se pueden expresar con casos de uso.

Prototipos: Durante la actividad de extracción de requerimientos, puede ocurrir que algunos requerimientos no estén demasiado claros o que no se esté muy seguro de haber entendido correctamente los requisitos obtenidos hasta el momento, todo lo cual puede llevar a un desarrollo no eficaz del sistema final. Entonces, para validar los requisitos hallados, se construyen prototipos. Los prototipos son simulaciones del posible producto, que luego son utilizados por el usuario final, permitiéndonos conseguir una importante retroalimentación en cuanto a si el sistema diseñado en base a los requerimientos recolectados le permite al usuario realizar su trabajo de manera eficiente y efectiva. El desarrollo del prototipo comienza con la captura de requisitos. Desarrolladores y clientes se reúnen y definen los objetivos globales del software, identifican todos los requerimientos que son conocidos, y señalan áreas en

---

<http://scielo.sld.cu>



las que será necesaria la profundización en las definiciones. Luego de esto, tiene lugar un “diseño rápido”. El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles al usuario (por ejemplo, entradas y formatos de las salidas). El diseño rápido lleva a la construcción de un prototipo. El prototipo es evaluado por el cliente y el usuario y utilizado para refinar los requerimientos del software a ser desarrollado. Un proceso de iteración tiene lugar a medida que el prototipo es “puesto a punto” para satisfacer las necesidades del cliente lo que permite al mismo tiempo una mejor comprensión del problema por parte del desarrollador.

**Etnografía:** Dado que los sistemas no existen aisladamente sino en un contexto organizacional y social, los requisitos tienden a restringirse a ese contexto. La etnografía es una técnica observacional que permite encontrar y entender los requisitos implícitos en los procesos reales, que quedan ocultos en los formales o supuestos.

**Talleres de lluvia de ideas y de diseño de aplicaciones conjuntas:** La lluvia de ideas comenzó en el ámbito de las empresas, aplicándose a temas tan variados como la productividad, la necesidad de encontrar nuevas ideas y soluciones para los productos del mercado, encontrar nuevos métodos que desarrollen el pensamiento creativo a todos los niveles, etc. Pero pronto se extendió a otros ámbitos, incluyendo el mundo de desarrollo de sistemas; básicamente se busca que los involucrados en un proyecto desarrollen su creatividad. Esta técnica facilita descubrir requisitos cruzados que no salen a luz en las entrevistas u otras técnicas individuales. La lluvia de ideas permite generar gran cantidad de ideas de todo tipo, mientras que en el diseño de aplicaciones conjuntas la información es más visual y para generar básicamente requisitos e interfaces de sistemas.

**Herramientas CASE:** Existen diversas aplicaciones informáticas utilizadas para aumentar la productividad en la actividad de desarrollo de software reduciendo los costos en tiempo y dinero. Son las denominadas herramientas CASE (Ingeniería de Software Asistida por Computadora). Se pueden utilizar en las actividades de captura de requisitos, administrarlos y generar las especificaciones de requisitos. Estas herramientas permiten entre otras cosas tener un mayor control en proyectos complejos, reducir costos y retrasos en los proyectos, además ayudan a determinar la complejidad y los esfuerzos necesarios.

## Consideraciones finales

La ingeniería de requisitos ha evolucionado hacia la búsqueda de métodos, técnicas y herramientas que puedan ser aplicados en el proceso de definición de los requisitos para encarar una etapa de diseño del sistema de manera eficaz, dejando a un lado la obtención de una metodología capaz de adaptarse a cualquier tipo de sistema y paradigma, brindando un marco de trabajo referencial, independientemente del método a aplicar.

---

<http://scielo.sld.cu>



Este documento está bajo [Licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/).

El proceso de ingeniería de requisitos ayuda a recolectar la información necesaria para establecer la funcionalidad que se quiere alcanzar con el sistema. Para ello, es preciso contar con buenos métodos y técnicas para hacerlo, además de una comunicación fluida y constante con el cliente, ya que los requisitos deben reflejar las necesidades reales que el cliente quiere satisfacer.

