



APUNTE ELECTRÓNICO

Informática II (Administración de Requerimientos)

Licenciatura en Informática



COLABORADORES

DIRECTOR DE LA FCA

Dr. Juan Alberto Adam Siade

SECRETARIO GENERAL

Mtro. Tomás Humberto Rubio Pérez

COORDINACIÓN GENERAL

Mtra. Gabriela Montero Montiel
Jefe de la División SUAyED-FCA-UNAM

COORDINACIÓN ACADÉMICA

Mtro. Francisco Hernández Mendoza
FCA-UNAM

AUTOR

Mtro. Rene Montesano Brand

REVISIÓN PEDAGÓGICA

CORRECCIÓN DE ESTILO

DISEÑO DE PORTADAS

L.CG. Ricardo Alberto Báez Caballero
Mtra. Marlene Olga Ramírez Chavero

DISEÑO EDITORIAL

Mtra. Marlene Olga Ramírez Chavero



Dr. Enrique Luis Graue Wiechers
Rector

Dr. Leonardo Lomelí Vanegas
Secretario General



Dr. Juan Alberto Adam Siade
Director

Mtro. Tomás Humberto Rubio Pérez
Secretario General



Mtra. Gabriela Montero Montiel
Jefa del Sistema Universidad Abierta
y Educación a Distancia

Informática II (Administración de requerimientos) **Apunte electrónico**

Edición: Agosto 2017

D.R. © 2017 UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Ciudad Universitaria, Delegación Coyoacán, C.P. 04510, México, Ciudad de México.

Facultad de Contaduría y Administración
Circuito Exterior s/n, Ciudad Universitaria
Delegación Coyoacán, C.P. 04510, México, Ciudad de México.

ISBN: En trámite
Plan de estudios 2012, actualizado 2016.

"Prohibida la reproducción total o parcial por cualquier medio sin la autorización escrita del titular de los derechos patrimoniales"

"Reservados todos los derechos bajo las normas internacionales. Se le otorga el acceso no exclusivo y no transferible para leer el texto de esta edición electrónica en la pantalla. Puede ser reproducido con fines no lucrativos, siempre y cuando no se mutile, se cite la fuente completa y su dirección electrónica; de otra forma, se requiere la autorización escrita del titular de los derechos patrimoniales."

Hecho en México

OBJETIVO GENERAL

El alumno será capaz de identificar y especificar los requerimientos de los involucrados en el desarrollo de un sistema de información a fin de orientar las actividades de análisis y diseño de sistemas.

TEMARIO DETALLADO

(Horas 96)

| | |
|--|-----------|
| 1. Introducción | 16 |
| 2. Identificación de requerimientos | 24 |
| 3. Especificación de requerimientos | 28 |
| 4. Validación de requerimientos | 28 |

INTRODUCCIÓN

A lo largo de esta asignatura se abordará uno de los aspectos más significativos del ciclo de vida de los sistemas de información y que tiene lugar en la fase del análisis del sistema, se trata de la gestión de requerimientos, que va desde la correcta especificación y captura de los mismos, hasta su revisión, validación y control de cambios, con lo cual, los desarrolladores y clientes, podrán establecer las características generales que tendrá el sistema que se va a desarrollar y sus funciones principales.

Una clara y ordenada planificación del sistema permitirá que éste cumpla satisfactoriamente con el objetivo para el que sea creado, es decir, que las acciones se enfoquen en que se produzca lo que realmente se desea y sea eficaz para la organización. Esto podrá realizarse a través de la especificación y gestión de los requerimientos del sistema; la falta de ello podría traer como consecuencia problemas de funcionalidad en el sistema y de producción para la organización.

Con la información obtenida de la fase de recopilación, el desarrollador podrá tener una visión general del sistema que se pretende desarrollar de acuerdo con las necesidades y características del negocio que se trate. En esta etapa es importante recalcar el contacto con los usuarios finales del sistema, ya que a partir de sus necesidades será posible diseñar y conformar un sistema robusto y amigable que permita el cumplimiento de sus actividades, e incluso que a partir de la información recabada se pueden mejorar los procedimientos y métodos acostumbrados.

La fase de análisis de los requerimientos del sistema tiene el propósito de proporcionar a los desarrolladores la información necesaria para determinar el comportamiento general del sistema y sus funciones principales. De ésta fase se obtiene información valiosa como las características de los equipos que operarán al

sistema, el tipo de estrategia de desarrollo (selección del modelo de desarrollo), el tipo de información que se va a manejar, la estructura general, etcétera.

La asignatura se distribuye en cuatro unidades, en las cuales se ahondará de forma específica el estudio de estas fases.

De este modo, la **unidad 1** (Introducción), nos proporcionará las bases necesarias para el desarrollo de un plan para la administración de los requerimientos con base en su clasificación y al perfil de la organización.

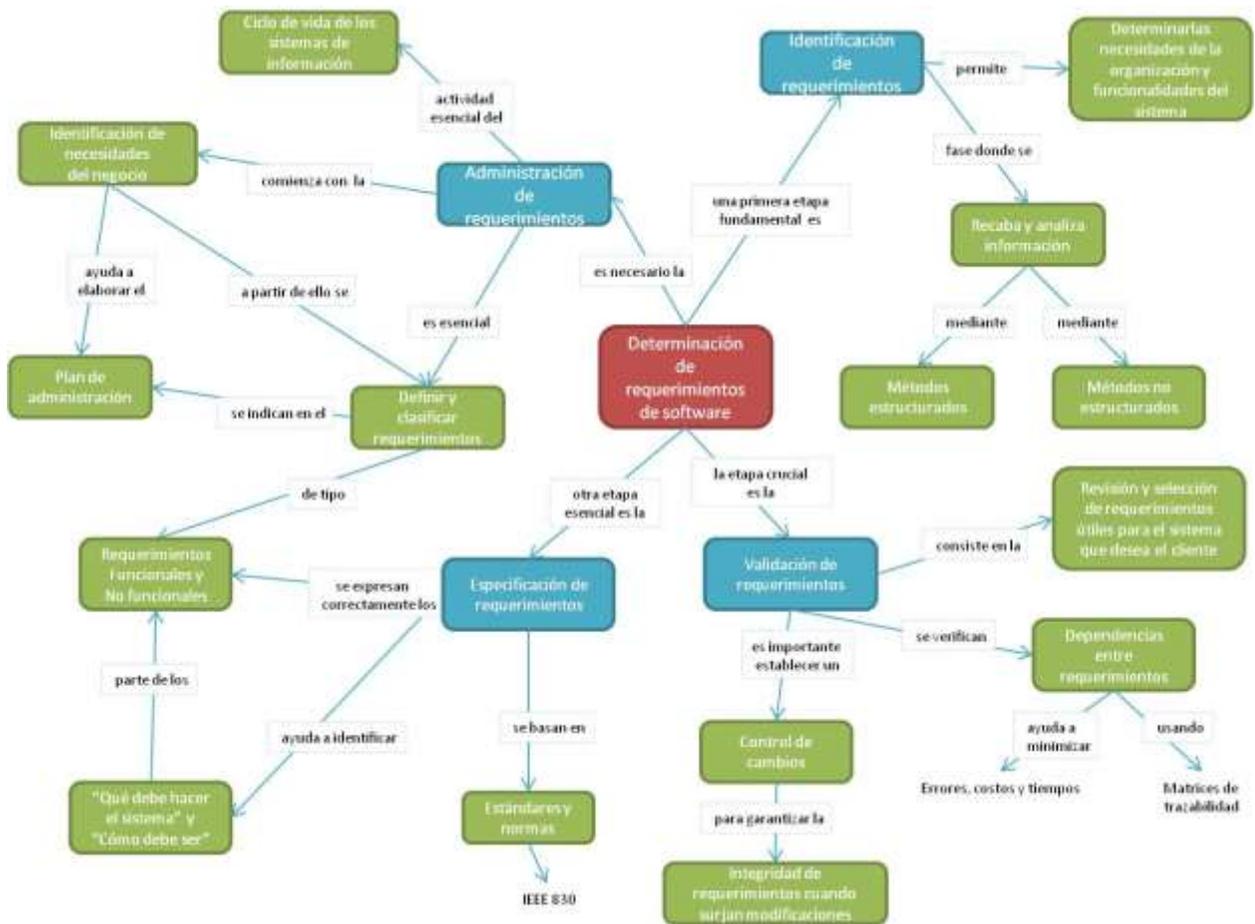
En la **unidad 2** (Identificación de requerimientos) se definirán los métodos y técnicas necesarias para la identificación de los requerimientos de un sistema de información.

La **unidad 3** (Especificación de requerimientos) nos ayudará a identificar los requerimientos funcionales y los no funcionales que se obtuvieron de los métodos y técnicas de identificación de requerimientos.

Y finalmente, en la **unidad 4** (Validación de requerimientos) veremos que, una vez recabada y analizada la información, se procederá a verificar si los requerimientos establecidos son los adecuados acordes con el perfil de la organización y sus necesidades.



ESTRUCTURA CONCEPTUAL





UNIDAD 1

Introducción



OBJETIVO PARTICULAR

Desarrollará un plan para la administración de requerimientos tomando como base los conceptos y clasificación de los requerimientos.

TEMARIO DETALLADO

(16 horas)

1. Introducción

1.1. Expectativas, necesidades, problemas y requerimientos

1.2. Definición de necesidades de negocio

1.3. Definición de requerimiento

1.4. Clasificaciones de requerimientos

1.5. Procesos para la administración de requerimientos

1.6. Planeación para administrar requerimientos

1.6.1. Plan de administración de requerimientos

1.6.2. Definición de atributos de los requerimientos

INTRODUCCIÓN

En el ámbito de desarrollo de sistemas de información, una de las fases más importantes y de mayor impacto a lo largo de todo el ciclo de vida de los sistemas, es la correcta especificación, redacción y administración de los requerimientos, los cuales definen la forma como el sistema deberá comportarse y sus características generales.

La administración de requerimientos es un proceso que debe comenzar con la identificación de las necesidades de una organización y problemas asociadas a ellas, que en un futuro próximo, la implementación de un sistema de información deberá de cubrir y resolver.

Los requerimientos identificados deberán ser especificados y redactados en un documento que represente un acuerdo entre desarrolladores y clientes, mismo que servirá para su clasificación según su tipo y que permitirá la implementación de un plan de administración efectivo, que reduzca al mínimo los errores de diseño y que facilite los cambios que se lleguen a efectuar en los requerimientos a lo largo del proceso de desarrollo del sistema.

1.1. Expectativas, necesidades, problemas y requerimientos

El ciclo de vida de desarrollo de sistemas informáticos está dividido en ciertas fases donde se van realizando actividades específicas y enlazadas todas ellas para que el producto final sea el esperado. Para que se pueda iniciar con este proceso, es importante que en primera instancia conozcas y diferencies ciertos conceptos básicos.



De manera general, un problema es un conjunto de hechos o circunstancias que dificultan la consecución de algún fin.

En lo que toca a los sistemas de información, un usuario puede identificar alguna situación que esté generando algún inconveniente en la organización (un problema) y hacerlo del conocimiento de los demás desde su punto de vista, no necesita proporcionar detalles al respecto ni mucho menos dar soluciones.

Precisamente los sistemas son desarrollados para dar solución a una cierta problemática presentada dentro de la organización, que ayude a la agilización de los procesos internos de la organización, por ejemplo, el manejo efectivo del inventario en un almacén, el proceso de la información financiera de la organización, entre otros.



Expectativas

Una expectativa, de forma simple, se puede definir como la esperanza de realizar o conseguir algo.

En el caso de los sistemas de información, podemos decir que es todo aquello que esperamos que el sistema realice para facilitar las actividades de la organización y la toma de decisiones.

Si las expectativas no se definen claramente, se puede correr el riesgo de que no se seleccionen las tecnologías apropiadas o no funcionen como se desea, o si son poco realistas, no se podrán identificar ciertos riesgos o alcanzar a ver algunas oportunidades.

Necesidades

Una necesidad se define como la carencia o falta de algo o alguien.

Desde el punto de vista de los sistemas de información, son todas aquellas cosas y personas que hacen falta para que el sistema sea construido y puesto en marcha.

La identificación de las necesidades en una organización es el primer paso del análisis de un sistema, que permite definir los faltantes y diseñar un plan para el proyecto a desarrollar.

Normalmente se realiza de forma grupal, entre los desarrolladores y los responsables del proyecto o de la institución.



Requerimiento

Un requerimiento, de forma general, se puede definir como la petición de una cosa que se considera necesaria.

Podemos decir que todo aquello que es necesario para poder realizar una tarea es un requerimiento; por ejemplo, para poder realizar una receta de cocina, uno de sus requerimientos son los ingredientes; para poder escribir una carta, uno de sus requerimientos es una hoja de papel con ciertas características, etc.

La definición de los requerimientos de un sistema especifica qué es lo que debe hacer éste y sus cualidades principales. Reflejan las necesidades de los clientes y ayudan a resolver algún problema.

Los requerimientos pueden ser de diversa naturaleza, pero de manera general cuentan con ciertas características únicas que ayudan a solventar una necesidad específica (en el tema 1.3 veremos a detalle los requerimientos desde el punto de vista de los sistemas de información).

Los cuatro conceptos definidos anteriormente se relacionan de tal forma, que cada uno de ellos ayuda a definir al otro, es decir, a partir de la identificación de un problema se identifican carencias, éstas generan expectativas y para cuando sean solventadas, esas carencias se convierten en necesidades y finalmente, esas necesidades se vuelven requerimientos de aquellas herramientas o medios que ayudarán a su solución.

1.2. Definición de necesidades de negocio

Cuando una empresa u organización decide poner a disposición de sus clientes un producto o servicio, se da pie para comenzar un proceso conocido como negocio. El fin último de cualquier empresa es la generación de utilidades; para poder hacerlo es necesario que ofrezca productos y/o servicios en un mercado interesado en ellos y por ende, que se genere el proceso de compra-venta de los productos y/o servicios ofertados para generar las utilidades.

Para que un producto o servicio nuevo tenga oportunidades de aceptación en un mercado, lo más recomendable es que se desarrolle previamente un documento formal denominado plan de negocios.

El plan de negocios ayuda a la empresa a establecer el *tipo de negocio* que se desee realizar, ya sea venta de productos como lácteos, servicios de banca comercial, venta de paquetes vacacionales, etc. A partir de la definición del tipo de negocio, se definen los objetivos que la empresa persigue con ese negocio y las acciones que deben realizarse para alcanzar dichos objetivos. Además, es una guía que permite a la empresa identificar aquellos aspectos que obstaculizan la realización del negocio, para implementar las acciones necesarias para solventarlos y tomar decisiones correctas.

El plan de negocios, adicionalmente, permite a las empresas determinar si el negocio en cuestión es viable tanto económica como financieramente. Adicionalmente, aporta información importante relacionada con los recursos humanos necesarios, las estrategias para conseguir los objetivos planteados, el mercado meta del negocio y todo el conjunto de actividades para poder realizarlo (parte operativa).

Cuando se desarrolla un plan de negocios, éste debe de partir de una idea o iniciativa que nace como respuesta a una *necesidad de los clientes de la organización*, la cual se convierte en el *objetivo fundamental del plan de negocios*.

De forma general, en un mercado encontramos personas que compran bienes o servicios para satisfacer sus necesidades básicas o específicas; dichas necesidades, desde el punto de vista empresarial, se convierten en oportunidades de negocio que las empresas buscan explotar para satisfacerlas a cambio de una utilidad.

Existen diversos métodos que permiten satisfacer las necesidades de los clientes, tan diferentes como los clientes mismos, pero de manera general estos métodos deben permitir a las empresas responder a las siguientes preguntas:

- ♦ ¿En cuál segmento de mercado estoy?
- ♦ ¿En cuál quiero estar?
- ♦ ¿A qué clientes quiero atender?
- ♦ ¿Con cuáles bienes o servicios?
- ♦ Mi vocación y mis aptitudes, ¿hacia cuál mercado me impulsan?
- ♦ ¿Cómo va a crecer ese sector en los próximos años?
- ♦ ¿Qué estoy haciendo para ingresar en él?

Las preguntas anteriores permiten a la empresa definir el curso de acción, y es precisamente en el plan de negocios donde se plantea de forma estratégica una forma de trabajar razonada y bien fundamentada en la investigación del entorno de la empresa.

Todo plan de negocios debe desarrollarse a partir de dos aspectos fundamentales: la misión y la visión de la empresa. El primero establece lo que hace la organización y para qué tipo de cliente lo hace; y el segundo muestra a dónde quiere llegar la organización a largo plazo.

La misión es un enunciado que permite ver de forma general qué hace la empresa y para quién lo hace, de tal forma que responde a las siguientes preguntas:

- a. ¿Qué vendemos? (oferta)
- b. ¿A quién se lo vendemos? (demanda)
- c. ¿Por qué nos eligen? (ventaja competitiva)

Al establecer una misión, la empresa establece una razón de ser y, por ende, puede destinar sus recursos a sus actividades de una forma más eficiente con el objetivo de cumplir con su misión.

Algunos ejemplos de la misión de diversas empresas:

“Refreshar al mundo en cuerpo, mente y alma. Inspirar momentos de optimismo a través de nuestras marcas y acciones, para crear valor y dejar nuestra huella en cada uno de los lugares en los que operamos.” Coca Cola Company, México.

“Proveer soluciones de calidad, a través de la iniciativa y respuesta de sus integrantes, ofreciendo tecnologías de vanguardia y servicios de valor agregado para asegurar la satisfacción de nuestros clientes.” Hewlett-Packard.



“Organizar la información del mundo y hacerla universalmente accesible y útil”.
Google.

Por otro lado, la visión de una empresa permite establecer un objetivo a largo plazo, ya que contesta a la pregunta ¿Dónde deseo estar?

Ejemplos de visión de algunas empresas:

- Utilidades:

Maximizar el retorno a los accionistas, sin perder de vista la totalidad de nuestras responsabilidades.

- Gente:

Ser un excelente lugar para trabajar, en donde nuestro personal se inspire para dar lo mejor de sí.

- Cartera de Productos:

Ofrecer al mundo una cartera de marcas de bebidas que se anticipan y satisfacen los deseos y las necesidades de las personas.

- Socios:

Formar una red de socios exitosa y crear lealtad mutua.

- Planeta:

Ser un ciudadano global, responsable, que hace su aporte para un mundo mejor. (Coca Cola Company, México)

Otro ejemplo:

Intel está superando los límites de la innovación para hacer que las vidas de las personas sean más excitantes, más satisfactorias y más fáciles de gestionar. Nuestra dedicación constante al avance de la tecnología ha transformado el mundo a pasos agigantados.

