

Contenido

INGENIERIA DE REQUERIMIENTOS	2
Requerimientos funcionales	4
Requerimientos no funcionales	5
Ingeniería del conocimiento	12
Especificaciones estructuradas	13
Adquisición y análisis de requerimientos	15
Descubrimiento de requerimientos	16
Descubrir: Técnica Entrevistas	18
Descubrir: Técnica Escenarios	18
Descubrir: Técnica Casos de uso	20
Descubrir: Técnica Etnografía	20
Descubrir: Técnica Revisar el mercado de software	21
Validación de requerimientos	22
Gestión de requerimientos –	24

INGENIERIA DE REQUERIMIENTOS

*Es hacer explícito todo aquello que se hace y cómo se hace para luego
Imaginar creativamente que es lo que se necesita.*

Los requerimientos para un sistema son descripciones de lo que el sistema debe hacer:
El servicio que ofrece y las restricciones en su operación.

Los requerimientos reflejan las necesidades de los clientes por un sistema que atienda cierto propósito, como sería controlar un dispositivo, colocar un pedido o buscar información.

Al proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se le llama ingeniería de requerimientos (IR).

La descripción de un “requerimiento” puede variar según el alcance del proyecto

Un requerimiento es simplemente un enunciado abstracto de alto nivel en un servicio que debe proporcionar un sistema, o bien, una restricción sobre un sistema.

O

También puede ser una definición detallada **y formal (documentado y aceptado)** de una función del sistema.

En este texto se distinguen con el uso del término “**requerimientos del usuario**” para representar los requerimientos abstractos de alto nivel; y “**requerimientos del sistema**” para caracterizar la descripción detallada de lo que el sistema debe hacer.

Definición del requerimiento del usuario

1. El MHC-PMS elaborará mensualmente informes administrativos que revelen el costo de los medicamentos prescritos por cada clínica durante ese mes.

Especificación de los requerimientos del sistema

- 1.1 En el último día laboral de cada mes se redactará un resumen de los medicamentos prescritos, su costo y las clínicas que los prescriben.
- 1.2 El sistema elaborará automáticamente el informe que se imprimirá después de las 17:30 del último día laboral del mes.
- 1.3 Se realizará un reporte para cada clínica junto con los nombres de cada medicamento, el número de prescripciones, las dosis prescritas y el costo total de los medicamentos prescritos.
- 1.4 Si los medicamentos están disponibles en diferentes unidades de dosis (por ejemplo, 10 mg, 20 mg) se harán informes por separado para cada unidad de dosis.
- 1.5 El acceso a los informes de costos se restringirá a usuarios autorizados en la lista de control de acceso administrativo.

Por ejemplo:
LA FUENTE DE UN
REQUISITO PUEDE
SER EL MANUAL
DE
PROCEDIMIENTOS
DE LA EMPRESA
(DE EXISTIR)

**El resultado de este
proceso es un
documento de
requerimientos que
puede formar parte
del contrato de
desarrollo del
sistema.**

Este contrato delimita el **alcance del sistema**- lo que va a hacer el sistema y no otra cosa
Observar que los requerimientos – son necesidades explicadas de manera coloquial con palabras- no son soluciones de lo que va a hacer el programa (que sería el diseño de las funcionalidades con las cuales un usuario va a ser capaz de atender sus necesidades)-
NO ES LA SOLUCION- no son las **funcionalidades**, hay que concentrarse en las necesidades y como se hacen o se podrían hacer las cosas.

Para hacer el llenado de requisitos- se pueden hacer platicas o reuniones -

TIPOS DE REQUERIMIENTOS

A menudo, los requerimientos del sistema de software se clasifican como requerimiento funcionales o requerimientos no funcionales:

1. **Requerimientos funcionales**- Son enunciados acerca de servicios que el sistema debe proveer, de **cómo debería reaccionar el sistema a entradas particulares** y de cómo debería comportarse el sistema en situaciones específicas. En algunos casos, los requerimientos funcionales también explican lo que no debe hacer el sistema.

Registrar – consultar – buscar – calcular – mostrar - controlar.

2. **Requerimientos no funcionales** - Son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de

desarrollo, como las impuestas por los estándares. Los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del sistema.