El proceso de ingeniería de requisitos incluye un estudio de viabilidad, así como la obtención, análisis, especificación, validación y gestión de requisitos. La obtención y el análisis de requisitos debe ser un proceso iterativo de las actividades siguientes: descubrimiento de los requisitos, clasificación y organización de los requisitos, ordenación de requisitos por prioridades y negociación de posibles conflictos y documentar los requisitos ordenados y revisados.

La validación de los requisitos es fundamental para conocer la calidad, consistencia, completitud, realismo y verificabilidad.

Como proceso, la administración de requerimientos es fundamental en todo proyecto de desarrollo de software, de modo que se disponga de una especificación de requisitos clara y completa desde las fases iniciales para que se eviten problemas que generen retrasos y costos adicionales. Es importante que el documento que se obtenga de esta etapa sea un reflejo real del acuerdo de las partes involucradas.

## Referencias

1. Escalona MJ, Koch N. Ingeniería de Requisitos en Aplicaciones para la Web – Un estudio comparativo [Internet]. 2002 [citado 29 de junio de 2020]. Disponible en: <http://www.lsi.us.es/docs/informes/LSI-2002-4.pdf>
2. Sommerville I. Ingeniería de software [Internet]. Séptima Ed.. Madrid: Pearson Educación S.A.; 2005. 712 p. [citado 15 de noviembre de 2019]. Disponible en: [https://archive.org/stream/pdfy-g7E-3nrc3Q4yRJ4y/Ingenieria+de+Software+-+Ian+Sommerville+7ma+Edicion\\_djvu.txt](https://archive.org/stream/pdfy-g7E-3nrc3Q4yRJ4y/Ingenieria+de+Software+-+Ian+Sommerville+7ma+Edicion_djvu.txt)
3. Ayala B, Ramírez CM, Ocampo LM. La Ingeniería de Requerimientos aplicada al desarrollo de sistema de información. [Internet]. 2002 [citado 5 de abril de 2020]. Disponible en: [http://sedici.unlp.edu.ar/bitstream/handle/10915/4057/2\\_-\\_Ingenier%C3%ADa\\_de\\_requerimientos.pdf](http://sedici.unlp.edu.ar/bitstream/handle/10915/4057/2/_Ingenier%C3%ADa_de_requerimientos.pdf)
4. Fuggetta A, Di Nitto E. Software process [Internet]. 2014 [citado 30 de junio de 2020]. Disponible en: <http://www.alfonsofuggetta.org/wordpress/wp-content/uploads/2014/06/ICSE-2014-2014-Fuggetta.pdf>
5. May LJ. Major Causes of Software Projects Failures [Internet]. CROSSTALK The Journal of Defense Software Engineering. 1998 [citado 29 de junio de 2020]. Disponible en: <https://pdfs.semanticscholar.org/ac16/cd41d74f3907c6193b6a7e8ee0e0930eb6fe.pdf>
6. The Standish Group Report. Chaos. [Internet]. 2000 [citado 28 de junio de 2020]. Disponible en: <https://www.cs.nmt.edu/~cs328/reading/Standish.pdf>
7. Standish Group International, Inc. Chaos report 2015 [Internet]. 2015 [citado 30 de