Somos una empresa que siempre está en movimiento, alimentando a un mercado que nunca descansa. Inspiramos a nuestros socios a que desarrollen productos y servicios innovadores, agrupamos al mercado para prestar asistencia a nuevos productos y promovemos estándares. Hacemos todo esto para poder ofrecer colectivamente mejores soluciones con mayores ventajas de forma más rápida. (Intel)

El correcto establecimiento de la misión y visión en las empresas genera una guía para sus acciones, ya que el objetivo principal es cumplir con su misión para alcanzar su visión.

El plan de negocios permite establecer los caminos para alcanzar los objetivos de la empresa, estos caminos a su vez ayudan a identificar los recursos con que se cuenta y las carencias de la empresa, esas carencias se convertirán en necesidades al ser incorporadas a las estrategias que a la postre, ayudarán al éxito del negocio.

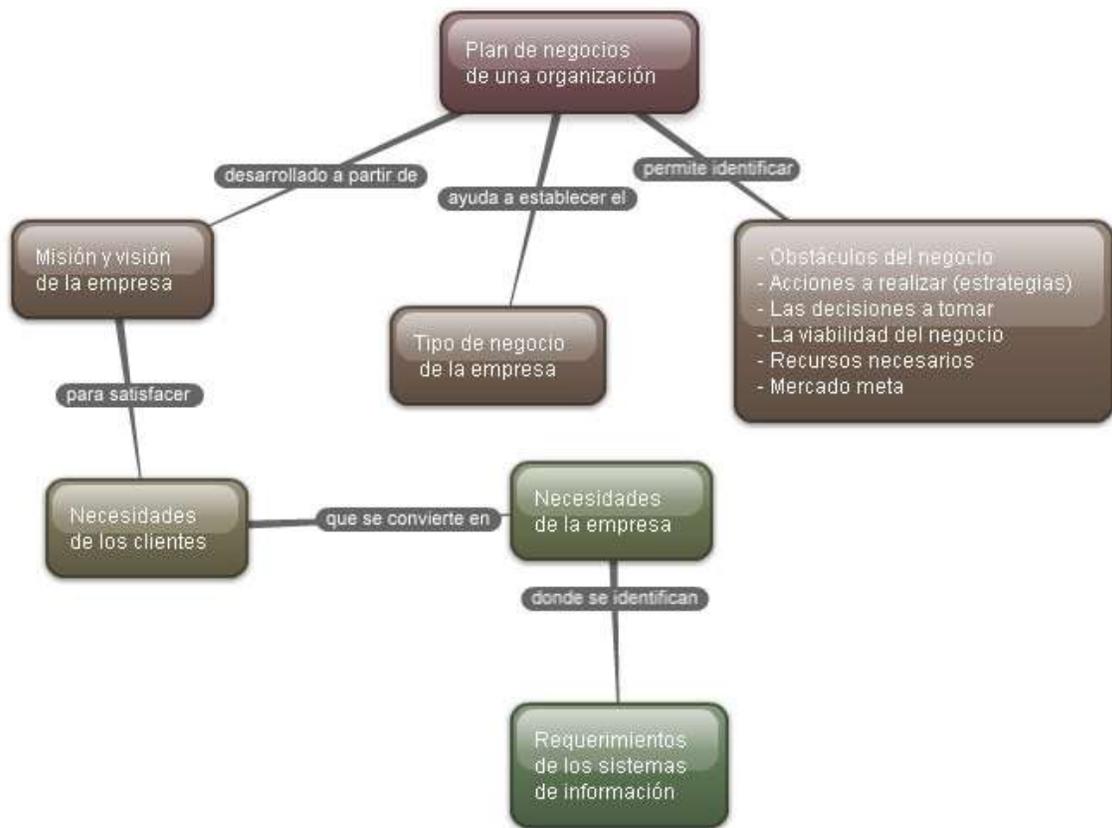


Figura 1.1. Misión y visión de una empresa

1.3. Definición de requerimiento

Dentro del ámbito de los requerimientos se manejan dos enfoques:

Dentro de la ingeniería de software, entendemos como “requerimientos” a “las declaraciones que identifican atributos, capacidades, características y/o cualidades que necesita cumplir un sistema para que tenga valor y utilidad para el usuario” (alegsa, Diccionario de informática, 2011). Es decir, un requerimiento muestra todos los elementos que son necesarios para la construcción del sistema.

Cuando hablamos de software en general, un requerimiento es una serie de elementos básicos necesarios para que las aplicaciones funcionen de manera correcta, estos pueden ser cantidad de memoria, sistema operativo, tipo de procesador, entre otros.

Pero a fin de cuentas, y de forma general, los requerimientos indican lo que el sistema debe de hacer, es decir, lo que el cliente o usuario desea y espera que haga ese sistema, aunque en ese momento no se indique el cómo.

La importancia de la definición de requerimientos se ve reflejada en el costo del proyecto, sobre todo cuando toca reparar algún error u omisión en el sistema que se pudo haber previsto.

1.4. Clasificaciones de requerimientos

Los requerimientos de los sistemas de información pueden ser divididos en dos clasificaciones básicas:

Requerimientos funcionales

Definen las capacidades que deberá tener el sistema que se va a desarrollar, describiendo los procesos que llevan a la transformación de las entradas del sistema para obtener las salidas deseadas (lo que el software debe o no hacer).

A su vez, los requerimientos funcionales se clasifican en:

Requerimientos de datos o información. Estos requerimientos hacen referencia al contenido o tipo de información que manejará el sistema. El objetivo de los requerimientos de datos es responder a la pregunta ¿qué información debe almacenar y administrar el sistema? Por ejemplo, el sistema debe almacenar la dirección completa de los trabajadores de la empresa.

Requerimientos de interfaz (con el usuario). Son los requerimientos que buscan determinar la forma en cómo los usuarios van a interactuar con el sistema. El objetivo de definir los requerimientos de interfaz es responder a la pregunta ¿cómo va a interactuar el usuario con el sistema? Por ejemplo, el sistema debe contener una pantalla donde los usuarios escriban su nombre y contraseña para poder ingresar a su contenido.

Requerimientos de navegación. Estos requerimientos están para a determinar la forma en que los usuarios van a navegar o a desplazarse dentro del sistema entre las diferentes secciones que lo componen. Por ejemplo, el sistema debe contener una pantalla de menú principal que permita el acceso a los diversos módulos que conforman al sistema.

Requerimientos de personalización. A través de los requerimientos de personalización se busca describir la forma en que el sistema deberá adaptarse a los diferentes tipos de usuario que interactuarán con él. Por ejemplo, los usuarios serán clasificados de acuerdo con su categoría dentro de la empresa y acorde con ella podrán tener acceso a todos los módulos del sistema o solo a algunos de ellos.

Requerimientos transaccionales o funcionales internos. Son los requerimientos que buscan definir la funcionalidad interna del sistema, excluyendo los aspectos de interacción. Po ejemplo, el sistema deberá generar un reporte de actividades de los usuarios diariamente (se define el formato del reporte).

Requerimientos no funcionales

Definen las posibles causas o características que son limitantes del sistema; como por ejemplo el rendimiento del sistema, la disponibilidad de equipos, etc. Sommerville indica que los requerimientos no funcionales, como su nombre sugiere, son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino que están orientados más hacia las necesidades que surgen del usuario (2005, p. 125).

Para definir los requerimientos no funcionales los desarrolladores se basan en [el estándar ISO/IEC 9126](#), ahora englobado en el proyecto SQuaRE para el desarrollo de la norma ISO 25000, que define un modelo independiente de la tecnología para caracterizar la calidad de software y considera las siguientes características:

Funcionalidad. Define las diferentes características que son necesarias para alcanzar el funcionamiento deseado del sistema, entre estas características podemos encontrar la interoperabilidad con los usuarios y la seguridad del sistema.

Confiabilidad. Se refiere a la capacidad del sistema para mantener sus niveles de rendimiento bajo ciertas condiciones específicas y en un tiempo de respuesta dado. Entre los requerimientos de confiabilidad encontramos la tolerancia a las fallas y la capacidad de recuperación del sistema.

Amabilidad (o algunas veces llamada *usabilidad*). Describe el nivel de complejidad que puede encontrar el usuario final para poder utilizar el sistema, entre estos requerimientos encontramos la curva de aprendizaje, la eficacia y la eficiencia de los usuarios.

Eficiencia. Son las características que definen la diferencia entre el rendimiento del sistema y la cantidad de recursos que éste consume durante su funcionamiento, entre estos se encuentran la cantidad de memoria RAM empleada, el tiempo de respuesta, la cantidad de procesos en el sistema operativo y los recursos de red empelados.

Capacidad de mantenimiento. Se refiere a la capacidad del sistema para aceptar cambios dentro de su estructura sin perder funcionalidad, dentro de estos aspectos se encuentran la estabilidad y las validaciones de los diferentes módulos del sistema.

Compatibilidad. Es la capacidad del sistema de ser instalado en diferentes plataformas operativas sin presentar cambios en su funcionamiento como por ejemplo: adaptabilidad, capacidad de instalación.

Portabilidad, capacidad del software de ser transferido de un entorno a otro.

Productividad. Capacidad del software de permitir a los usuarios gastar la cantidad apropiada de recursos en relación con la efectividad obtenida.

Seguridad. Capacidad del software para cumplir con los niveles de riesgo permitidos tanto para posibles daños físicos como para posibles riesgos de datos.

Satisfacción. Capacidad del software de cumplir con las expectativas de los usuarios en un contexto determinado.

Los requerimientos no funcionales pueden catalogarse en tres categorías principales:

Requerimiento del producto. Enfocados a las características que define la forma de comportarse del producto, en este caso el sistema de información. Entre ellos podemos encontrar por ejemplo, la rapidez del sistema, la cantidad de memoria empleada, el espacio de almacenamiento en disco duro, etc.

Requerimientos organizacionales. Estos requerimientos se enfocan principalmente en la observancia de las políticas y procedimientos de la organización, así como de las de los desarrolladores.

Requerimientos externos. Incluye a todos los factores externos del sistema y del proceso de desarrollo, como por ejemplo la interoperabilidad, es decir, la forma como el sistema interactuaría con otros sistemas, los legislativos, que se enfocan en el acatamiento de las leyes vigentes como por ejemplo derechos de autor, propiedad intelectual, etc.

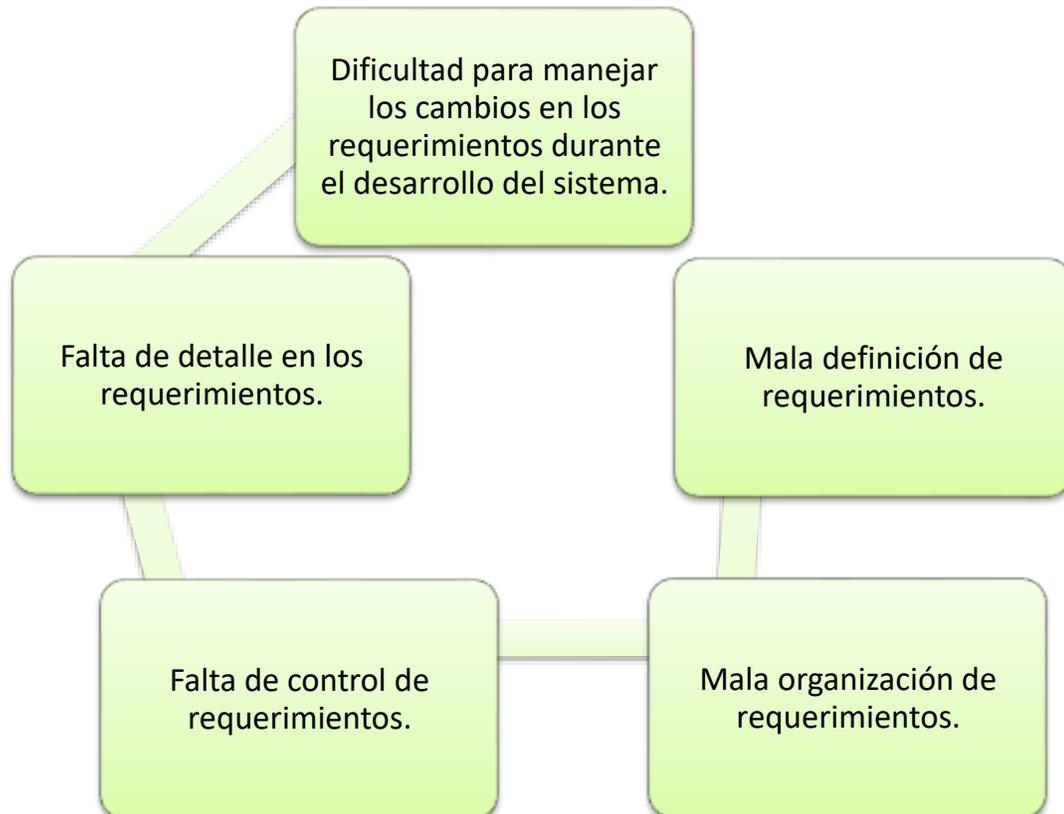
La siguiente figura 1.2. muestra de forma general los tipos de requerimientos no funcionales de acuerdo con Ian Sommerville:



Figura 1.2. Tipos de requerimientos no funcionales (Sommerville, 2005, p. 126)

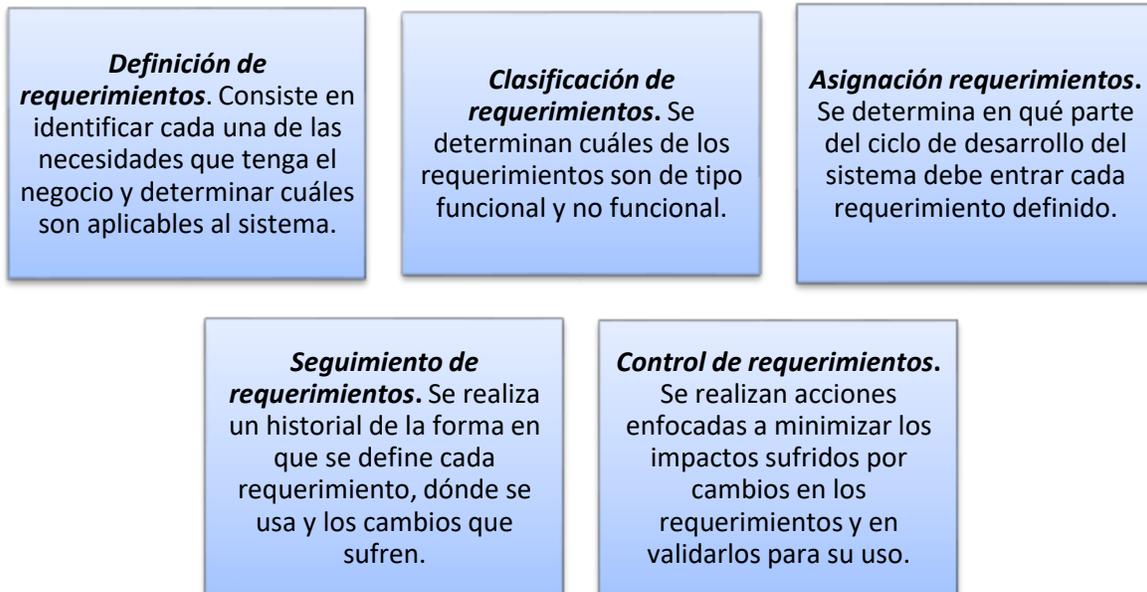
1.5. Procesos para la administración de requerimientos

Cuando se desarrolla un nuevo sistema de información, la mala administración de requerimientos es, en muchos de los casos, la causa principal de que los sistemas no funcionen de forma adecuada. Dentro de los principales problemas que encontramos debido a la mala administración de los requerimientos tenemos:





La administración de requerimientos en el desarrollo de sistemas de información está relacionada con diversas actividades que permiten realizar un levantamiento y control adecuado de los requerimientos, estas actividades son:



Estas actividades comienzan durante la fase de análisis del sistema y continúan durante todo su ciclo de vida; su correcto seguimiento permite realizar un control y monitoreo del proyecto adecuado y asegurar un producto de calidad.

El proceso de administración de requerimientos de RUP (*Rational Unified Process*) captura varias de las “mejores prácticas” en lo que a desarrollo de software se refiere, de tal forma que los procesos establecidos por RUP son aplicables a un rango amplio de proyectos de software.

RUP es un proceso de desarrollo de software, que junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas de información, empleado principalmente bajo el paradigma de programación orientada a objetos, ya que

ayuda a definir de una forma eficiente los atributos y características de cada elemento que conformarán a los objetos y clases que compondrán al sistema.

El proceso de administración de requerimientos consiste en definir, organizar y documentar las especificaciones funcionales del sistema, sus limitantes y restricciones; adicionalmente, se da seguimiento y documenta todas las decisiones y alternativas que son tomadas durante el desarrollo del sistema, así como también permite la captura y comunicación de los requerimientos del negocio. El empleo de RUP en la administración de requerimientos establece el empleo de “casos de uso” y “escenarios” para la captura de los requerimientos funcionales del sistema (temas que abordaremos con más detalle en la unidad 3).

Los requerimientos del sistema son las partes más importantes en el desarrollo del proyecto. Los requerimientos, podemos decir, definen a la perfección el modo en que debe comportarse el sistema, es por ello por lo que los usuarios finales tienen que aceptar y comprender todos los requerimientos que sean especificados.

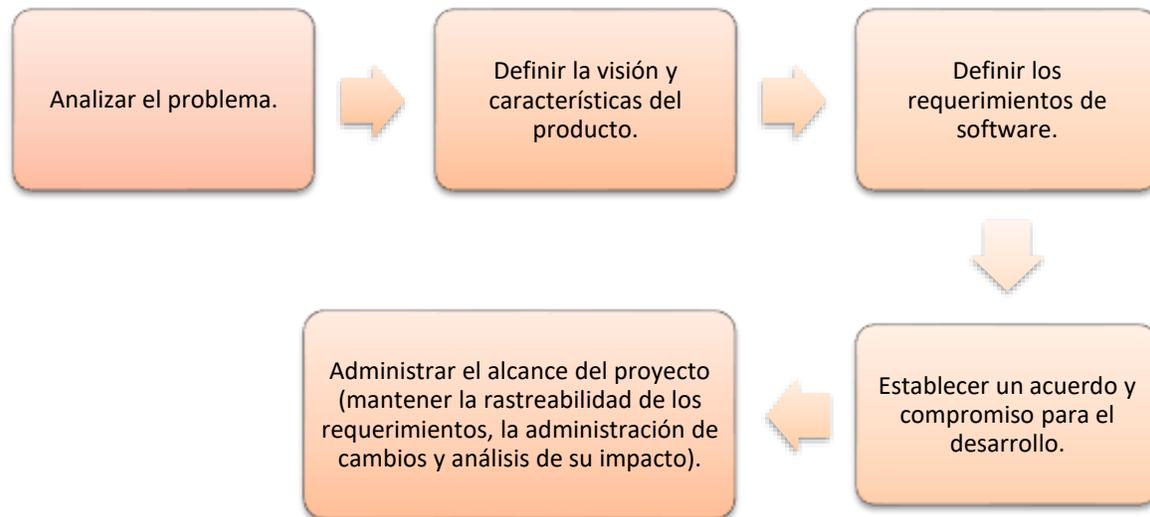
Los objetivos del proceso de administración de requerimientos son:



La identificación de los requerimientos es un proceso que requiere de entrevistar a todos los involucrados en el desarrollo del sistema, desde los clientes, desarrolladores, hasta los usuarios finales; consiste en anotar todas las peticiones que se realicen y a partir de ellas determinar las necesidades del proyecto y entonces expresarlas como un requerimiento.