– el sistema también tiene respuestas para decirte que no lo va a hacer

4 formas de respuestas de los sistemas –

No hacer nada - poner letras en campos numericos

Usar una omisión - si no pone color el objeto se crea color gris

Avisar al usuario que corrija – y continuar

Avisar y si el error es grande terminar la ejecución del sistema.

NOTA: Un requerimiento de un usuario interesado por la seguridad, como el enunciado que limita el acceso a usuarios autorizados, parecería un requerimiento no funcional. Sin embargo, cuando se desarrolla con más detalle, este requerimiento puede generar otros requerimientos que son evidentemente funcionales, como la necesidad de incluir facilidades de autenticación en el sistema.

Esto muestra que los requerimientos no son independientes y que un requerimiento genera o restringe normalmente otros requerimientos. Por lo tanto, los requerimientos del sistema no sólo detallan los servicios o las características que se requieren del mismo, sino también especifican la **funcionalidad necesaria** para asegurar que estos servicios y características se entreguen de manera adecuada.

Requerimientos funcionales

Estos dependen del tipo de software que se esté desarrollando, de los usuarios esperados del software y del enfoque general que adopta la organización cuando se escriben los requerimientos. Al expresarse como requerimientos del usuario, los requerimientos funcionales se describen por lo general de forma abstracta que entiendan los usuarios del sistema. Sin embargo, requerimientos funcionales más específicos del sistema detallan las funciones del sistema, sus entradas y salidas, sus excepciones, etcétera.

Los requerimientos funcionales del sistema varían desde requerimientos generales que cubren lo que tiene que hacer el sistema, hasta requerimientos muy específicos que reflejan maneras locales de trabajar o los sistemas existentes de una organización. Por ejemplo, veamos algunos casos de requerimientos funcionales para el sistema MHC-PMS, que se usan para mantener información de pacientes que reciben tratamiento por problemas de salud mental:

1. Un usuario podrá buscar en todas las clínicas las listas de citas.
2. El sistema elaborará diariamente, para cada clínica, una lista de pacientes que se espera que asistan a cita ese día.
3. Cada miembro del personal que usa el sistema debe identificarse de manera individual con su número de ocho dígitos.

La inexactitud en la especificación de requerimientos causa muchos problemas en la ingeniería de software. Es natural que un desarrollador de sistemas interprete un requerimiento ambiguo de forma que simplifique su implementación.

Requerimientos de dominio - no todos los requerimientos salen del usuario.

Los requerimientos de dominio se derivan del dominio de aplicación del sistema, más que a partir de las necesidades específicas de los usuarios del sistema. Pueden ser requerimientos funcionales nuevos por derecho propio, restricciones a los requerimientos funcionales existentes o formas en que deben realizarse cosas particulares. (Ej. los sistemas de facturación) - en este caso los usuarios se dan cuenta de que los procesos se pueden hacer de otra forma. Como estas sistematizando – descubres cosas – preguntar se conoce o existen normas, o comités de desarrollo (tienes que ir e **investigar**).

El problema con los requerimientos de dominio es que los ingenieros de software “**no pueden entender**” las características del dominio en que opera el sistema. Por lo común, no pueden indicar si un requerimiento de dominio se perdió o entró en conflicto con otros requerimientos.

<http://www.SoftwareEngineering-9.com/Web/Requirements/DomainReq.html>

Dominio es el campo de conocimiento al que se refiere un problema.

Contabilidad - Sistemas de Control- Financieros – Recursos Humanos - Matemáticos

- ANALIZAR LOS PROCESOS Y SUS REGLAS – **Ingeniería de conocimiento**
- Es casi imposible lograr la consistencia y la totalidad de los requerimientos.

DESARROLLO DE PROYECTOS:

Stakeholders – personas que participan en la definición de requisitos

Ingeniero de proyecto – Project Manager.

Juicio de expertos – expertos.

Patrocinadores – intervención de los patrocinadores.

NOTA: no olvidar que se puede tratar de proyectos pequeños

Requerimientos no funcionales.