---

<http://scielo.sld.cu>



- junio de 2020]. Disponible en: [https://www.standishgroup.com/sample\\_research\\_files/CHAOSReport2015-Final.pdf](https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf)
8. Leffingwell D. Calculating the Return on Investment from More Effective Requirements Management [Internet]. American Programmer. 1997 [citado 25 de junio de 2020]. Disponible en: <http://www.ibm.com/developerworks/rational/library/347.html>.
9. Wiegers K, Beatty J. Software Requirements, Third Edition [Internet]. Redmond, Washington: Microsoft Press; 2013 [citado 20 de junio de 2020]. 84 p. Disponible en: <https://ptgmedia.pearsoncmg.com/images/9780735679665/samplepages/9780735679665.pdf>
10. IEEE Standard Glossary of Software Engineering Terminology. STD 610.12-1990 [Internet]. 1990 [citado 06 de abril de 2020]. Disponible en: [http://www.mit.jyu.fi/ope/kurssit/TIES462/Materiaalit/IEEE\\_SoftwareEngGlossary.pdf](http://www.mit.jyu.fi/ope/kurssit/TIES462/Materiaalit/IEEE_SoftwareEngGlossary.pdf)
11. Moya S, Hinojosa C, Reyes R. Formulación de un modelo para evaluar herramientas de análisis de requerimientos, basado en la norma ISO 25000. Revista DECC Report, Tendencias en computación [Internet]. 2010 [citado 25 de junio de 2020];1(2). Disponible en: <https://journal.espe.edu.ec/ojs/index.php/geeks/article/view/241>
12. Fernández Alarcón V. Desarrollo de sistemas de información. Una metodología basada en el modelado. [Internet]. Edicions UPC. E-aula politécnica; [citado 25 de junio de 2020]. 120 p. Disponible en: <https://upcommons.upc.edu/handle/2099.3/36751>
13. Mariño SI, Godoy MV, Alfonzo P, Matozo A, Fernández M. Técnicas de Ingeniería de Requerimientos aplicadas en proyectos de desarrollo de software y su relación con la investigación y la docencia universitaria. Un caso de estudio [Internet]. RedUNCI. 2012. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/23697>
14. Ingeniería de requerimientos. Características de los requerimientos [Internet]. [citado 3 de abril de 2020]. Disponible en: <https://prezi.com/tlzhvytxsbnc/caracteristicas-basicas-de-los-requerimientos-de-software/>
15. Boehm B. Software Engineering Economics [Internet]. 1984 [citado 30 de junio de 2020]. Disponible en: [http://extras.springer.com/2002/978-3-642-59413-7/4/rom/pdf/Boehm\\_hist.pdf](http://extras.springer.com/2002/978-3-642-59413-7/4/rom/pdf/Boehm_hist.pdf).
16. Pressman RS. Ingeniería de software. Un enfoque práctico [Internet]. Séptima. México: MCGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V.; 778 p. Disponible en: <http://cotana.informatica.edu.bo/downloads/Id>  
Ingenieria.de.software.enfoque.practico.7ed.Pressman.pdf
17. Arias Chaves M. La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software [Internet]. Sistema de Información Científica Redalyc. Revista de las Sedes Regionales. 2005 [citado 4 de diciembre de 2019]. Disponible en: <http://www.redalyc.org/articulo.oa?id=66612870011>
18. Herrera LJ. Ingeniería De Requerimientos Ingeniería De Software [Internet]. [citado 3 de abril de 2020]. Disponible en: <http://www.monografias.com/trabajos6/resof/resof.html>
19. Guerra CA. Obtención de Requerimientos. Técnicas y Estrategia [Internet]. [citado 1 de abril de 2020]. Disponible en: <https://sg.com.mx/revista/17/obtencion-requerimientos-tecnicas-y-estrategia>

---

<http://scielo.sld.cu>



20. 7 Técnicas de levantamiento de requerimientos software [Internet]. PMOinformatica.com. 2016 [citado 1 de abril de 2020]. Disponible en: <http://www.pmoinformatica.com/2016/08/tecnicas-levantamiento-requerimientos.html>
21. Markus Manies, Uolevis Nikual. La elicitación de requisitos en el contexto de un proyecto de software. Ing USBMed. 2011;2(2):25-9.

#### Conflicto de interés

El autor declara que no existe conflicto de intereses.

---

<http://scielo.sld.cu>



Este documento está bajo [Licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/).