Uno de los factores de riesgo presente a lo largo de todo el desarrollo del sistema está en los cambios; generalmente se pueden presentar desde el inicio del desarrollo hasta fase posteriores a la puesta en marcha, donde se puede solicitar un cambio durante la etapa de mantenimiento. La correcta administración de los cambios y la configuración en un proyecto de software es importante debido a que estos impactan directamente en los requerimientos, por ello, es necesario considerar su importancia, evaluar su costo y determinar si es posible o no implementarlos.

Las actividades que se realizan en el proceso de administración de requerimientos de RUP son las siguientes:



En el análisis del problema se busca comprender de mejor manera la problemática, las necesidades de los clientes y proponer una solución de alto nivel y establecer sus límites.

En lo que se refiere a entender las necesidades de los clientes, se deben determinar cuáles son las fuentes de información adecuadas que serán empleadas y la forma en que se va a obtener dicha información. Cuando se desarrollan sistemas de información, es deseable que dentro de las fuentes de información, se incluyan personas con experiencia dentro de la organización y que de una u otra forma, van a tener contacto con el sistema; estas personas pueden ser por ejemplo, el personal del área en donde se planea instalar el sistema, las personas que van a tener contacto directo con el sistema, los inversionistas y directivos de la empresa, etc.

Existen múltiples técnicas que nos pueden ayudar a obtener la información necesaria que nos permita identificar las necesidades de nuestros clientes, entre

ellas encontramos los cuestionarios, las entrevistas, prototipos conceptuales, etc. El resultado de aplicar estas técnicas, es la generación de una lista de peticiones o necesidades que pueden ser descritas de forma gráfica o textual, las cuales al ser priorizadas, se convertirán en requisitos del sistema.

La etapa de definición de las características del producto o del sistema se enfoca principalmente en definir a los actores que intervendrán en los casos de uso que ayudarán en la definición de los requerimientos no funcionales del sistema, en esta etapa se recomienda realizar algunos prototipos conceptuales y modelos según las peticiones de los clientes recabadas en la etapa de análisis del problema. El resultado generado por la etapa de definición del sistema es una descripción de su funcionamiento de forma gráfica y en un lenguaje natural.

Con la ayuda de lo recabado en las etapas anteriores, será posible generar una visión común de las características del sistema para los clientes y desarrolladores que permitirán establecer acuerdos en lo que se refiere a los recursos que serán asignados para su desarrollo y los tiempos del mismo.

El alcance del sistema se define a partir del conjunto de requerimientos obtenidos en las etapas anteriores, la clave de esta etapa se centra en el cumplimiento de los tiempos asignados para el mismo y la correcta administración de los recursos disponibles asignados para esos lapsos de tiempo. La definición de atributos de los requerimientos como prioridad, esfuerzo y riesgo son técnicas útiles que permiten manejar el alcance del proyecto de una manera más eficiente.

Pasadas las etapas anteriores, será posible realizar un perfeccionamiento de la definición de las características del sistema, esta definición deberá de estar detallada de tal forma, que los clientes puedan entenderla, lo que permitirá que se alcancen acuerdos más fácilmente. La definición perfeccionada contempla el cumplimiento de los requerimientos funcionales y de aquellos que sean de carácter

legal o regulatorio, de usabilidad, fiabilidad, rendimiento, soporte y de mantenimiento. Es importante considerar que la redacción de cualquier documento empleado para definir las características del sistema, que será presentado al cliente, debe redactarse de forma simple, evitando el uso de terminología técnica, con el objetivo de que el cliente entienda claramente lo que se está proponiendo, y con ello evitar errores en los datos de entrada del sistema. Para lograr lo anterior, se recomienda emplear casos de uso en combinación de prototipos visuales, que ayuden a mostrar el funcionamiento del sistema y sus características, y a definir los requerimientos con mayor claridad.

La etapa final en la administración de requerimientos es la administración de cambios de los mismos. A lo largo del tiempo de desarrollo del sistema pueden existir cambios en algunos de los requerimientos, lo que posiblemente tenga impacto en uno o más requerimientos asociados a este. Para minimizar tales impactos, debemos asegurarnos de que al definir los requerimientos del sistema, estos tengan una estructura resistente a los cambios y emplear ligas que relacionen a cada requerimiento con cada fase del ciclo de desarrollo del sistema.

La administración del cambio incluye actividades como el establecimiento de una línea base de trabajo, determinación de dependencias entre módulos y requerimientos, la trazabilidad entre objetos interconectados y el control de los cambios.

1.6. Planeación para administrar requerimientos

1.6.1. Plan de administración de requerimientos

Establecer una correcta administración de requerimientos es indispensable para poder desarrollar un sistema de información de forma eficiente y libre de errores. Un plan de administración de requerimientos tiene que realizar con las siguientes funciones:

1. Identificar los requerimientos formales (no los documentos).
2. Identificar los tipos de documentos (cada tipo de documento, maneja de forma predeterminada un tipo de requerimiento formal).
3. Por cada uno de los requerimientos formales, identificar atributos:
 - Prioridad
 - Riesgo
 - Dificultad
 - etc.
4. Realizar matriz de trazabilidad, para saber cómo rastrear los cambios.

El primer punto se enfoca en el uso de diversas técnicas y herramientas que permitan a los desarrolladores identificar y establecer los requerimientos del sistema. Entre estas técnicas y herramientas tenemos la entrevista, cuestionarios,

casos de uso, diagrama de flujo de datos, etc. (en la unidad 2, se verán con mayor detalle algunas de estas técnicas).

La identificación del tipo de documento se refiere al tipo de documento que va a estar asociado a cada requerimiento que sea definido; de manera estandarizada, se emplea una plantilla de documento definida bajo la norma [IEEE-STD-830-1998](#), denominada “Especificación de Requerimientos del Software (Software Requirements Specification, SRS). Este documento sirve a los desarrolladores para facilitar la comunicación entre clientes y desarrolladores; como un contrato legal entre desarrolladores y clientes; como base para realizar las estimaciones del proyecto en costos, tiempo y magnitud; como una herramienta de evaluación del sistema ya terminado; y para tener una base para el control de cambios.

En cuanto a la trazabilidad se refiere, el documento SRS, permite contar con una fuente de información base que permite trazar los cambios. Entre las características que contiene este documento y que ayuda a hacer la trazabilidad, tenemos que:

Debe identificar las fuentes de los objetivos, requerimientos, y presunciones.

Debe relacionar requerimientos y presunciones con los objetivos.

Debe facilitar referenciar requerimientos en documentación futura (diseño, test, casos de test)

Un plan de administración de requerimientos ayuda a responder a las siguientes preguntas:

- ¿Cómo deben usarse las herramientas de gestión de requerimientos?
- ¿Qué tipos de requerimientos serán trazados en el proyecto?
- ¿Cuáles son los atributos de estos requerimientos?



- ¿Dónde se crearán los requerimientos, en una base de datos o solo en documentos?
- ¿En qué requerimientos necesitamos implementar trazabilidad?
- ¿Qué documentos se requieren?
- ¿Qué requerimientos y documentos serán utilizados como contrato con un proveedor?
- ¿Qué metodología será utilizada?
- ¿El cliente necesita algún documento específico para cumplir con su proceso de desarrollo?
- ¿Cómo se implementará la gestión de cambios?
- ¿Qué proceso garantizará que todos los requerimientos serán implementados y comprobados?
- ¿Qué requerimientos u opiniones necesitamos para generar informes? Ibáñez ([Gestión de requerimientos VI](#), 2010)

1.6.2. Definición de atributos de los requerimientos

Como se mencionó en el punto anterior, la definición de los atributos de los requerimientos es parte esencial del plan de administración de requerimientos; entre los atributos que pueden identificarse al definir los requerimientos de un sistema de información encontramos:

- *Disponibilidad.* Define la cantidad de tiempo que el sistema se encuentra trabajando.
- *Integridad conceptual.* Define la consistencia del sistema, incluyendo el diseño de sus componentes y módulos, la codificación de los mismos y la definición y uso de variables.
- *Flexibilidad.* Se relaciona con la capacidad del sistema para adaptarse a los cambios de ambiente y en políticas y reglas del negocio.



- *Interoperabilidad.* Hace referencia a la capacidad del sistema para interactuar con componentes de otros sistemas para intercambiar información de forma correcta.
- *Capacidad de mantenimiento.* Hace referencia en la capacidad de los componentes del sistema para permitir cambios en sus características cuando se realiza una actualización del sistema, se realizan cambios en algunas funciones, o se registran cambios en los requerimientos.
- *Capacidad de administración.* Se enfoca en la facilidad de administración del sistema, este atributo generalmente hace uso de herramientas que facilitan su monitoreo para el mejoramiento de su rendimiento y detectar errores.
- *Rendimiento.* Permiten medir la capacidad de respuesta del sistema al ejecutar algunas acciones en un tiempo determinado.
- *Confiabilidad.* Permiten mantener al sistema operacional todo el tiempo. Generalmente se mide a través de la probabilidad de que el sistema se mantenga funcional al ejecutar las funciones para las que fue diseñado.
- *Escalabilidad.* Permiten al sistema soportar cambios en su funcionamiento al instalar nuevas funciones o al aumentar su carga de trabajo.
- *Seguridad.* Permite al sistema estar protegido contra la pérdida de información y garantizar su funcionamiento.
- *Riesgo.* Define qué tan vulnerable es un requisito o qué tan difícil será su uso.
- *Prioridad.* Define el nivel de importancia de cada requerimiento.

RESUMEN

La administración de requerimientos es una de las actividades más importantes dentro del ciclo de vida de los sistemas de información e involucra a desarrolladores, clientes y usuarios.

La administración de requerimientos conjuga una serie de actividades que comienza con la detección de las necesidades del negocio, esto se logra a través del establecimiento de un plan de negocios, principalmente de la planeación estratégica. Permiten identificar los recursos y carencias que tiene la organización que ayudarán a definir las características que se desean incorporar al sistema, las carencias de la organización, sus recursos y establecer las necesidades que a la postre, definirán a los requerimientos del sistema.

Dentro del proceso de administración, es importante clasificar los requerimientos, dividiendo los funcionales y los no funcionales, para poder darles una prioridad y establecer con ellos las características del sistema y definirlo.

En la definición del sistema, es importante documentar los requerimientos preferentemente bajo un documento estandarizado como lo es la norma IEE-STD 830-1998 *Especificación de Requerimientos del Software (SRS)*, donde se describe a detalle cada uno de ellos y sus atributos, y que sirve como base para el control de cambios, ya que dentro de cada documento se especifican las asociaciones de cada requerimiento con los diversos módulos del sistema o con otros requisitos, que facilitará realizar el rastreo e impacto de cada uno de los cambios solicitados y realizados.

BIBLIOGRAFÍA



SUGERIDA

Bruegge, Bernd y Dutoit, Allen H. (2002). *Ingeniería de software orientada a objetos*. México: Pearson.

Joyanes Aguilar, Luis. (2003) *Fundamentos de programación: algoritmos, estructuras de datos y objetos*. (3ª ed.) Madrid / México: McGraw-Hill.

Pfleeger, Shari L. (2002). *Ingeniería de software: teoría y práctica*. México: Prentice Hall.

Piattini Velthuis, Mario y García, Félix (coords.). (2003). *Calidad en el desarrollo y mantenimiento de software*. México: Alfaomega / Ra-Ma.

Piattini Velthuis, Mario; Calvo-Manzano Villalón, José; Cervera Bravo, Joaquín y Fernández Sanz, Luis. (2003). *Análisis y diseño de aplicaciones informáticas de gestión*. México: Alfaomega / Ra-Ma.

Pressman, Roger S. (2002) *Ingeniería de software: un enfoque práctico*. (5ª. ed.) México / Madrid: McGraw-Hill.

Sommerville, Ian. (2001) *Ingeniería de software*. (6ª ed.) México: Pearson.



Weitzenfield, Alfredo. (2003) *Ingeniería de software orientada a objetos con UML, Java e Internet*. México: Thomson.



UNIDAD 2

Identificación de requerimientos



OBJETIVO ESPECÍFICO

Seleccionará y aplicar los métodos y las técnicas más apropiadas para identificar los requerimientos para la construcción de un sistema.

TEMARIO DETALLADO

(24 horas)

2. Identificación de requerimientos

2.1. Métodos y técnicas estructuradas para la identificación de requerimientos

2.1.1. Entrevista

2.1.2. Cuestionario

2.1.3. Método Delphi

2.1.4. Desarrollo Conjunto de Aplicaciones

2.1.5. Diagrama Causa-Efecto de Ishikawa

2.2. Métodos y técnicas no estructuradas para la identificación de requerimientos

2.2.1. Observación



2.2.2. Revisión de documentos de la organización

INTRODUCCIÓN

La etapa de análisis es fundamental para el buen desarrollo de cualquier sistema de software. En esta etapa se recaba toda la información necesaria para la construcción del sistema, partiendo de los requisitos mínimos para su funcionamiento, hasta la determinación de todas las variables que pueden afectar su funcionalidad y limitar su rendimiento.

Es conveniente contar con personal de experiencia en el análisis y selección de requerimientos para evitar omisiones en la recopilación de datos y asegurar mayor fiabilidad de los mismos.

En esta unidad, se revisarán algunas de las técnicas de recolección de información, que ayudan a los desarrolladores de software a identificar los requerimientos de los sistemas de información.

2.1. Métodos y técnicas estructuradas para la identificación de requerimientos

Hemos visto en la unidad 1 que un requerimiento manifiesta las características solicitadas por los clientes que debe tener un sistema para atender y resolver algún problema en particular; y se definen a partir del análisis de las necesidades que se haga del sistema que se desea.

Ahora bien, para que se puedan especificar las acciones para la construcción del sistema, es importante que el analista distinga los tipos de requerimientos que necesita, es decir los funcionales y los no funcionales, ya que ello dará pauta para establecer su funcionalidad y sus limitantes.

Una forma de poder identificar los requerimientos funcionales y no funcionales de un sistema es mediante el empleo de métodos de recopilación de información que pueden ser aplicados a los clientes y usuarios del sistema (fuente primaria de información). Existen diferentes métodos que permiten obtener información amplia y exacta para la construcción de los sistemas. Entre estos se encuentran los métodos estructurados y los no estructurados.

Dentro del ciclo de desarrollo de un sistema, los métodos estructurados para la identificación de requerimientos permiten a los desarrolladores conceptualizar un modelo inicial del sistema que se desea construir. Estos métodos fueron desarrollados en la década de 1970 y han ayudado a dar soporte a las etapas de análisis y diseño de software.

Los métodos estructurados, normalmente definen un proceso que puede ser empleado para la generación de modelos de sistemas, proporcionando un marco detallado de información para el análisis de requerimientos.

Los métodos estructurados parten de un estándar para su elaboración, contrario a los no estructurados. Entre estos podemos ubicar el método Delphi y los diagramas causa-efecto de Ishikawa, los que ayudarán a la identificación de los requerimientos mediante una serie de acciones definidas dentro de cada uno.

A pesar de la utilidad de los métodos estructurados, tienen una serie de desventajas que hay que observar:

1. No proporcionan un soporte efectivo para la comprensión o el modelado de requerimientos no funcionales del sistema.
2. No discriminan, ya que no incluyen guías que ayuden a los usuarios de estos métodos a decidir si un método es adecuado para un problema concreto.
3. A menudo generan documentación excesiva que puede causar que el requerimiento en sí pierda su objetividad.
4. Los modelos producidos suelen ser muy detallados y difíciles de comprender para el usuario. (Sommerville, 2001, p.170)

En la práctica, los desarrolladores e ingenieros no están restringidos a un método en particular y pueden emplear varios de ellos, según sean sus necesidades.

Entre las técnicas comúnmente empleadas para recabar información de forma estructurada, tenemos a los cuestionarios, las entrevistas, el JAD (Desarrollo

Conjunto de Aplicaciones) y los diagramas de causa-efecto. Estas nos permitirán detectar ideas, necesidades, preferencias, hábitos, problemas, causas, entre otros.

2.1.1. Entrevista

Las entrevistas son una técnica general y sencilla de recabar información directa de personas o grupos (se centra en la persona y su trabajo), donde por lo regular, los entrevistados forman parte del grupo de usuarios del sistema que se va a desarrollar.

Debido a que las entrevistas son laboriosas y lleva tiempo su aplicación, no son siempre el método más empleado entre los diseñadores, pero son una forma muy confiable de recopilación de información, ya que tiene la ventaja de ser de individuo a individuo, dando pauta para observar reacciones corporales y tonos de voz.

Las entrevistas pueden clasificarse como estructuradas o no estructuradas. Las *entrevistas no estructuradas* (o entrevistas no dirigidas) utilizan un formato pregunta-respuesta y son apropiadas cuando el analista desea adquirir información general del sistema. Este formato anima a los entrevistados a compartir sus sentimientos, ideas y creencias.

Por otro lado, las *entrevistas estructuradas* (entrevistas dirigidas) utilizan preguntas estándar en un formato de respuesta abierta o cerrada. El primero permite que el entrevistado dé respuesta a las preguntas con sus propias palabras; el segundo utiliza un conjunto anticipado de respuestas. Cada enfoque tiene sus ventajas y desventajas (Senn, [Análisis](#)), aunque en la práctica se puede encontrar que manejan los dos tipos de entrevista a la vez.