Estos se relacionan directamente con los servicios específicos que el sistema entrega a sus usuarios. Pueden relacionarse con propiedades emergentes del sistema, **como fiabilidad, tiempo de respuesta y uso de almacenamiento.**

De forma alternativa, pueden definir restricciones sobre la implementación del sistema, como las capacidades de los dispositivos I/O o las representaciones de datos usados en las interfaces con otros sistemas.

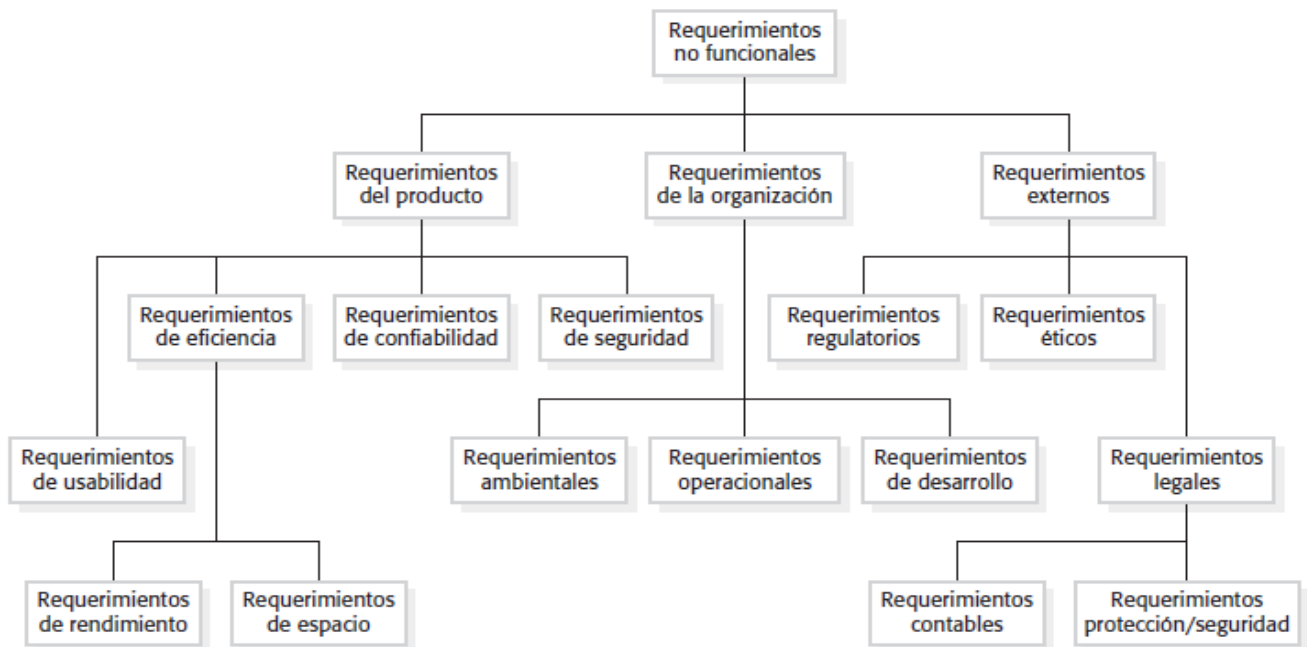
Los requerimientos no funcionales, como el rendimiento, la seguridad o la disponibilidad, especifican o restringen por lo general características del sistema como un todo.

“Los requerimientos no funcionales a menudo son más significativos que los requerimientos funcionales” individuales. Es común que los usuarios del sistema encuentren formas para trabajar en torno a una función del sistema que realmente no cubre sus necesidades.

Lo difícil de los no funcionales es detectarlos y una vez detectados tenemos que determinar como resolverlo, por ejemplo, buscar un mejor algoritmo, optimizar algún recurso o implementar un requerimiento funcional que cuando menos avise o indique que es lo que el sistema no puede hacer.