Los pasos para preparar una entrevista son los siguientes pasos:

1. **Estudio del dominio del problema o tema.** Es necesario que el entrevistador revise la temática que se va a tratar, para ello, es posible que consulte documentos de proyectos o situaciones similares; también, es recomendable que revise los perfiles de los usuarios para abordarlos de mejor manera y ajustar las preguntas de acuerdo con su categoría.
2. **Selección de los entrevistados.** Es necesario evitar realizar entrevistas que resulten en mucha información y sobre todo redundante, por lo que es recomendable limitar el número de entrevistas y seleccionar un grupo representativo de cada tipo de usuarios. Se recomienda comenzar con los directivos de las organizaciones para poder establecer una visión general de lo que se desea en el sistema.
3. **Establecimiento de objetivos y contenido de la entrevista.** Es necesario que el entrevistador establezca los objetivos de cada entrevista y determine qué contenido debe tener cada una, es recomendable que el entrevistador envíe a los candidatos a ser entrevistados un documento que ayude en la introducción del proyecto y un cuestionario para ser rellenado y devuelto por los candidatos, con la finalidad de que el entrevistador recopile información de utilidad para la entrevista.

Una entrevista se desarrolla en tres etapas principales:

1. **Apertura.** Durante esta etapa el entrevistador se presenta e informa al entrevistado sobre el objetivo de la entrevista, lo que se pretende conseguir, el uso que se le dará a la información, etc.
2. **Desarrollo.** Se recomienda que la entrevista se desarrolle dentro de un tiempo no mayor a 2 horas, tratando de distribuir el tiempo, dándole 80% como máximo al entrevistado y 20% al entrevistador.



Es recomendable que el entrevistador tenga control completo sobre el proceso para evitar caer en monólogos, contemplando la intervención de una tercera persona para la documentación de la entrevista. Dentro de esta etapa, se sugiere seguir lo siguiente:

Empleo de preguntas abiertas.
Permite al entrevistado expresarse más libremente sin restricciones, unos ejemplos de estas preguntas son: ¿Cuál es el proceso de registro una compra?, ¿Cómo doy seguimiento a un pedido de los clientes?

Empleo de un lenguaje apropiado.
Es deseable que las preguntas estén elaboradas en un lenguaje simple, libre de tecnicismos que pueda no ser entendido por el entrevistado.

Mostrar interés en todo momento.
Se debe cuidar el lenguaje corporal como son movimientos corporales, tono de voz, expresiones faciales, etc.

3. **Cierre.** Es deseable recapitular los conceptos abordados en la entrevista con el objetivo de que la información recabada esté libre de errores, agradecer al entrevistado y dejar abierta la posibilidad de realizar futuras entrevistas.

Una vez realizada la entrevista, se procede al análisis de la misma, para ello, es necesario leer las notas obtenidas, pasar en limpio la información, revisarla y contrastarla con otras fuentes. Es deseable, que una vez obtenido el reporte en limpio, este sea enviado al entrevistado para la corroboración del mismo (véase, Durán, 2000, pp. 20-22).



Con la información obtenida, se pueden comenzar a identificar los diferentes tipos de requerimientos y se puede comenzar con la elaboración del documento de especificación de requerimientos (SRS).

El empleo de entrevistas es la metodología más usual en el desarrollo de sistemas, ya que a través de ellas generalmente se inicia el contacto con todas las partes involucradas en el proyecto.

2.1.2. Cuestionario

El uso de los cuestionarios permite a los analistas reunir información relacionada con varios aspectos de un sistema de un grupo grande de personas, es aplicable tanto a los clientes como a los usuarios finales y sirve como método de obtención de información directo.

El empleo de formatos estandarizados para las preguntas puede proporcionar datos más confiables que otras técnicas; por otra parte, su amplia distribución asegura el anonimato de los encuestados, situación que puede conducir a respuestas más honestas. Sin embargo, este método no permite al analista observar las excepciones o reacciones de los encuestados. Asimismo, la respuesta puede ser limitada ya que es posible que no tenga mucha importancia para los encuestados llenar el cuestionario (véase, Senn, [Cuestionarios](#)).

El cuestionario tiene ciertas características básicas, como son:

- Es un procedimiento de investigación.
- Es una entrevista altamente estructurada.
- Un cuestionario consiste en un conjunto de preguntas respecto a una o más variables a medir.
- Presenta la ventaja de requerir relativamente poco tiempo para reunir información sobre grupos numerosos.
- El sujeto que responde, proporciona por escrito información sobre sí mismo o sobre un tema dado.
- Presenta la desventaja de que quien contesta responde escondiendo la verdad o produciendo notables alteraciones en ella. Además, la uniformidad de los resultados puede ser aparente, pues una misma palabra puede ser interpretada en forma diferente por personas distintas, o ser comprensibles para algunas y no para otras. Por otro lado, las respuestas pueden ser poco

claras o incompletas, haciendo muy difícil la tabulación. (véase, Osorio Rojas, [El cuestionario](#), 2002)

Existen diversos tipos de cuestionarios. El primero y tal vez el más simple de generar es el cuestionario restringido o cerrado.

El *cuestionario restringido* se caracteriza por contener preguntas específicas y delimitadas que anticipan la respuesta, en algunos casos pueden restringirse a respuestas como “sí” o “no”, “verdadero o falso”, donde se tienen preguntas con varias opciones de respuestas donde el encuestado selecciona una o varias opciones.

La ventaja de este tipo de cuestionarios radica en la facilidad de evaluar las respuestas y su objetividad, ya que difícilmente se sale del tema en cuestión.

Otro tipo de *cuestionario* es el *abierto*, donde las preguntas se formulan de forma más natural, dejando que el encuestado exprese más abiertamente sus ideas y con mayor profundidad, estos cuestionarios son útiles cuando se busca recabar información de algún tema poco conocido.

La desventaja que se presenta al aplicar un cuestionario abierto es que requieren de más tiempo de análisis de las respuestas para su tabulación, resumen e interpretación.

Existen cuestionarios que combinan a ambos tipos, dependiendo del objetivo que se persiga al formularlos.

A continuación, mostramos algunos aspectos a considerar para elaborar un buen cuestionario:



- Hacer una lista de aspectos (variables) que se consideran importantes de incluir.
- Determinar el propósito del cuestionario. Se refiere a un tema significativo.
- Señalar el título del proyecto, del aspecto o tema a que se refiere, y una breve indicación de su contenido. Las instrucciones deben ser claras y completas.
- Especificar algunos datos generales: Institución, fecha, nombre del encuestador, etc.
- Establecer la mejor secuencia de dichos aspectos o temas.
- Los términos importantes deben estar definidos.
- El cuestionario no ha de ser demasiado largo.
- No es conveniente iniciar el cuestionario con preguntas difíciles o muy directas.
- Escribir un esquema de posibles preguntas pensando lo que se pretende averiguar con cada una de ellas, procediendo posteriormente, si es necesario, a su reubicación, modificación o eliminación. Cada pregunta implica una sola idea. Las preguntas deben ser objetivas, es decir, sin sugerencias hacia lo que se desea como respuesta. Con relación a este punto, es conveniente hacerse las siguientes interrogantes:
 - ¿Es necesario o útil hacer esta pregunta?
 - ¿Es demasiado general?
 - ¿Es excesivamente detallada?
 - ¿Debería la pregunta ser subdividida en otras preguntas más pequeñas y ser más concreta, específica?
 - ¿La pregunta se refiere preferentemente a un solo aspecto?
 - ¿Se refiere a un tema sobre el cual las personas encuestadas poseen la información necesaria?
 - ¿Es posible contestarla sin cometer errores?
 - ¿Son las palabras suficientemente simples como para ser comprendidas por el encuestado?
 - ¿Es la estructura de la frase fácil y breve?
 - ¿Son las instrucciones claras y precisas?
 - ¿Es necesario clarificarla con alguna ilustración?
 - ¿Es posible que tal pregunta incomode al encuestado?
 - ¿La pregunta induce la respuesta? ("Las preguntas no pueden apoyarse en instituciones, ideas respaldadas socialmente ni en evidencia comprobada"). (Osorio Rojas, 2010)

Como ya se mencionó en el tema de la entrevista, los cuestionarios pueden ser de ayuda para su preparación o para recabar información de un gran número de

usuarios para identificar requerimientos desde su punto de vista, al igual que la entrevista, es deseable resumir y pasar en limpio la información para analizarla de forma más eficiente.

2.1.3. Método Delphi

El método DELPHI es un método estructurado de comunicación grupal que permite a un grupo de individuos (generalmente especialistas en el tema), analizar y resolver un problema complejo. Las características principales de este método son:

- Mantenimiento del anonimato de los participantes.
- La retroalimentación controlada.
- La respuesta estadística grupal.

En esencia, el método consiste en recabar las opiniones de todos los participantes del grupo sobre la problemática en cuestión; el anonimato ayuda a que las opiniones no sean sesgadas y permite que se tomen en consideración diversos puntos de vista que ayuden a enriquecer el conocimiento organizacional y a resolver el problema de forma efectiva, el método se repite cuantas veces sea necesario hasta encontrar una solución adecuada.

Cuando se emplea la metodología Delphi, se emplea la siguiente terminología.

| | |
|---------------------|--|
| Circulación | Se trata de la sucesión de cuestionarios que se aplican a los integrantes del grupo. |
| Cuestionario | Se trata de los documentos que se hacen llegar a los integrantes del grupo. Los cuestionarios no solo contienen una serie de preguntas como en los tradicionales, sino que en Delphi, se usan como las |



| | |
|------------------|---|
| | herramientas de interacción entre los participantes, ya que en ellos se presentan los resultados de las circulaciones realizadas. |
| Panel | Se trata del conjunto de participantes en el grupo de trabajo, se recomienda que los integrantes sean expertos en el tema. |
| Moderador | Es la persona encargada de regular las interacciones del grupo, prepara los cuestionarios, recoge las respuestas de cada circulación y realiza las circulaciones. |

Antes de comenzar a emplear la metodología Delphi, se recomienda realizar previamente las siguientes actividades:

1. Delimitar el contexto y establecer los tiempos de duración de cada circulación.
2. Realizar la selección del panel o grupo, procurando que sean expertos en el área donde se presenta la problemática, es recomendable que los expertos establezcan un compromiso con el trabajo y que procuren dar respuestas universales para evitar el sesgo en la información que se pretende obtener.
3. Explicar a los participantes la metodología de trabajo de Delphi y sus objetivos para que estos se preparen adecuadamente y expongan previsiones fiables.

La metodología Delphi, contempla cuatro circulaciones como base para poder alcanzar un resultado en consenso; pero si no se logra lo anterior, se procede a realizar más circulaciones.

Las cuatro circulaciones de Delphi iniciales son las siguientes:

① Primera circulación.

En esta circulación se genera un cuestionario sin estructura (más abierto), con el objetivo de que los participantes expongan sus argumentos sobre los eventos y las tendencias más importantes en el área del tema en cuestión. Al final de la circulación, el moderador recoge los cuestionarios y sintetiza la información, seleccionando aquellos eventos coincidentes que permiten una mejor definición y manejo, a partir de esos eventos se estructura el cuestionario de la segunda circulación.

② Segunda circulación.

Los cuestionarios estructurados con los eventos obtenidos de la primera circulación son proporcionados a los participantes y se les pide que den un pronóstico sobre la fecha en que ocurrirán dichos eventos. Después de responderlos, el moderador recoge los cuestionarios y, de nueva cuenta, analizará y sintetizará la información. Ésta se analiza de forma estadística centrando los resultados en la mediana (la mediana representa que hay un 50% de probabilidad de que el evento suceda en la fecha señalada), el primer cuartil o valor inmediato inferior representa la probabilidad del 25% y el cuartil superior representa el 75%. La información obtenida del análisis estadístico es la que va a servir de base para estructurar el siguiente cuestionario, que incluirá la lista de eventos obtenidos de la primera circulación y sus valores estadísticos obtenidos de la segunda.



③ Tercera circulación.

El grupo recibe el cuestionario obtenido de la circulación anterior y se les solicita que realicen nuevos pronósticos. Si el nuevo pronóstico se reafirma y queda entre los cuartiles superior o inferior, se pide al participante que explique las razones de su pronóstico y que diga las razones por las que cree que los pronósticos de los demás participantes son incorrectos. La ventaja del método Delphi, al garantizar el anonimato de los participantes, es que permite que se expresen de forma más abierta y libre. El moderador recoge los resultados y realiza un nuevo análisis estadístico de ellos, organiza los argumentos de los participantes que dieron un pronóstico por arriba o debajo de la mediana. El cuestionario que será preparado para la siguiente circulación contendrá el nuevo análisis estadístico y los argumentos de los participantes.

④ Cuarta circulación.

En esta circulación se solicita a los participantes leer los argumentos presentados y realizar un nuevo pronóstico; adicionalmente, se les solicita a los participantes dar una opinión sobre las opiniones discrepantes. Al finalizar, de nueva cuenta el moderador analiza la información y sintetiza las opiniones.

Cuando las opiniones y los pronósticos tienden a centralizarse en un valor, se puede decir que el proceso ha concluido, solamente queda pendiente que el moderador elabore un reporte con las fechas de los pronósticos y los comentarios realizados. De no existir coincidencias en los comentarios y pronósticos el método continúa con más circulaciones hasta alcanzar un consenso.

En la siguiente dirección electrónica, encontrarás un ejemplo de la aplicación del método Delphi presentado en el congreso EDUCTEC, llevado a cabo en Pachuca, Hidalgo en el 2011: <http://www.edutec.es/revista/index.php/edutec-e/article/viewFile/187/18>.

Cabero, J. & Infante, A. Empleo del método Delphi y su empleo en la investigación en comunicación y educación. EDUTEC, Revista Electrónica de Tecnología Educativa, 48.

Recuperado el 30/05/2017

2.1.4. Desarrollo conjunto de aplicaciones

El desarrollo conjunto de aplicaciones (*Joint Application Development, JAD*), es una **técnica** desarrollada por la IBM que se basa en la entrevista y se apoya en la dinámica de grupos. La técnica consiste en realizar una serie de reuniones en un periodo de tiempo comprendido entre 2 y 4 días, y se basa en los siguientes principios:

1. **Dinámicas grupales**, que permiten la integración del grupo de trabajo.
2. **Uso de ayudas audiovisuales**. Algunos problemas o conceptos requieren de un soporte audiovisual como diagramas, diapositivas, pizarrones, etc., con el objetivo de tener una visión amplia del problema y mejorar su comprensión.
3. **Modo de trabajo sistemático**. Los participantes definirán la forma de trabajo y la forma de analizar y dar solución al problema, este modo de trabajo debe ser aceptado por todos los participantes.
4. **Generación de documentos bajo el modelo WYSIWYG**. Todos los documentos generados durante las actividades de trabajo son previsualizados de forma inmediata, dando una idea de la forma como quedará el contenido final.

Cuando se trabaja bajo la metodología JAD, se tiene dos partes: el JAD/Plan que se enfoca a la obtención de requerimientos y el JAD/Design, enfocada al diseño de la aplicación, en nuestro caso, nos enfocaremos a la primer parte de la metodología, el JAD/Plan.

La metodología JAD cuenta con un grupo de participantes donde se tienen de ocho a doce usuarios finales del sistema y los desarrolladores, los participantes tienen asignado un rol, el cual puede ser alguno de los siguientes:

1. **Jefe de JAD.** Es el encargado de dirigir las reuniones que se realicen, es deseable que cuente con características de liderazgo y facilidad de comunicación, también debe de ser capaz de lidiar con los conflictos que se puedan llegar a presentar para evitar que las sesiones se alejen de los objetivos.
2. **Analista.** Es el encargado de recabar la información resultante de las sesiones y ponerla por escrito, es deseable que el analista tenga un dominio pleno del tema en cuestión y sepa utilizar de forma adecuada las herramientas de apoyo elegidas por el grupo.
3. **Patrocinador.** Se trata de los clientes, generalmente los dueños o dueño del negocio, que tengan la capacidad de decidir sobre el desarrollo del sistema. Su función principal es la de informar sobre las necesidades que debe cubrir el sistema.
4. **Representantes de los usuarios.** Son un grupo de usuarios finales del sistema de los clientes que aportarán una visión general de lo que debe realizar el sistema.
5. **Desarrolladores.** Se trata de un grupo de personas con experiencia en el desarrollo de software, su función principal será identificar e informar de problemas en la implementación de las características del sistema.
6. **Especialistas.** Son personas que tienen conocimientos y experiencia suficientes como para considerarlos una autoridad en el ámbito de desarrollo de sistemas. Su función es asesorar sobre aspectos específicos tanto de parte de los usuarios como de parte de los desarrolladores.

Una vez determinados los roles, se puede proceder a realizar las fases de trabajo. En el JAD/Plan podemos encontrar tres fases principales:

1. Adaptación. Durante esta fase, la carga de trabajo recae sobre el jefe del JAD, quien es el responsable de adaptar las sesiones de trabajo de acuerdo con las características del proyecto. Durante esta fase el jefe del JAD tiene las siguientes actividades:

- Obtener información sobre la empresa y el proyecto que se va a desarrollar.
- Organizar las sesiones considerando el lugar, duración, número de sesiones y los participantes de cada sesión. Es deseable que las sesiones sean programadas fuera de la empresa para evitar cualquier tipo de distracción.

2. Sesiones JAD. Una vez que el jefe del JAD decide la forma de llevar a cabo las reuniones, éstas se desarrollan conforme los siguientes pasos:

- *Presentación.* Tanto el jefe del JAD como el patrocinador se presentan ante el grupo de trabajo. El patrocinador debe explicar los motivos y características generales del proyecto y el jefe del JAD explicará la mecánica de la reunión.
- *Definición de requerimientos de alto nivel.* Durante este paso se comienza a recabar información clave para el desarrollo del sistema, se da respuesta a preguntas como ¿qué beneficios se esperan del sistema?, ¿cuáles son los recursos disponibles?, ¿en qué tiempos se debe desarrollar?, etc. La información obtenida se presenta en una pizarra o en algún medio que permita que todos los participantes puedan verla.
- *Definición del ámbito del sistema.* Una vez recabados los requerimientos, se puede proceder a definir las características y alcances del sistema.
- *Documentación de temas abiertos.* Cuando se generan temas que quedan sin resolver en las sesiones de trabajo, estos deben de ser documentados y se debe de asignar a un responsable para su análisis y solución, que serán discutidos en otra sesión de trabajo.

- *Conclusión.* El jefe del JAD recopila la información de la sesión y realiza un repaso de la misma ante el grupo; durante este paso se realizan correcciones o añadidos a la información recabada.
- 3. Organización de la documentación.** Una vez recabada la información en la sesión de trabajo, el jefe del JAD debe de generar un documento formal, para ello se siguen los pasos de:
- *Compilación.* La información generada se debe de redactar en un documento estandarizado como el SRS. “(Especificación de Requerimientos del Software, SRS -Software Requirements Specification)”¹
 - *Revisión.* Una vez generado el documento, se envía a los participantes de la sesión para su revisión y corrección.
 - *Validación.* El patrocinador revisa el documento y se encarga de decidir si lo aprueba o no.

El proceso se repite cuantas veces sea necesario para recopilar todos los requisitos del sistema que se va a desarrollar.

El empleo de la metodología JAD presenta las siguientes ventajas en el proceso de desarrollo de sistemas:

- Los requerimientos son validados casi de forma inmediata, ya que al estar todos los involucrados con el sistema, la información se puede corregir y validar más rápidamente y de forma concreta.
- La inclusión de clientes y usuarios finales reduce la resistencia al cambio ya que se les involucra activamente en el proceso de desarrollo.

¹ Consulta un ejemplo de documento SRS en la siguiente dirección electrónica: <http://es.scribd.com/doc/63229261/Documento-SRS-Final>



El empleo de la metodología JAD, si bien es eficiente e incluyente, también puede ser difícil de implementar, principalmente debido a que los participantes no siempre están disponibles al mismo tiempo para organizar las sesiones de trabajo (Durán, 2002, pp. 20-22).

2.1.5. Diagrama causa-efecto de Ishikawa

El diagrama de causa-efecto de Ishikawa, también conocido como “de pescado” por su particular parecido a un esqueleto de pescado, es una forma gráfica de representar las diferentes teorías de las causas que provocan un cierto problema. Este tipo de diagrama se emplea generalmente para realizar diagnósticos sobre las problemáticas que se presentan y para analizar y solucionar sus posibles causas.²

La técnica requiere de la formación de un grupo de trabajo de especialistas sobre el ámbito del problema, posteriormente se realiza el trazado de un rectángulo donde se escribe el problema, colocando del lado izquierdo las principales causas que pueden ocasionar dicho problema, y del lado derecho, se colocan los efectos derivados del problema.

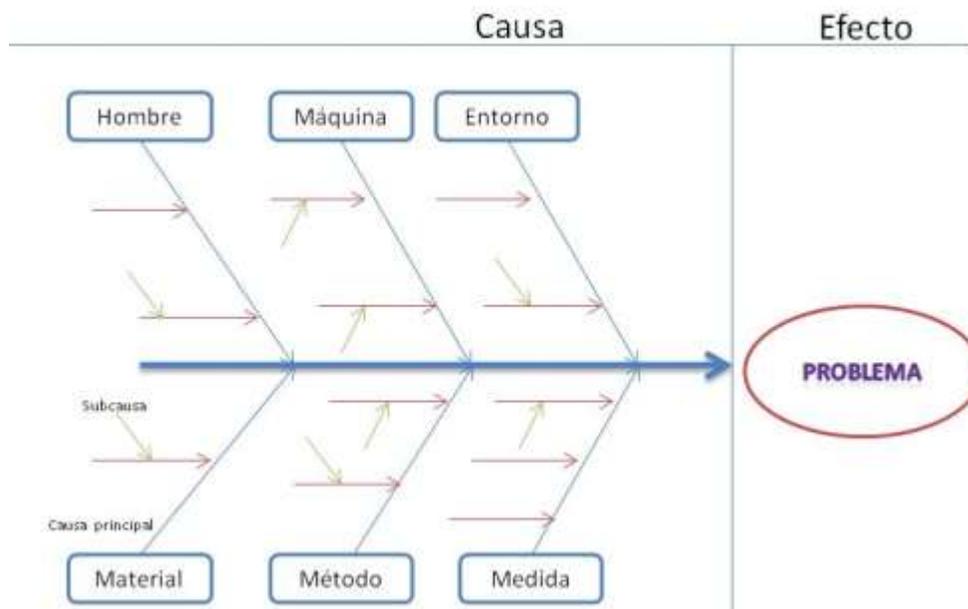


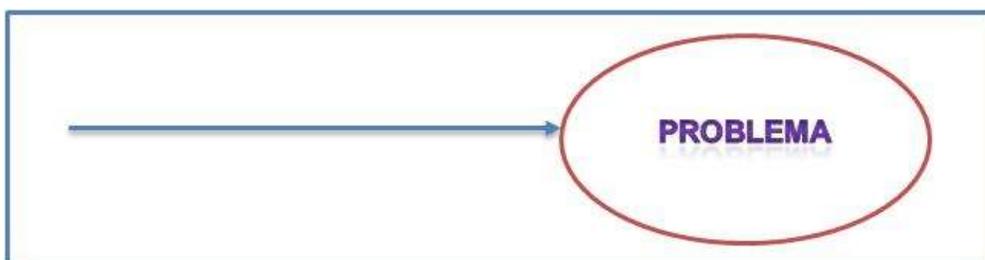
Figura 2.1 Diagrama de causa y efecto de Ishikawa ([Wikipedia](https://es.wikipedia.org/wiki/Diagrama_de_causa_y_efecto))

² Aunque la mayoría de los diagramas de causa efecto tienen esta presentación también pueden tener otra estructura, ve más modelos en: http://www.educationoasis.com/curriculum/GO/cause_effect.htm. Consultado el 18 de octubre de 2012.

Uno de los principales problemas que encontramos en los diagramas causa-efecto es que las causas por lo general son mutuamente excluyentes, ya que no se relacionan entre sí; esta situación puede ser solventada tratando de encontrar relación entre las causas y escribirlas en el mismo diagrama, con el objetivo de obtener una mejor perspectiva del problema.

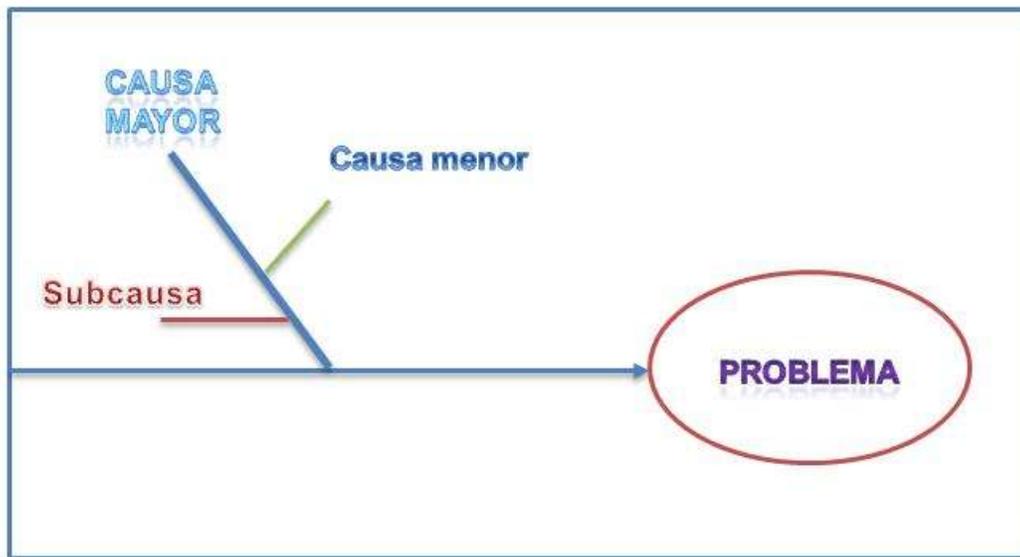
El uso de diagramas de causa-efecto tiene tres pasos principales:

1. **Construcción del diagrama.** Este paso se debe de realizar siguiendo los siguientes pasos:
 - *Formación del grupo de trabajo.* Se integra un grupo de trabajo con especialistas en el contexto del problema, con un conocimiento relativamente profundo del problema. Es deseable que el grupo no exceda los 15 participantes. Dentro del grupo de trabajo se debe de nombrar a un facilitador que se encargue de la organización y dirección de la sesión de trabajo.
 - *Planteamiento del problema.* El facilitador debe explicar la forma de trabajo y solicita a los participantes que expliquen el problema. Se debe procurar especificar con el mayor detalle posible el problema para no generar ambigüedades. Una vez definido el problema, se dibuja un rectángulo donde se escribe el problema y se dibuja una flecha del lado izquierdo con sentido de izquierda a derecha, terminando con la punta señalando al problema.





- *Identificación de las posibles causas y su clasificación.* A través de una lluvia de ideas de los participantes, se busca establecer cuáles son las posibles causas que generen el problema en cuestión. Las causas identificadas en el proceso se van escribiendo en un pizarrón en forma de lista para su posterior clasificación calificándolas según su peso.
- *Agrupación de las causas de acuerdo a su clasificación.* Las causas identificadas en el paso anterior se agrupan de acuerdo con la clasificación asignada, buscando aquellas que tengan mayor impacto en el problema, este proceso también ayuda a que se identifiquen causas repetidas o similares e ir depurándolas.
- *Construcción del diagrama.* Una vez clasificadas las causas de mayor a menor, se procede a generar el diagrama, escribiendo las causas mayores en un extremo unido por una línea a la flecha dibujada señalando al problema. Las causas menores y subcausas identificadas se unen a las causas mayores a través de líneas unidas a la línea de la causa mayor. Es importante que los participantes estén totalmente convencidos de que las causas anotadas estén plenamente asociadas al problema.



2. **Identificación de las causas y efectos más probables.** Una vez dibujado el diagrama, se procede a seleccionar aquellas causas que sean más probables, lo anterior se puede realizar por votación directa. En este punto es importante que los participantes estén completamente convencidos de que las causas seleccionadas sean las de mayor impacto en el problema.
3. **Generación de posibles soluciones.** Una vez identificadas las causas principales del problema, se procede a proponer posibles soluciones, dicha soluciones se analizan y se genera un documento donde se exponga la o las soluciones más viables para su aplicación y evaluación.

El uso de diagramas de causa-efecto en el desarrollo de sistemas de información es una herramienta que permite identificar la problemática dentro de la organización que se desea solucionar y a detectar las soluciones más viables a implementar a través del sistema.

Puedes consultar varios ejemplos de casos de aplicación del diagrama causa-efecto de Ishikawa.



Ejemplos de causa-efecto (Sánchez Guerrero, 2003)

Como ya se ha mencionado, los desarrolladores no se encuentran sujetos al empleo de una técnica en particular o varias de ellas, el empleo de las técnicas dependerá en gran medida de las características del proyecto y del criterio de los desarrolladores.



2.2. Métodos y técnicas no estructuradas para la identificación de requerimientos

Los métodos no estructurados son métodos de recopilación de información sin una estructura previamente elaborada, sin un guión por así decirlo, como una encuesta o un cuestionario. Las técnicas más empleadas son la observación (participativa y dirigida), entrevistas (más profundas, como la de un psicólogo), el estudio de la documentación del negocio, etc.

2.2.1. Observación

La observación permite al analista ganar información que no se puede obtener por otras técnicas. Por medio de la observación el analista obtiene información de primera mano sobre la forma en que se efectúan las actividades. El método es más útil cuando el analista necesita observar, por un lado, la forma en que se manejan los documentos y se llevan a cabo los procesos y, por otro, si se siguen todos los pasos especificados. Los observadores experimentados saben qué buscar y cómo evaluar la significancia de lo que observan. (Senn, [Observación](#))

Ventajas y desventajas del empleo de la observación

Ventajas.

- No hay necesidad de intermediarios, se aprovecha el papel de cada participante en el proyecto para recolectar datos.
- Aporta información sobre prácticas reales en la organización.
- Permite verificar hechos.
- No se requiere de muchos recursos.
- Permite comprender la forma de interacción de los grupos observados y las normas de la organización.
- Permite una mejor comprensión del entorno del proyecto.
- El análisis de datos es más sencillo y objetivo.

Desventajas.

- Puede ser necesario aprender un idioma específico o tecnicismos para poder interactuar con las personas.
- Es posible pasar por alto datos debido a una mala planeación de la observación.
- Requiere de una inversión alta de tiempo.
- Algunos campos de estudio pueden ser difíciles o imposibles (violencia de género, prácticas sexuales, etc.)

La observación participativa

Se caracteriza por la existencia de un conocimiento previo entre observador y observado y una permisividad en el intercambio, lo cual da lugar a una iniciativa por parte de cada uno de ellos en su interrelación con el otro. El observado puede dirigirse al observador, y el observador al observado, en una posición de mayor cercanía psicológica pero con un nivel de participación bajo o nulo.

La observación participativa se refiere a una práctica que consiste en convivir con la gente que uno estudia, llegar a conocerlos, a conocer su lenguaje y sus formas de vida a través de una intrusa y continuada interacción con ellos en la vida diaria.

El mecanismo de la observación consiste en buscar siempre una regularidad en las interacciones y una amplitud de forma continuada, manteniendo y creando relaciones.

Las normas de la observación participante son:

1. No bajar la guardia dando las cosas por supuesta.
2. Prestar atención a los aspectos culturales de la situación.
3. Tener experiencias desde dentro y desde fuera.
4. Realizar un registro sistemático de la observación.

La observación dirigida o sistémica

Este tipo de observación se enfoca a un terreno o hechos específicos, objetos de estudio, alternando entre sesiones de observación en el campo o terreno de interés y sesiones de reflexión sobre lo observado, posterior a estas sesiones de reflexión se redactan las conclusiones sobre el campo en cuestión.

La observación sistémica requiere de una planificación donde se establezcan los tiempos de observación y los objetivos que se van a estudiar. Posterior a cualquier sesión de observación se debe realizar un trabajo de redacción a modo de diario, en donde se van a recoger todos los datos recopilados para su análisis formal.

La observación es una técnica que se emplea en el sitio donde va a estar funcionando el sistema, por lo que es una herramienta excelente para poder determinar requerimientos no funcionales que tienen que ver principalmente con el ámbito del sistema, una de las ventajas que presenta la observación, es que

cualquiera de los miembros del equipo de desarrollo (desarrolladores), puede aplicarla, previa planificación por parte de todos los participantes incluyendo a los clientes.

2.2.2. Revisión de documentos de la organización

Según García Gutiérrez:

de reseñas. (en Dulzaides y Molina, 2004) El análisis documental es una forma de investigación técnica, un conjunto de operaciones intelectuales, que buscan describir y representar los documentos de forma unificada y sistemática para facilitar su recuperación. Comprende el procesamiento analítico-sintético que, a su vez, incluye la descripción bibliográfica y general de la fuente, la clasificación, indización, anotación, extracción, traducción y la confección

En el caso de la recopilación de información para los sistemas de información, este tipo de análisis nos ayuda a recabar información a través de documentos ya existentes que nos ayuden en la construcción del sistema, algunos de los documentos que son susceptibles de análisis son: documentación sobre sistemas similares, manuales de la organización, manuales de procedimientos, etc.

El análisis documental se realiza en dos fases principales: el **análisis formal** y el **análisis de contenido**.

El **análisis formal** recolecta toda la información objetiva del documento que permite conocer aquellos datos que permiten diferenciar a un documento de otro, como es el caso de tipo de documento, autores, título de la obra, editoriales, fechas, idiomas, número de páginas, entre otros.

El **análisis documental** permite realizar un control e identificación de los diversos documentos en una colección de los mismos realizando dos operaciones básicas, *la catalogación y descripción documental* (Solis, 2003).

La *catalogación* tiene como objetivo principal el establecer un listado de todos aquellos documentos que integran a una colección, separando perfectamente los temas, autores y tipos de documentos existentes, el resultado final de la fase de catalogación es el catálogo de documentos, un instrumento que sirve de vínculo entre los usuarios de la colección y los documentos contenidos en ella.

La *descripción documental* es el proceso por el cual se describen los documentos de acuerdo con sus características internas y externas, en éste proceso se consideran atributos como el autor, el título, el lugar de edición, el editor, el año de publicación, entre otras.

La descripción documental requiere estar sujeta a normas concretas que nos permitan manejar la información de forma precisa y que facilite la clasificación y posterior búsqueda de la información dentro de la colección de documentos, las normas de catalogación más extendidas son las ISBD (International Standard Bibliographic Description) y las Normas de Catalogación Angloamericanas.

En el análisis de contenido se realizan acciones que permiten llegar a la descripción del contenido o temática de un documento generando tres resultados que permiten un mejor manejo de la información que son, la **clasificación**, la **indización** y el **resumen**.



Indizar

Es extraer una serie de conceptos que responden a los temas tratados en el documento, y que servirán como puntos de acceso para su recuperación (Rubio, 2004, p.5).

Clasificación

Es un conjunto ordenado de conceptos que se presentan distribuidos sistemáticamente en clases conformando una estructura. La creación de catálogos e índices ha sido de mucha ayuda a lo largo de la historia de la humanidad para buscar y recuperar información. Las clasificaciones más extendidas en el mundo son la CDU (Clasificación Decimal Universal), la Clasificación Decimal Dewey y la LCC (Clasificación de la Biblioteca del Congreso de Washington).

Resumen

El es una representación abreviada y precisa del contenido de un documento, sin interpretación crítica y sin mención del autor del documento (ISO) (Rubio, 2004, p. 9). En otras palabras, es una visión general sintetizada del contenido completo de un documento elaborado en un lenguaje natural. Los resúmenes ayudan a los usuarios a determinar si la información que buscan se encuentra contenida en el documento, de esta forma es posible seleccionar y descartar aquellos documentos que son útiles para el usuario.

El análisis documental implica el conocimiento del documento, el análisis, la síntesis, la representación y la recuperación.

El análisis documental es una herramienta que permite a los desarrolladores conocer el historial de la organización, sus operaciones y estructura interna, con lo anterior, es posible identificar necesidades internas y de operación que ayuden al desarrollo del sistema de información.

RESUMEN

La identificación de los requerimientos es una fase del ciclo de vida de los sistemas de información, que ayuda a identificar las necesidades de la organización y a determinar las funciones que el sistema realizará.

El proceso de identificación de requerimientos puede emplear diversos métodos o técnicas, que involucran tanto a los desarrolladores del sistema, como a los dueños y directivos de la organización, los empleados que serán los usuarios finales y a expertos que aporten su experiencia en la identificación de problemas y de necesidades.

Dentro de las diversas formas que podemos encontrar para la identificación de requerimientos, podemos encontrar métodos estructurados, como son los cuestionarios, entrevistas, diagramas de causa-efecto, etc., que permitan establecer una forma perfectamente dirigida a la obtención de cierta información en particular.

Existen otros métodos no estructurados, como el caso de la observación y el análisis documental, que permiten obtener información sobre el entorno en que se desempeñará el sistema y las expectativas del mismo.

BIBLIOGRAFÍA



Bruegge, Bernd y Dutoit, Allen H. (2002). *Ingeniería de software orientada a objetos*. México: Pearson.

Joyanes Aguilar, Luis. (2003). *Fundamentos de programación: algoritmos, estructuras de datos y objetos*. (3ª ed.) Madrid / México: McGraw-Hill.

Pfleeger, Shari L. (2002). *Ingeniería de software: teoría y práctica*. México: Prentice Hall.

Piattini Velthuis, Mario y García, Félix (coords.). (2003). *Calidad en el desarrollo y mantenimiento de software*. México: Alfaomega / Ra-Ma.