Lo que no nos podemos permitir es que el programa resuelva MAL un problema. “BUGS”

```
Int i = 0;    if( i == 0) --- NO - 0.000000000001
```



A partir de este diagrama que los requerimientos no funcionales provienen de características requeridas del software (requerimientos del producto), la organización que desarrolla el software (requerimientos de la organización) o de fuentes externas:

1. **Requerimientos del producto** Estos requerimientos especifican o restringen el comportamiento del software. Los ejemplos incluyen requerimientos de rendimiento sobre qué tan rápido se debe ejecutar el sistema y cuánta memoria requiere, requerimientos de fiabilidad que establecen la tasa aceptable de fallas, requerimientos de seguridad y requerimientos de usabilidad.
2. **Requerimientos de la organización** Son requerimientos de sistemas amplios, derivados de políticas y procedimientos en la organización del cliente y del desarrollador. Los ejemplos incluyen requerimientos del proceso operacional que definen cómo se usará el sistema, requerimientos del proceso de desarrollo que especifican el lenguaje de programación, estándares del entorno o el proceso de desarrollo a utilizar, y requerimientos ambientales que definen el entorno de operación del sistema.
3. **Requerimientos externos** Este término cubre todos los requerimientos derivados de factores externos al sistema y su proceso de desarrollo. En ellos se incluyen requerimientos regulatorios que establecen lo que debe hacer el sistema para ser aprobado en su uso por un regulador, como sería un banco central; requerimientos legislativos que tienen que seguirse para garantizar que el sistema opere conforme a la ley, y requerimientos éticos que garanticen que el sistema será aceptable para sus usuarios y el público en general.

Ejemplo:

REQUERIMIENTO DEL PRODUCTO

El MHC-PMS estará disponible en todas las clínicas durante las horas de trabajo normales (lunes a viernes, de 8:30 a 17:30). En cualquier día, los tiempos muertos dentro de las horas laborales normales no rebasarán los cinco segundos.

REQUERIMIENTOS DE LA ORGANIZACIÓN

Los usuarios del sistema MHC-PMS se acreditarán a sí mismos con el uso de la tarjeta de identidad de la autoridad sanitaria.

REQUERIMIENTOS EXTERNOS

Como establece la HStan-03-2006-priv, el sistema implementará provisiones para la privacidad del paciente. **ACERCARSE A LAS NORMAS Y ESTANDARES**

Un problema común con requerimientos no funcionales es que los usuarios o clientes con frecuencia proponen estos requerimientos como metas generales, como facilidad de uso, capacidad de que el sistema se recupere de fallas, o rapidez de respuesta al usuario.

Las metas establecen buenas intenciones; no obstante, ocasionan problemas a los desarrolladores del sistema, pues dejan espacio para la interpretación y la disputa posterior

una vez que se entregue el sistema. Por ejemplo, la siguiente meta del sistema es típica de cómo un administrador expresa los requerimientos de usabilidad:

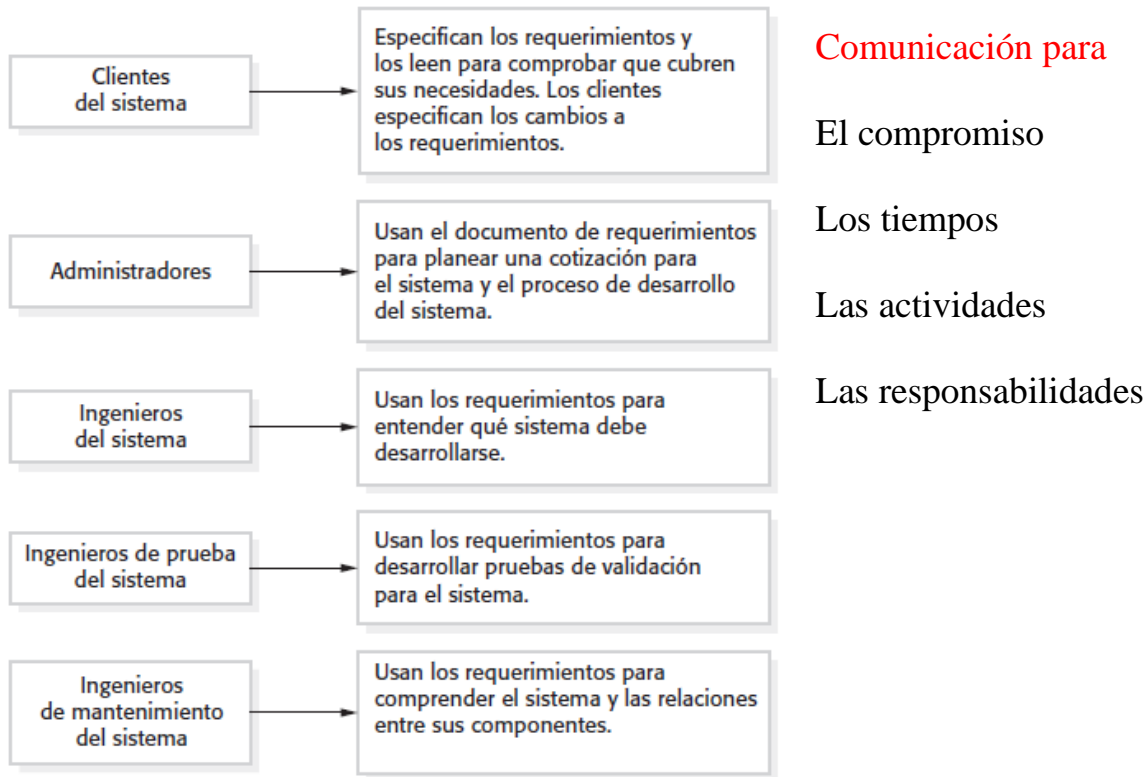
Para el personal médico debe ser fácil usar el sistema, y este último debe organizarse de tal forma que minimice los errores del usuario.