Piattini Velthuis, Mario; Calvo-Manzano Villalón, José; Cervera Bravo, Joaquín y Fernández Sanz, Luis. (2003). *Análisis y diseño de aplicaciones informáticas de gestión*. México: Alfaomega / Ra-Ma.

Pressman, Roger S. (2002). *Ingeniería de software: un enfoque práctico*. (5ª. ed.) México / Madrid: McGraw-Hill.



- Sommerville, Ian. (2001). *Ingeniería de software*. (6ª ed.) México: Pearson.
- Weitzenfield, Alfredo. (2003). *Ingeniería de software orientada a objetos con UML, Java e Internet*. México: Thomson.
- Durán Toro, A. (2000) *Metodología para la elicitación de requisitos de sistemas de software*, disponible en línea: <http://es.scribd.com/doc/49567897/15/Joint-Application-Development>, recuperado el 09/11/12
- Figuroa, Pablo. (1998). *Análisis de requerimientos*, curso disponible en línea: <http://webdocs.cs.ualberta.ca/~pfiguero/soo/metod/requerimientos.html>, consultado el 20/03/11.
- (1998a) Conceptos de un diagrama de un caso de uso, disponible en línea: http://webdocs.cs.ualberta.ca/~pfiguero/soo/uml/casos_uso01.html, consultado el 20/03/11
- Ince, Darrel. (1993). *Ingeniería de Software*. México: Addison-Wesley.
- Kendall, Kenneth. (1990). *Análisis de diseño de sistemas*. México: Prentice Hall.
- Larman, Craig. (1999). *UML y patrones: introducción al análisis y diseño orientado a objetos*. México: Prentice-Hall.
- Márquez Vite, J. M. (2002). *Sistemas de información por computadora, metodología de desarrollo*. México: Trillas.
- Meyer, Bertrand. (1999). *Construcción de Software Orientado a Objetos*. Madrid: Prentice-Hall.



Osorio Rojas, Ricardo. (2002). *El cuestionario*. Magister Educación, disponible en línea: <http://www.nodo50.org/sindpitagoras/Likert.htm>, recuperado el 08/10/12.

Rubio Liniers, María C. (2004). *El análisis documental: indización y resumen en bases de datos especializadas*, disponible en línea: http://eprints.rclis.org/bitstream/10760/6015/1/An%C3%A1lisis_documental_indizaci%C3%B3n_y_resumen.pdf, consultado el 09/11/12

Senn, James A. (s.f.). "Primera fase: Análisis", *Análisis y diseño de sistemas de información*, disponible en línea: [http://une-senn.tripod.com/new_page_1.htm#Herramientas para determinar requerimientos de sistemas](http://une-senn.tripod.com/new_page_1.htm#Herramientas_para_determinar_requerimientos_de_sistemas), recuperado el 20/03/11



UNIDAD 3

Especificación de requerimientos



OBJETIVO PARTICULAR

Registrará el detalle de los requerimientos funcionales y no funcionales.

TEMARIO DETALLADO

(28 horas)

3. Especificación de requerimientos

3.1. Estándares para especificar requerimientos

3.2. Normas para la redacción de requerimientos funcionales

3.3. Generación de la lista de requerimientos

3.4. Especificación de requerimientos funcionales

3.4.1. Especificación de casos de uso

3.5. Especificación de requerimientos no funcionales

INTRODUCCIÓN

La especificación de los requerimientos de un sistema de información es una de las partes más importantes en el ciclo de vida del mismo, de ella depende, en gran medida, la manera en que funcionará el sistema y cómo se comportará con su entorno.

La especificación de requerimientos ayuda a los desarrolladores a identificar “qué debe hacer el sistema” y “cómo debe ser”, y se hace a partir de los requerimientos funcionales y no funcionales asociados al mismo.

Una mala especificación de requerimientos conlleva un diseño deficiente y costoso de un sistema de información; para evitarlo, se han establecido formas estandarizadas que resumen las mejores prácticas en el diseño de sistemas para especificar los requerimientos del mismo. A lo largo de la presente unidad revisaremos los principales estándares y formas para especificarlos correctamente.



3.1. Estándares para especificar requerimientos

La estandarización es un proceso mediante el cual se busca alcanzar la calidad de los productos y/o servicios. Podemos decir, en palabras simples, que cuando uno estandariza un proceso, se asegura de que todo lo que se realice dentro de ese proceso, siempre entregará el mismo resultado con las mismas características, evitando con ello diferencias y por tanto variaciones en el producto o servicio.

En el caso de la ingeniería de software, se sigue una serie de normativas que permiten estandarizar los procesos que integran el ciclo de vida de los sistemas.

El IEEE (Instituto de Ingeniería Eléctrica y Electrónica por sus siglas en inglés) es una institución sin fines de lucro, dedicada a la promoción de la innovación tecnológica y el desarrollo de nuevas tecnologías para el desarrollo de la humanidad.

El IEEE es una de varias instituciones³ alrededor del mundo que se encarga de emitir normas para la estandarización de las formas en como se desarrollan nuevas tecnologías, entre sus estándares más conocidos encontramos:

- IEEE 1394 – Estándar para la entrada y salida de datos (conocido como Firewire).
- IEEE 488 - Es un estándar bus de datos digital de corto rango.
- IEEE 754 - Estándar para aritmética de punto flotante.

³ Otra es la norma de la OSI, que se enfoca en el factor de la calidad que se aplica en la parte de pruebas de sistemas y como modelos de referencia para redes; otro consorcio es el de Fuerza de Tarea de Internet, (IETF) etc., encargada de otras áreas de la informática.

En nuestro caso, hablando de los requerimientos, nos abocaremos al estándar IEEE 830, enfocado a la especificación de requerimientos de software. Este estándar busca generar un buen contenido y especificación de requerimientos de software a través de diversos esquemas. Su objetivo es especificar los requerimientos del software a desarrollar y establecer un acuerdo documentado entre el cliente y los desarrolladores de sistemas, sobre el sistema que se va a construir.

Con base en ello y siguiendo la plantilla del documento SRS (especificaciones de requerimiento de software)⁴, contiene las siguientes secciones:

1. Introducción.
2. Descripción general.
3. Requerimientos específicos.
4. Apéndices.

La **introducción** del documento SRS generado a partir del estándar IEEE 830 deberá contener los siguientes apartados:

Propósito. Describe el objetivo que se persigue al construir el sistema y su análisis.

Ámbito del sistema. Describe el entorno en el cual operará el sistema a desarrollar.

Definiciones, acrónimos y abreviaturas. Es una especie de glosario, donde se encuentra toda la terminología técnica que se empleará durante el desarrollo del sistema.

Referencias. Son todos los documentos de consulta que serán empleados a lo largo del desarrollo del documento.

Visión general del documento. Describe de manera general el contenido del documento.

⁴ Ejemplo de Especificación de requerimientos de software de Jason Robins (2003), disponible en línea: <http://readysset.tigris.org/nonav/es/templates/srs.html>.

La **descripción general** contiene los siguientes apartados:

- *Perspectiva del producto.* Se trata de una descripción de los resultados esperados al final del desarrollo del sistema, acordes con los requerimientos provistos por el cliente.
- *Funciones del producto.* Describe la forma como el sistema funcionará de manera particular (en cada módulo) y de forma general.
- *Características de los usuarios.* Define los perfiles y características generales de todos aquellos usuarios que tendrán contacto directo o indirecto con el sistema.
- *Restricciones.* Describe las limitaciones que tendrán tanto el sistema como sus usuarios.
- *Suposiciones y dependencias.* Se describen los objetos (interfaces, requerimientos, usuarios, etc.) que pueden presentar una dependencia de otros objetos durante el desarrollo del sistema.
- *Requisitos futuros.* Se trata de algunas características o funciones adicionales que el usuario podría pedir en un futuro.

Dentro de la sección de **requerimientos** encontramos los siguientes apartados:

- *Interfaces externas.* Dentro de éste, se escriben aquellos requerimientos que describen la forma como se realizará la comunicación entre el sistema y su entorno (incluyendo a los usuarios).
- *Funciones.* Esta es la parte más extensa del documento, aquí se especifican todas las funciones que deberán ser ejecutadas por el sistema, dichas funciones se expresan en forma natural con enunciados concretos de la forma “el sistema deberá.”, pudiendo emplear herramientas gráficas para ayudar en su definición,



pero siempre llegando a un enunciado. El estándar IEEE 830 permite dividir esta sección en varios segmentos:

- *Tipo de usuarios.* Aquí se organiza a los usuarios por jerarquías y dependiendo del tipo de interacción que tendrán con el sistema, posteriormente se definen requerimientos funcionales asociados a cada uno de ellos que tengan relación con sus actividades.
 - *Por objetos.* De acuerdo con el paradigma orientado a objetos, cada objeto es una representación de entidades del mundo real con atributos y funciones específicas, los objetos pueden agruparse en clases y cada una de ellas puede definir una serie de requerimientos funcionales asociados a ellos.
 - *Por objetivos.* A partir de los objetivos que se plantean al planificar la construcción del sistema, se determinan las entradas que se requieran para alcanzar dicho objetivo y la salida deseada (lo que debe hacer el sistema). Por cada objetivo se debe establecer las funciones necesarias para alcanzarlo.
 - *Por jerarquía funcional.* Se da prioridad a cada una de las funciones que realizará el sistema con el objetivo de jerarquizarlas, especificando cuáles comparten entradas y salidas de datos y detallando cada función de manera minuciosa para determinar sus requerimientos.
-
- *Requisitos de rendimiento.* Se trata de los requerimientos asociados con el desempeño del sistema, como son tiempos de respuesta, formas de almacenamiento de datos, entre otros.
 - *Restricciones de diseño.* Son todos aquellos factores que limiten el diseño del sistema, como son por ejemplo: el hardware, otros estándares o la normativa de la organización.
 - *Atributos del sistema.* Dentro de este apartado se describen los atributos que definen la calidad del sistema, como por ejemplo: su fiabilidad, su



mantenibilidad, su seguridad, etc. Es necesario que se detalle la forma en que serán implementadas cada una de ellas, por ejemplo, los formatos de nombre de usuario y contraseña, restricciones de acceso a los usuarios, mecanismos de recuperación de datos, entre otros.

- *Otros requisitos.* Se describen aquellos requerimientos funcionales que no pueden ser englobados en las secciones anteriores.

En la sección de **apéndices** se coloca toda la información que puede ser necesaria para la construcción del sistema que no se encuentra contemplada dentro del documento SRS, como son entrevistas, informes, análisis de tiempos, costos, etc.

Para que quede más clara esta explicación, en la siguiente dirección electrónica encontrarás un ejemplo de un documento SRS desarrollado por Liliana Machuca (2008), de la Universidad del Valle, en Santiago de Cali, Colombia.

Ejemplo de especificación de requerimientos

<http://es.scribd.com/doc/38497429/Ejemplo-SRS>

(Consultado el 21/11/2012)



3.2. Normas para la redacción de requerimientos funcionales

Cuando redactamos un documento de especificación de requerimientos funcionales, se debe tener cuidado de seguir las siguientes recomendaciones (derivadas de la norma IEEE 830):

1. *No ambiguo.* Se debe asegurar que todos los requerimientos escritos tengan una sola interpretación.
2. *Completo.* Incluir todo aquello que se espera que el sistema haga, entendiendo por “completitud” que deben describirse todas las posibles opciones de respuesta que una entrada al sistema pueda generar y todas las situaciones que pueden enfrentar en el ámbito del sistema.
3. *Correcto.* Todo requerimiento especificado debe ayudar a resolver una necesidad real.
4. *Comprensible.* Toda aquella persona que tenga acceso a la lectura de un SRS, debe poder interpretarlo sin dificultad.
5. *Verificable.* Se debe establecer un procedimiento que permita demostrar que cada requerimiento será satisfecho por el sistema al ser construido.
6. *Consistencia interna.* No deben existir requerimientos contradictorios.
7. *Consistencia externa.* Ninguno de los requerimientos deberá estar en contradicción con otros documentos de especificaciones superiores o del mismo nivel del SRS. (Ejemplo: Las especificaciones del hardware no deben ir en contradicción con las del software).

8. *Realizable*. El requerimiento especificado en el documento SRS debe poder implementarse con los recursos actuales.
9. *Conciso*. La redacción del documento SRS debe ser lo más breve posible sin afectar los atributos de calidad marcados.
10. *Independiente del diseño*. El documento SRS debe enfocarse en la funcionalidad general del sistema, siendo independiente de la metodología de diseño e implementación del sistema seleccionadas.
11. *Trazable*. Cada requerimiento debe poder ser referenciado de forma única y sin equivocación, para ayudar a determinar qué requerimientos son implementados en cada fase.
12. *Modificable*. Los cambios realizados deben ser fáciles de implementar.
13. *Almacenables electrónicamente*. Deseablemente, cada documento redactado debe poder almacenarse en un medio electrónico como un archivo o registro en una base de datos, es preferible si se generan con herramientas para la administración de requisitos.
14. *Anotado por importancia relativa*. Se debe dar una clasificación al requisito dándole un calificativo como “Obligatorio, Deseable u Opcional”, que ayude a la asignación de recursos y su implementación.
15. *Anotado por estabilidad relativa*. Cada requerimiento debe tener asignada una probabilidad denominada de cambio, lo que ayuda a determinar si será estable o volátil en su implementación.
16. *Versionado*. Es deseable establecer por escrito qué requerimientos serán satisfechos en una versión determinada del sistema o número de prototipo.
17. *No redundante*. Cada requerimiento debe redactarse en un solo lugar del documento SRS, evitando repeticiones.
18. *Abstracto*. El nivel de detalle de cada requerimiento debe ser el suficiente para ser comprendido, sin excesos o carencias de detalles.
19. *Preciso*. Se recomienda emplear una notación numérica para calificar las características del sistema aunque depende de la tecnología empleada para hacerlo y no siempre es posible.



20. *Reutilizable*. Determina si ciertas partes del SRS son reusables en otros requerimientos.
21. *Trazado*. Identifica claramente la fuente que dio origen al requerimiento.
22. *Organizado*. El lector del documento debe poder localizar fácilmente la información contenida en él.
23. *Con referencias cruzadas*. Se debe establecer claramente si los requerimientos tienen relación entre otros o entre otras secciones del documento.

Cabe recalcar que los documentos SRS son redactados por seres humanos, por lo que no es posible tener un documento perfecto pero sí de calidad, lo que disminuye o elimina errores de implementación.



3.3. Generación de la lista de requerimientos

Una de las formas de documentación de requerimientos son los listados. Por lo general, estas listas describen los requerimientos funcionales del sistema desde un punto de vista global.

La lista debe tener las siguientes características.

| | |
|---------------------------------|--|
| Identificador único | Número asociado a cada requerimiento identificado para el sistema. |
| Nombre del requerimiento | Nombre asociado que permita identificar de forma sencilla el requerimiento listado. |
| Necesidad | Se refiere a la categoría que va a recibir, como esencial, opcional, condicional, etc. |
| Estado | Indicando si el requerimiento ha sido definido perfectamente y es lo suficientemente claro para su uso (Cerrado) o para indicar si es necesario detallar más el requerimiento (Abierto). |
| Versión | Referente a la cantidad de cambios que ha sufrido el requerimiento a lo largo del proceso de desarrollo del sistema. |



| Identificador | Nombre | Necesidad | Estado | Versión |
|---------------|--------|------------------------------------|-----------------|---------|
| 1 | | Escencial/Opcional/ condicional | Cerrado/Abierto | 3 |

Independientemente de estas características, algunos autores agregan más a los listados, tantos como sean o consideren necesarios para registrar y manejar los requerimientos capturados.

Puedes consultar un ejemplo de listado de requerimientos en la siguiente dirección electrónica.

Pérez Pedraza, Alejandro. (2004). "Apéndice B: Lista de requerimientos", *Implementación y explotación de un datawarehouse empresarial para la toma de decisiones: aplicación a la empresa Textiles Carmelita*, tesis para licenciatura, UDLA, disponible en línea: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/perez_p_a/apendiceB.pdf, consultado el (21/11/12)

3.4. Especificación de requerimientos funcionales

Los requerimientos funcionales (como lo vimos en la unidad 1) son aquellos que describen las funciones del sistema, es decir, los que especifican qué es lo que el sistema debe de hacer de acuerdo con las características del sistema a desarrollar, el tipo de usuarios del sistema y del enfoque organizacional del sistema, como por ejemplo, el formato que debe tener un reporte de almacén, la estructura y composición de una consulta financiera, etc.

Algunos ejemplos de requerimientos funcionales son:

1. “El usuario podrá consultar la base de datos del sistema en todo momento”.
2. “El sistema deberá generar reportes en un formato compatible con un procesador de texto”.
3. “Cada operación realizada en el sistema deberá registrarse y contener un identificador único”.

Cuando algunos de los requerimientos establecidos por los usuarios tienden a ser poco claros o muy generales, deberán examinarse y detallarse; por ejemplo el requerimiento 2 donde solicita reportes compatibles con un procesador de texto, aquí se debe especificar con más claridad los formatos de los reportes y el procesador de texto que se va a usar para evitar problemas de compatibilidad.

Normalmente, cuando se especifican requerimientos funcionales, se busca que se tengan las siguientes características:

- *Completos*. Significa que todas las funciones solicitadas por el usuario deben estar perfectamente definidas.
- *Consistentes*. Las definiciones de los requerimientos no deben contener definiciones que sean contradictorias para evitar confusiones.
- *Claros*. Deben de ser redactados de tal forma que los usuarios puedan comprender a la perfección lo que se solicita en el requerimiento.
- *Funcionales*. Establecen las características externas del sistema y su comportamiento, pero no se deben establecer características de diseño.
- *Priorizados*. Los requerimientos deben jerarquizarse para que los desarrolladores puedan establecer cuáles serán obligatorios, deseables u opcionales.

Un ejemplo de un requerimiento mal especificado es el siguiente:

“Para facilitar el uso del editor gráfico, se podrá activar y desactivar una rejilla que permitirá alinear las figuras del diagrama. Cuando se ajuste la figura al tamaño de la pantalla, se reducirá el número de líneas de la rejilla para que no se dificulte la visualización del diagrama.” (Berzal, “[Especificación de requerimientos](#)”, p. 14)

Al leer la especificación, podremos notar que dentro del enunciado se encuentran involucrados varios requerimientos, por lo que la especificación queda muy abierta o vaga. La forma correcta de especificación sería:

“El editor permitirá el uso de una rejilla de líneas horizontales y verticales que aparecerán dibujadas tras el diagrama.”



Justificación:

La rejilla facilita la creación de diagramas cuidados en los que las figuras se puedan alinear con facilidad.” (Berzal, “[Especificación de requerimientos](#)”, p. 15)

Al leer la especificación, podemos notar que cumple con las características de ser *completo y consistente*, así como de contar con una justificación que refuerza la función solicitada.

Una herramienta muy útil para especificar requerimientos funcionales, es mediante el empleo de *casos de uso*, los cuales revisaremos a continuación.

3.4.1. Especificación de casos de uso

Los diagramas de caso de uso son documentos que describen la forma como el sistema deberá comportarse desde el punto de vista de los usuarios. Estos diagramas son la fuente principal de información que ayuda a los diseñadores de sistemas a determinar los requerimientos funcionales del sistema, en otras palabras, los diagramas de caso de uso representan las funciones que el sistema puede ejecutar.

Una de las principales ventajas de usar diagramas de casos de uso es la facilidad para ser interpretados, haciendo más sencilla la comunicación entre desarrolladores y los clientes.

Los elementos principales que integran un diagrama de caso de uso son los siguientes:

Actores

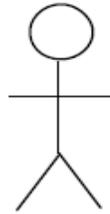
Los actores son la representación gráfica de los diversos tipos de usuarios del sistema, concibiéndose que un usuario es toda aquella entidad externa que tiene interacción con el sistema, sean estas personas, departamentos de una empresa, otros sistemas de información, etc.

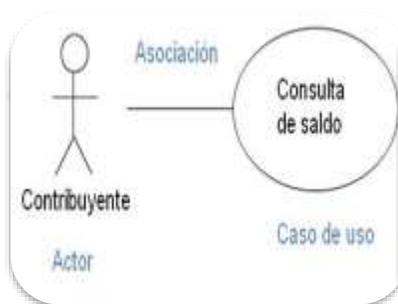
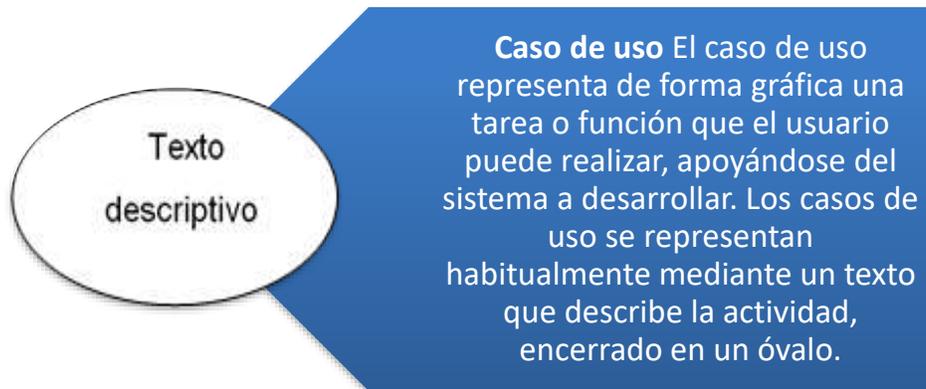
Para desarrollar de forma efectiva un diagrama de caso de uso, es recomendable independizar a los actores de acuerdo con la forma en que actúan con el sistema, por ejemplo, en un sistema de consulta, el usuario que solamente consulta la información del sistema no realiza las mismas tareas que el usuario encargado de alimentar dicha información dentro del sistema.



Los actores, en un diagrama de casos de usos, representan papeles que una entidad puede jugar al interactuar con el sistema, estos papeles o roles —como también se les denomina— pueden representar a más de un usuario. Volviendo al ejemplo del sistema de consulta, la forma de interactuar de dicho usuario con el sistema puede representar a más de una persona que realiza la misma acción, por ejemplo, en el caso de las páginas de Internet del Gobierno Federal, como la de Hacienda, existen diferentes usuarios que utilizan su portal para enviar sus declaraciones anuales, en un diagrama de caso de uso no vamos a representar a cada individuo, más bien representamos a ese conjunto de usuarios como una sola entidad que realiza una tarea genérica.

El símbolo para representar a los actores es el siguiente:





Asociaciones Las asociaciones son líneas rectas que relacionan a cada caso de uso con los actores que realizan dicha tarea. Entonces, para cada usuario existirá al menos una asociación con una acción o caso de uso, todo dependiendo de la forma en que se interactúe con el sistema.



Escenarios

Los escenarios son la representación de una interacción entre un usuario y el sistema. Dentro de la fase de diseño del sistema pueden surgir múltiples escenarios que describan la forma en que diferentes usuarios interactúen con el sistema.

Ejemplos

Escenario 1

Cliente consultando su saldo en un cajero automático.

Escenario 2

Capturista ingresando datos personales para su registro en una base de datos.

Durante el proceso de documentación del sistema, todos los escenarios que sean generados deben registrarse en un diagrama denominado de *secuencia*, que tiene por objetivo numerar e indicar la forma de interacción de los usuarios en todas las formas posibles con el sistema.

Al especificar un caso de uso, se busca realizar una descripción de cada una de las partes definidas en él, para lograr establecer una descripción a detalle del mismo.

Cuando se especifican casos de uso se debe considerar lo siguiente:

Interacciones

Describir la forma de participación de los actores, primarios y secundarios, a lo largo del desarrollo de todo el caso de uso.

Eventos

Describe cada uno de los eventos que tiene ocurrencia en el caso de uso, por ejemplo, la captura de datos, la consulta de datos, la compra y pago de algo, etc.

Nivel de detalle

Se debe describir lo mejor posible las funciones que realizará el sistema a lo largo del desarrollo del caso de uso, lo que ayudará a extraer los requerimientos funcionales con mayor precisión.

Escenarios

Se debe describir cada uno de los escenarios que contemple el caso de uso, lo anterior se debe realizar estableciendo un flujo de acciones principales y los flujos secundarios y extraordinarios que puedan aparecer.

Claros y enfocados en el usuario

Se deben redactar empleando un lenguaje simple, evitando usar tecnicismos para que puedan ser interpretados por los clientes y los usuarios. (Véase, Orozco, [Especificaciones de casos de uso](#)).

Al redactar las especificaciones de los casos de uso, podemos seguir la siguiente plantilla:



| | |
|--|---|
| Caso de uso | Nombre e identificador único para el caso de uso. |
| Fuentes | Nombre de los clientes o los usuarios que proporcionan la información para la elaboración del caso de uso. |
| Actores | Nombre de los actores e identificadores asociados a cada uno para su seguimiento. Ejemplo: Cliente C1, Vendedor de mostrador VM1, etc. |
| Descripción | Descripción del caso de uso. |
| Flujo principal | Descripción de las acciones realizadas en el caso de uso, se recomienda redactarlo en forma de lista: Paso 1: descripción. Paso 2: descripción. |
| Flujos alternos y/o extraordinarios | Descripción de las acciones secundarias derivadas del flujo principal. |
| Condiciones previas | Descripción de la situación antes de comenzar el caso de uso. |
| Condiciones posteriores | Descripción de la situación posterior al caso de uso. |
| Requerimientos trazados | Listado y descripción de los requerimientos funcionales identificados en el caso de uso. |
| Puntos de inclusión | Son elementos del caso de uso que son empleados en otros casos de uso. |
| Puntos de extensión | Descripción de los enlaces que permiten extender la funcionalidad de un caso de uso a uno o varios puntos dentro del flujo principal. |
| Notas | Información adicional que ayude a la comprensión y análisis del caso de uso. |

Cuadro 3.1. Cómo especificar casos de usos (Pérez Rodríguez, [Estructuración y especificación de casos de uso](#))

En la siguiente dirección electrónica encontrarás un ejemplo del empleo de casos de uso y su especificación en el desarrollo de un sistema de gestión de artículos deportivos, desarrollado por el Departamento de Sistemas Informáticos y Computación de la Universidad Politécnica de Valencia en España.

Ejemplo de desarrollo software utilizando la metodología RUP

http://users.dsic.upv.es/asignaturas/facultad/lsi/ejemplorup/Casos_Uso.html,

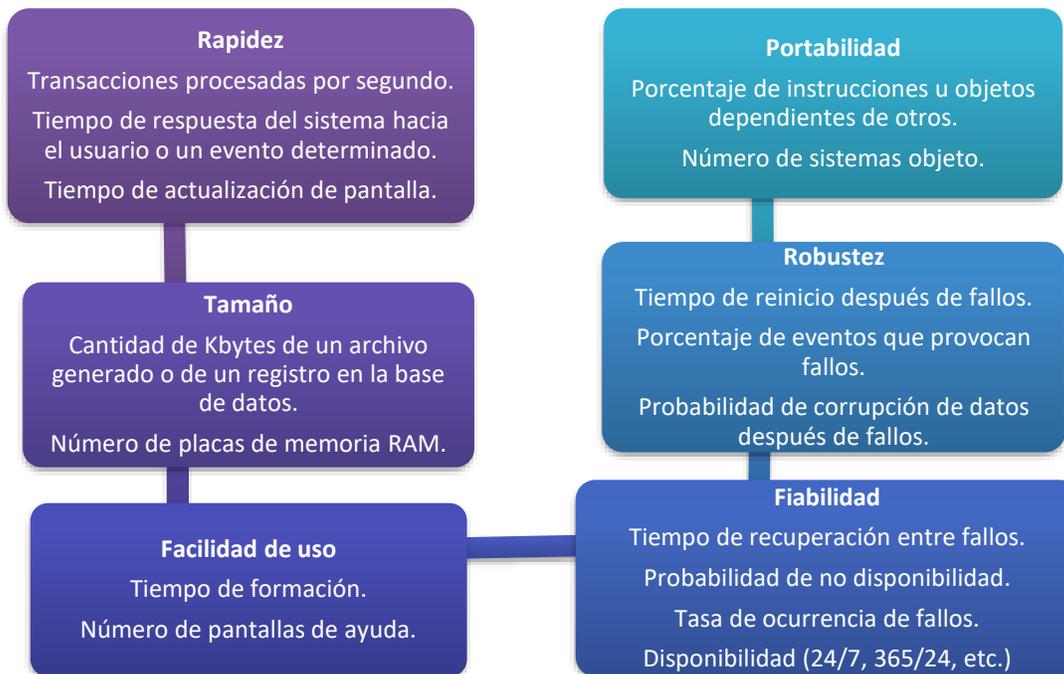
consultado el 19/10/2012.

3.5. Especificación de requerimientos no funcionales

Los requerimientos no funcionales definen las propiedades que debe tener el sistema, como son fiabilidad, tiempo de respuesta y capacidad de almacenamiento, entre otros. También ayudan a definir las restricciones que tendrá el sistema como son: capacidad de los dispositivos de entrada/salida (capacidad de almacenamiento, cantidad de consultas por usuario, etc.) y la forma de representación de los datos en las interfaces del sistema para su interpretación por los usuarios (formatos de reportes, formato de pantalla, etc.)

Los requisitos no funcionales se redactan en forma cuantitativa, siempre que sea posible, con la finalidad de que puedan evaluarse de forma objetiva.

A continuación se presenta una tabla con algunas métricas sugeridas por Ian Sommerville, que pueden ser empleadas para especificar un requerimiento no funcional (Sommerville, 2001, p. 114).



Un ejemplo de la especificación de un requerimiento no funcional es el siguiente:

“Cuando haya hasta 100 usuarios accediendo simultáneamente al sistema, su tiempo de respuesta no será en ningún momento superior a 2 segundos” (Berzal, [Especificación de requerimientos](#), p. 17).

Podemos observar, al leer la especificación, que cuenta con una métrica que permitirá evaluar el requerimiento una vez que sea implementado.

RESUMEN

La especificación de requerimientos se refiere a la forma correcta de expresar las necesidades de un sistema de información; estos requerimientos pueden ser funcionales o no funcionales, y dependiendo de su naturaleza se especificarán.

Dentro de los diversos enfoques existentes para el desarrollo de software, ya sea empleando el paradigma orientado a objetos; en cascada; el estructural, etc., podemos encontrar diversas metodologías para identificar requerimientos, pero de manera general, para poder especificar un requerimiento debemos seguir un proceso estandarizado establecido por la norma IEEE 830, donde figura un formato estándar para la especificación de requerimientos de software conocido como SRS; en dicho documento se conjugan las “buenas prácticas” para la especificación de los requerimientos.

Los requerimientos en sí, pese a que se cuenta con plantillas y normas que nos ayudan a documentarlos y resguardarlos, no siempre son redactados de forma adecuada. Esta mala redacción conlleva problemas de claridad y redundancia en la información, lo que tiene como consecuencia una mala implementación del requerimiento en el sistema.

Una buena redacción de requerimientos comienza con el empleo de un lenguaje natural libre de terminología técnica, tomando en cuenta que el documento de especificación de requerimientos servirá a la vez como un contrato entre clientes y desarrolladores para establecer las características y funciones del sistema.

Adicionalmente, hay que buscar que los requerimientos sean concretos, simples, completos, que sean trazables y que cumplan con una serie de características que nos ayuden a implementar, evaluar y dar seguimiento a los requerimientos establecidos.

Cada tipo de requerimiento, dependiendo de si es funcional o no, deberá seguir una serie de formatos sugeridos para su especificación y también será posible emplear herramientas como los casos de uso, para obtener información más detallada en la especificación de requerimientos.

Bibliografía



sugerida

Bruegge, Bernd y Dutoit, Allen H. (2002). *Ingeniería de software orientada a objetos*. México: Pearson.

Joyanes, L. (2003). *Fundamentos de programación: algoritmos, estructuras de datos y objetos*. (3ª ed.) Madrid / México: McGraw-Hill.

Pfleeger, Shari L. (2002). *Ingeniería de software: teoría y práctica*. México: Prentice Hall.

Piattini Velthuis, Mario y García, Félix (coords.). (2003). *Calidad en el desarrollo y mantenimiento de software*. México: Alfaomega / Ra-Ma.

Piattini Velthuis, Mario; Calvo-Manzano Villalón, José; Cervera Bravo, Joaquín y Fernández Sanz, Luis. (2003). *Análisis y diseño de aplicaciones informáticas de gestión*. México: Alfaomega / Ra-Ma.

Pressman, Roger S. (2002). *Ingeniería de software: un enfoque práctico*. (5ª. ed.) México / Madrid: McGraw-Hill.



Sommerville, Ian. (2001). *Ingeniería de software*. (6ª ed.) México: Pearson.

Weitzenfield, Alfredo. (2003). *Ingeniería de software orientada a objetos con UML, Java e Internet*. México: Thomson.



UNIDAD 4

Validación de requerimientos



OBJETIVO PARTICULAR

Seleccionará los requerimientos que están alineados con las necesidades del negocio.

TEMARIO DETALLADO

(28 horas)

4. Validación de requerimientos

4.1. Identificación de dependencias entre requerimientos

4.2. Validación de requerimientos contra objetivos de la organización, del proyecto y del sistema

4.3. Control de cambios a los requerimientos

INTRODUCCIÓN

La validación de requerimientos es un proceso muy similar al de análisis, con la diferencia principal de que el objetivo de la validación es revisar y seleccionar aquellos requerimientos completos que son útiles para el desarrollo del sistema.

El proceso de validación es muy importante debido a que, si existe algún error en los requerimientos, estos pueden ocasionar costos excesivos que impacten en la construcción del sistema.

Aunque puede resultar difícil demostrar que un conjunto de requerimientos es de utilidad para el usuario, sin éstos el sistema no podrá ponerse en marcha. Para poder hacer dicha demostración, será necesario validar los requerimientos, para ello abordaremos los siguientes aspectos:

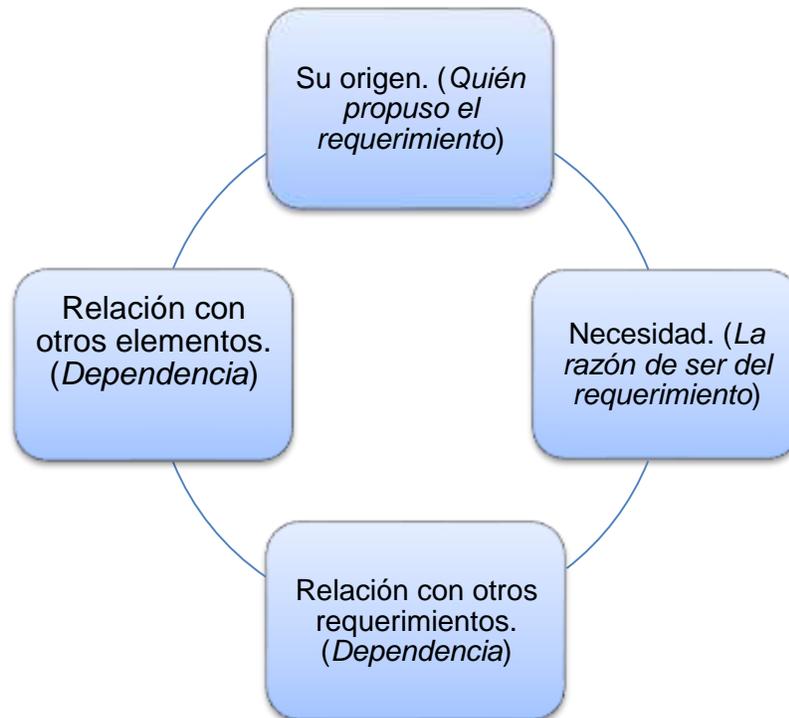
1. *Identificación de dependencias entre requerimientos*, donde buscaremos aquellos requerimientos que se encuentren enlazados con otros y que pudieran causar problemas.
2. *Validación de requerimientos contra objetivos de la organización*, en donde se verificará que los requerimientos establecidos sean acordes con lo que buscan los clientes y sean los adecuados para el sistema que se va a desarrollar.
3. *Control de cambios a los requerimientos*, que se encargará de garantizar la integridad de los requerimientos si es que alguno de ellos requiere de una modificación.

A lo largo de esta unidad profundizaremos en estos temas tan importantes para la validación de los requerimientos.

4.1. Identificación de dependencias entre requerimientos

Una de las características fundamentales se considerar en la especificación de requerimientos es su trazabilidad o rastreabilidad, que quiere decir que se debe de poder identificar todas las partes del sistema relacionadas con un requisito. De manera general, todos los requerimientos deben ser rastreables o trazables para poder mantener consistencia con el resto de los requerimientos y de los elementos del sistema.

La trazabilidad proporciona información como:



El empleo de matrices de trazabilidad ayuda a establecer los puntos anteriores de manera eficiente y controlada.

Un tipo de matriz empleado para la trazabilidad, es la matriz “*hacia adelante/hacia atrás*”

La matriz “*hacia adelante*” ayuda a enlazar los requisitos con las etapas de diseño e implementación. La matriz “*hacia atrás*” enlaza a los requerimientos con sus fuentes.