Propiedad	Medida
Rapidez	Transacciones/segundo procesadas Tiempo de respuesta usuario/evento Tiempo de regeneración de pantalla
Tamaño	Mbytes Número de chips ROM
Facilidad de uso	Tiempo de capacitación Número de cuadros de ayuda
Fiabilidad	Tiempo medio para falla Probabilidad de indisponibilidad Tasa de ocurrencia de falla Disponibilidad
Robustez	Tiempo de reinicio después de falla Porcentaje de eventos que causan falla Probabilidad de corrupción de datos en falla
Portabilidad	Porcentaje de enunciados dependientes de objetivo Número de sistemas objetivo

**METRICAS PARA
ESPECIFICAR
REQUERIMIENTOS
NO FUNCIONALES**

DOCUMENTO DE REQUERIMIENTOS DE SOFTWARE (SRS)

El documento de requerimientos tiene un conjunto variado de usuarios, desde el administrador ejecutivo de la organización que paga por el sistema, hasta los ingenieros responsables del desarrollo del software. La figura 4.6, tomada del libro del autor con Gerald Kotonya sobre ingeniería de requerimientos (Kotonya y Sommerville, 1998), muestra a los posibles usuarios del documento y cómo ellos lo utilizan, ellos son:



La siguiente tabla indica **una posible organización** para un documento de requerimientos basada en un estándar del IEEE para documentos de requerimientos (IEEE, 1998).

Capítulo	Descripción
Prefacio	Debe definir el número esperado de lectores del documento, así como describir su historia de versiones, incluidas las causas para la creación de una nueva versión y un resumen de los cambios realizados en cada versión.
Introducción	Describe la necesidad para el sistema. Debe detallar brevemente las funciones del sistema y explicar cómo funcionará con otros sistemas. También tiene que indicar cómo se ajusta el sistema en los objetivos empresariales o estratégicos globales de la organización que comisiona el software.
Glosario	Define los términos técnicos usados en el documento. No debe hacer conjeturas sobre la experiencia o la habilidad del lector.
Definición de requerimientos del usuario	Aquí se representan los servicios que ofrecen al usuario. También, en esta sección se describen los requerimientos no funcionales del sistema. Esta descripción puede usar lenguaje natural, diagramas u otras observaciones que sean comprensibles para los clientes. Deben especificarse los estándares de producto y proceso que tienen que seguirse.
Arquitectura del sistema	Este capítulo presenta un panorama de alto nivel de la arquitectura anticipada del sistema, que muestra la distribución de funciones a través de los módulos del sistema. Hay que destacar los componentes arquitectónicos que sean de reutilización.
Especificación de requerimientos del sistema	Debe representar los requerimientos funcionales y no funcionales con más detalle. Si es preciso, también pueden detallarse más los requerimientos no funcionales. Pueden definirse las interfaces a otros sistemas.
Modelos del sistema	Pueden incluir modelos gráficos del sistema que muestren las relaciones entre componentes del sistema, el sistema y su entorno. Ejemplos de posibles modelos son los modelos de objeto, modelos de flujo de datos o modelos de datos semánticos.
Evolución del sistema	Describe los supuestos fundamentales sobre los que se basa el sistema, y cualquier cambio anticipado debido a evolución de hardware, cambio en las necesidades del usuario, etc. Esta sección es útil para los diseñadores del sistema, pues los ayuda a evitar decisiones de diseño que restringirían probablemente futuros cambios al sistema.
Apéndices	Brindan información específica y detallada que se relaciona con la aplicación a desarrollar; por ejemplo, descripciones de hardware y bases de datos. Los requerimientos de hardware definen las configuraciones, mínima y óptima, del sistema. Los requerimientos de base de datos delimitan la organización lógica de los datos usados por el sistema y las relaciones entre datos.
Índice	Pueden incluirse en el documento varios índices. Así como un índice alfabético normal, uno de diagramas, un índice de funciones, etcétera.