Un formato de este tipo de matriz es el siguiente:

| Matriz de trazabilidad de requerimientos | | | | | | | | | |
|--|---------|---------|----------------------|------------------|--------|-----------------------------|-----------------------------------|--------------------------------|--------------------|
| Requerimiento | | | Diseño de alto nivel | Diseño detallado | Código | Id. Caso de prueba unitario | Id. Caso de prueba de integración | Id. caso de prueba del sistema | Petición de cambio |
| Negocio | Usuario | Sistema | | | | | | | |
| | | | | | | | | | |
| | | | | | | | | | |

Figura 4.1. Formato de matriz de trazabilidad hacia atrás/hacia adelante (INTECO, [2008](#), p. 16)

Dentro de la matriz se registran todos los requerimientos de la organización. Por cada requerimiento de la organización se identifican los requerimientos de los usuarios y de cada requerimiento de usuario se identifican los del sistema asociados, y así sucesivamente; por ejemplo, la organización solicita que el sistema entregue una consulta con las ventas del último mes ordenado por fecha y tipo de operación, este requerimiento se puede asociar con los usuarios del departamento de ventas que solicitan a los desarrolladores implementar un catálogo de tipo de operación que se despliegue en forma de lista para hacer más eficiente la captura de los registros de ventas.

Para identificar la relación existente entre cada requerimiento, se emplea otra matriz denominada **de dependencia**, la cual se ilustra a continuación.

| | Requerimientos (A) | | | | |
|--------------------|--------------------|---|---|---|---|
| Requerimientos (B) | 1 | 2 | 3 | 4 | 5 |
| 1 | | * | | | |
| 2 | | | * | | |
| 3 | * | | | * | |
| 4 | | | | | * |
| 5 | | | * | | |
| | | | | | |
| | | | | | |

Figura 4.2. Formato de matriz de dependencia (ITECO, 2008, p. 17)

Los requerimientos (A) representan aquellos requerimientos que dieron origen a las dependencias. Los requerimientos (B) son los que dependen de otros requerimientos.

Por ejemplo, de la matriz anterior, podemos leer que los requerimientos (B) 2 y 5, dependen del requerimiento (A) 3, el 1 (B) del 2 (A), el 3 (B) del 1 (A) y del 4(A) y el 4 (B) del 5 (A).

La matriz ayuda a los desarrolladores a entender de mejor manera cómo se relacionan los requerimientos, lo que permite un mejor análisis al momento de manejar los cambios en los mismos.

Otro formato alternativo más simple de la matriz de dependencia puede ser:



| Requisito | Depende de: |
|-----------|-------------|
| R1 | R3, R4 |
| R2 | R5, R2 |
| R3 | R4, R5 |
| R4 | R2 |
| R5 | R1 |

Figura 4.3. Formato alternativo de matriz de dependencia

Es importante que una vez establecidas las matrices de trazabilidad, se establezca un documento complementario al de especificación de requerimientos (SRS), denominado *Manual de trazabilidad*.

El manual de trazabilidad incluye las políticas de trazabilidad establecidas en el desarrollo del sistema y toda la información sobre la trazabilidad de los requerimientos, estas políticas se establecen de acuerdo con los objetivos de la organización para la cual se desarrolla el sistema y a la forma de trabajo de los desarrolladores.

Algunos de los factores que influyen en el establecimiento de las políticas de trazabilidad son:

- **Número de requerimientos.** Entre más requerimientos sean especificados, será mayor la necesidad de establecer políticas formales de trazabilidad.
- **Tiempo de vida estimado del producto.** Entre mayor sea el tiempo de vida proyectado del sistema o software, será necesario emplear políticas de trazabilidad más comprensibles.



- **Nivel de madurez de la organización.** Entre más experiencia tenga la organización, será más madura, por lo que la implementación de las políticas de trazabilidad será más simple.
- **Tamaño y estructura del equipo de desarrollo.** Dependiendo de la diversidad y número de integrantes del equipo de desarrollo, será necesario implementar políticas de trazabilidad acorde a sus perfiles.
- **Tipo de producto.** Dependiendo del tipo de sistema a desarrollar, se deberán empelar políticas de trazabilidad acordes con ellos, es decir, entre mayor importancia e impacto en las operaciones de la organización, las políticas deberán ser más específicas, indicando los parámetros para realizar la trazabilidad del requerimiento.

Algunas de las ventajas que se tienen al manejar un manual de trazabilidad son:



El manual de trazabilidad así como las matrices deben actualizarse en cada momento, por lo que es recomendable asignar a un responsable que se encargue de dicha tarea.

4.2. Validación de requerimientos contra objetivos de la organización, del proyecto y del sistema

A partir de los requerimientos adquiridos, es necesario realizar una comprobación de su validez, haciendo una comparación con las descripciones iniciales y verificando si el modelo seleccionado corresponde a lo que se ha solicitado.

La **fase de validación de requerimientos** suele realizarse haciendo una simulación que nos permita comprobar que el modelo de sistema seleccionado responde a la forma en que lo ha solicitado el cliente. Otra manera de validar los requerimientos, es verificar con el cliente si el modelo seleccionado es de su conveniencia. En algunos casos, dependiendo del modelo de solución empleado, será necesario construir prototipos con una funcionalidad similar que se aproxime al resultado final.

La *revisión de requisitos* es uno de los métodos de validación más eficiente; por medio de la técnica de revisión es posible:

- Descubrir inconsistencias y defectos en los requisitos.
- Disminuir los costos en la fase de desarrollo del sistema alrededor de un 30%
- Reducir los tiempos de la fase de pruebas hasta en un 50%.

El *proceso de revisión de requisitos* consiste, principalmente, en realizar reuniones planificadas entre los clientes y desarrolladores; dichas reuniones tienen el objetivo de corroborar que los requerimientos seleccionados posean los atributos necesarios para garantizar la calidad del sistema.

Como parte de los integrantes de las reuniones, adicionalmente al cliente y el desarrollador, es deseable que se incluyan a personas con experiencia en el manejo de requerimientos que sean externos al proyecto; lo anterior con la finalidad de que la selección de los mismos sea de forma imparcial.

Las *reuniones de revisión* siguen seis etapas principales:

- 1. Establecimiento del plan de revisión.** Este paso se enfoca a establecer las tareas a realizar en la reunión, los participantes de la misma y la planificación temporal.
- 2. Distribución de documentos para revisión.** El documento generado en las especificaciones de requerimientos es el centro de la revisión, pero es necesario complementar a los participantes con aquellos documentos que faciliten su comprensión y análisis.
- 3. Preparación de la reunión.** Aquí se define la logística de la reunión, los tiempos y formas de realizarla, adicionalmente cada participante debe revisar de forma individual la documentación recibida para exponer las diferencias que encuentren en los diversos requerimientos.
- 4. Realización de la reunión.** Esta parte es la ejecución propiamente, hablando de lo planificado en el paso 3.

5. Identificación de defectos. Una vez realizada la reunión, antes de finalizarla, es deseable listar y especificar aquellos defectos en los requerimientos identificados en la revisión, la lista puede ser desarrollada en forma de tabla como se muestra a continuación:

| No. de requisito | Defectos detectados | Acciones recomendadas |
|------------------|---|---|
| 1 | Error de estilo, que lleva a ambigüedad | Modificar el texto del requerimiento de tal forma que diga algo como “El sistema deberá permitir el registro de los fondos bibliográficos”. |
| 2 | Idem | Idem |
| 11 | Ambigüedad | Precisar la duración de las reservas |
| 12 | Concisión | No se han identificado diferencias entre profesores y alumnos a todo lo largo de la lista de requisitos. Este requisito se deberá eliminar, ya que no proporciona ningún tipo de información relevante. |
| 15 | Realizabilidad | El sistema no puede realizar automáticamente los préstamos. Quizás se refiere a que debe proporcionar el máximo de automatización. Precisar, en este caso, o eliminar por irreal. |
| 17 | Concisión | Separar lo referido a libros prestados de lo referido a libros reservados. |

Formato recomendado para listar defectos de requerimientos (Validación de requisitos, UPM, 2011).

6. Correcciones de documentos. Durante esta fase, el desarrollador tiene que evaluar y realizar los cambios que sean convenientes, resultado del análisis obtenido en la reunión y notificar de los cambios a los participantes para su aprobación.

Durante la preparación y realización de la reunión se aconseja utilizar un listado de revisión o “requirements review checklist”, este documento es un estándar establecido en la norma IEEE-730-2002, para el aseguramiento de la calidad que se debe de generar a partir de la obtención de los requerimientos para revisarlos y aprobarlos en la fase de validación.



La lista de revisión de requerimientos debe contener una serie de preguntas que serán de utilidad para su validación y deben agruparse por los siguientes criterios (Criterio de validación de requerimientos, 2010):



Tbh creative: [Business Blogging Checklist](#)

1. Necesario (determina si el requerimiento a validar es indispensable para el desarrollo del sistema).

→ ¿Es necesario el requerimiento para el objetivo de la aplicación?

→ ¿Es clara la importancia del requerimiento dentro de la organización?

→ ¿La omisión del requerimiento provocaría una deficiencia en el sistema?

2. No ambiguo (determina si el requerimiento está claro tanto para los clientes como desarrolladores y evita repeticiones).

→ ¿El requerimiento se interpreta igual al ser leído por más de una persona?

→ ¿Está la funcionalidad claramente definida?

→ ¿La redacción del requerimiento es simple y clara?

3. Verificable (ayuda a determinar la fuente de origen del requerimiento).

→ ¿Existen soportes que ayuden a comprender el requerimiento?

→ ¿Se encuentra definido un método de verificación para cada funcionalidad?

→ ¿Se puede hacer una trazabilidad del documento hasta su origen?



4. Completo (ayuda a determinar si se incluye todo aquello que se espera que el sistema deba de hacer).

→ ¿Se encuentran identificadas las funcionalidades que involucran al requerimiento?
→ ¿Están definidas las interfaces requeridas?

→ ¿Están definidas las entradas y salidas?
→ ¿Están definidos los límites operativos de las funcionalidades?

→ ¿Están definidas todas las funcionalidades que se necesitan?
→ ¿Están definidos los requerimientos de prueba?

5. Correcto (ayuda a determinar si el requerimiento ayuda a resolver una necesidad real de la organización).

→ ¿Están los acrónimos bien definidos?

→ ¿El documento se encuentra libre de errores ortográficos y gramaticales?

→ ¿En el requerimiento se identifican casos de prueba o vivencias reales?

6. Priorizable (ayuda a determinar la importancia del requerimiento en forma jerárquica).

→ ¿Se encuentra incluida la prioridad de implementación del requerimiento?

→ ¿Se estableció un orden jerárquico en la definición de los requerimientos?

→ ¿Los requerimientos que están relacionados unos con otros se encuentran identificados con un nivel de impacto o importancia?



7. Viable (determina si el requerimiento se puede implementar).

→ ¿Los requerimientos están dentro del ámbito del sistema?

→ ¿Hay identificación de riesgos en los requerimientos?

→ ¿Hay consistencia con las políticas de la organización del cliente?

8. Consistente (ayuda a definir si el requerimiento no tiene contradicciones).

→ ¿El documento se encuentra bien organizado?

→ ¿El esquema general es consistente con los requerimientos de alto nivel?

→ ¿Se encuentra el requerimiento duplicado o en conflicto con otro requerimiento?
→ ¿Las referencias o relaciones con otros requerimientos se encuentran bien definidas?

Un ejemplo del listado de revisión de requerimientos se puede descargar de la siguiente dirección electrónica: <http://es.scribd.com/doc/55686224/Lista-de-Chequeo-Requerimientos>

4.3. Control de cambios a los requerimientos

El proceso de **control o gestión de cambios a los requerimientos** se aplica a cada uno de los cambios que sean propuestos en los requerimientos. Es un proceso formal que ayuda a tratar los cambios de forma consistente y controlada. Existen tres etapas en el proceso:

1. **Análisis del problema y especificación del cambio.** Comienza con la identificación de algún problema en algún requerimiento o con la propuesta por parte del cliente o algún desarrollador para el cambio en un requerimiento. La propuesta o el problema se analizan para verificar su validez, posteriormente, se entrega el resultado del análisis a quien identificó el problema o solicitó el cambio.
2. **Análisis del cambio y análisis de costos.** Los efectos que puede tener el cambio propuesto se evalúan empleando la información de rastreo y el conocimiento general de los requerimientos del sistema. El *costo del cambio* se determina en términos de la modificación del documento de requerimientos (SRS) y con base en el impacto en el diseño y la implementación del sistema. Realizado el análisis, se tiene que tomar una decisión sobre si se debe continuar o no con el cambio de requerimiento.

3. **Implementación del cambio.** Una vez que se decide continuar con el cambio en el requerimiento, éste debe efectuarse en el documento de requerimientos, y de ser necesario, en el diseño e implementación del sistema. El documento de requerimientos debe estar organizado de tal forma que sea posible hacer cambios en su contenido sin tener que redactar grandes partes de él. Los cambios en el documento se realizan minimizando las referencias externas y haciendo el documento tan modular como sea posible para permitir el cambio de alguna sección sin afectar otras partes del documento.

Hay que tener cuidado en el control de cambios en los requerimientos, ya que se puede desfasar la especificación de los requerimientos y la implementación del sistema.

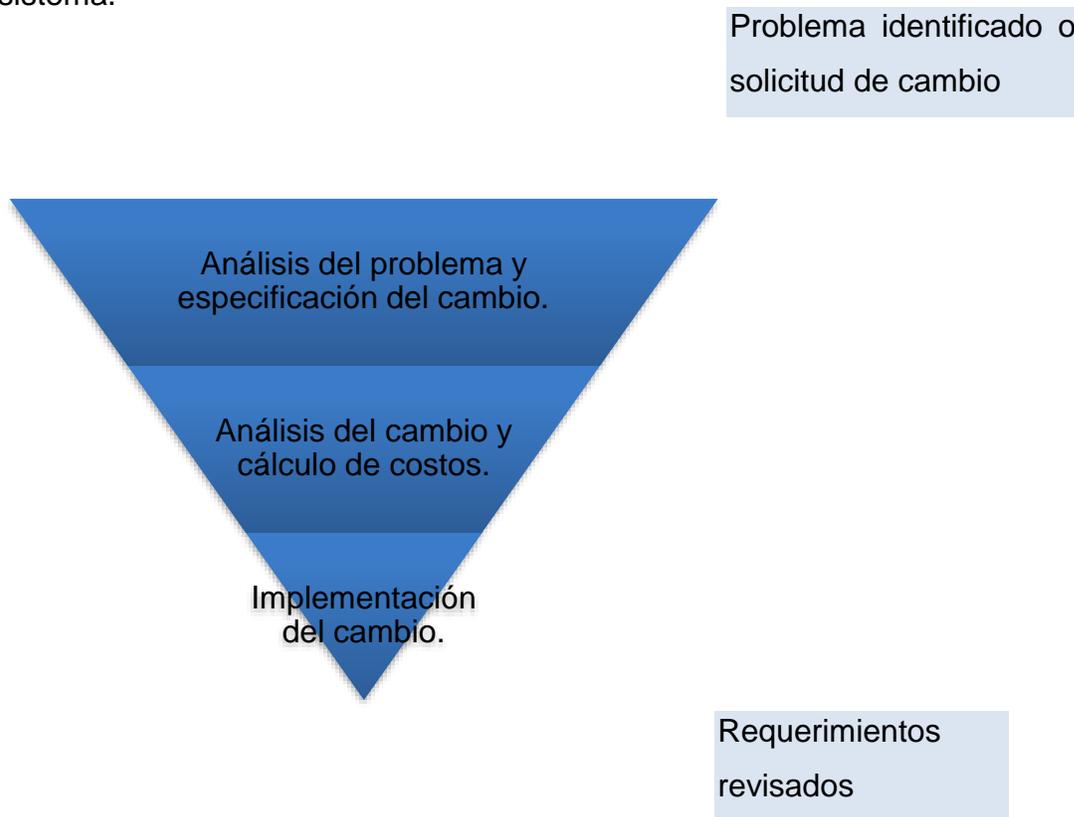


Figura 4.4. Proceso de control de cambios en los requerimientos

RESUMEN

La validación de requerimientos es una fase crucial en el desarrollo de los sistemas de información. Desde el inicio del análisis del sistema se recaban requerimientos que, a la postre, se prueban y validan para poder determinar si son o no de utilidad.

El objetivo de la validación de requerimientos es la detección de aquellos requerimientos que son incorrectos, que presentan inconsistencias o errores, y que son de difícil implementación, lo anterior será registrado en un documento en forma de lista que permita a los desarrolladores identificar y corregir dichos requerimientos.

La primera fase de la validación comienza con el establecimiento de la dependencia de los requerimientos, ya sea entre ellos o con otros elementos del sistema; lo anterior se logra mediante el empleo de matrices de trazabilidad que ayudan a establecer las relaciones existentes entre los mismos requerimientos y elementos en la fase de diseño e implementación.

Una vez establecidas las relaciones, es necesario realizar una revisión conjunta con el cliente y, de ser posible, con expertos en desarrollo de sistemas ajenos al proyecto, con la finalidad de revisar y verificar la validez de los requerimientos para que estén acordes tanto con el proyecto, como con las necesidades del cliente.

De la revisión de requerimientos, en muchas ocasiones, se derivan cambios en los mismos, el establecimiento de un control de cambios formal permitirá minimizar efectos tanto en la parte de diseño e implementación del proyecto, así como en los costos del mismo.

BIBLIOGRAFÍA



SUGERIDA

Bruegge, Bernd y Dutoit, Allen H. (2002). *Ingeniería de software orientada a objetos*. México: Pearson.

Joyanes, L. (2003). *Fundamentos de programación: algoritmos, estructuras de datos y objetos*. (3ª ed.) Madrid / México: McGraw-Hill.

Pfleeger, Shari L. (2002). *Ingeniería de software: teoría y práctica*. México: Prentice Hall.

Piattini Velthuis, Mario y García, Félix (coords.). (2003). *Calidad en el desarrollo y mantenimiento de software*. México: Alfaomega / Ra-Ma.

Piattini Velthuis, Mario; Calvo-Manzano Villalón, José; Cervera Bravo, Joaquín y Fernández Sanz, Luis. (2003). *Análisis y diseño de aplicaciones informáticas de gestión*. México: Alfaomega / Ra-Ma.

Pressman, Roger S. (2002). *Ingeniería de software: un enfoque práctico*. (5ª. ed.) México / Madrid: McGraw-Hill.



Sommerville, Ian. (2001). *Ingeniería de software*. (6ª ed.) México: Pearson.

Weitzenfield, Alfredo. (2003). *Ingeniería de software orientada a objetos con UML, Java e Internet*. México: Thomson.

Plan 2012
2016
actualizado