La especificación de requerimientos es el proceso de escribir, en un documento de requerimientos, los requerimientos del usuario y del sistema.

De manera ideal, los requerimientos del usuario y del sistema deben ser claros, sin ambigüedades, comprensibles aún para los que no cuentan con un conocimiento técnico detallado, completos y consistentes.

Los requerimientos del usuario para un sistema deben describir los requerimientos funcionales y no funcionales

El documento de requerimientos **no debe incluir detalles de la arquitectura o el diseño del sistema**. En consecuencia, si usted escribe los requerimientos del usuario, no tiene que usar jerga de software, anotaciones estructuradas o formales.

Si el proceso es de cascada los requisitos deben ser descritos en su totalidad y a detalle, si el proceso es incremental se puede partir de lo general hacia lo particular en cada etapa o iteración del proyecto.

Notación	Descripción
Enunciados en lenguaje natural	Los requerimientos se escriben al usar enunciados numerados en lenguaje natural. Cada enunciado debe expresar un requerimiento.
Lenguaje natural estructurado	Los requerimientos se escriben en lenguaje natural en una forma o plantilla estándar. Cada campo ofrece información de un aspecto del requerimiento.
Lenguajes de descripción de diseño	Este enfoque usa un lenguaje como un lenguaje de programación, pero con características más abstractas para especificar los requerimientos al definir un modelo operacional del sistema. Aunque en la actualidad este enfoque se usa raras veces, aún tiene utilidad para especificaciones de interfaz.
Anotaciones gráficas	Los modelos gráficos, complementados con anotaciones de texto, sirven para definir los requerimientos funcionales del sistema; los casos de uso del UML y los diagramas de secuencia se emplean de forma común.
Especificaciones matemáticas	Dichas anotaciones se basan en conceptos matemáticos como máquinas o conjuntos de estado finito. Aunque tales especificaciones sin ambigüedades pueden reducir la imprecisión en un documento de requerimientos, la mayoría de los clientes no comprenden una especificación formal. No pueden comprobar que representa lo que quieren y por ello tienen reticencia para aceptarlo como un contrato de sistema.

Formas de escribir una especificación de requerimientos de sistema.

En la mayoría de los casos, los sistemas deben interoperar con los sistemas existentes, lo cual restringe el diseño e impone requerimientos sobre el nuevo sistema.

Especificación en lenguaje natural

Desde los albores de la ingeniería de software, el lenguaje natural se usa para escribir los requerimientos de software. Es expresivo, intuitivo y universal. También es potencialmente vago, ambiguo y su significado depende de los antecedentes del lector. Como resultado, hay muchas propuestas para formas alternativas de escribir los requerimientos. Sin embargo, ninguna se ha adoptado de manera amplia, por lo que el lenguaje natural seguirá siendo la forma más usada para especificar los requerimientos del sistema y del software.

Bomba de insulina/Software de control/SRS/3.3.2

Función	Calcula dosis de insulina: nivel seguro de azúcar.
Descripción	Calcula la dosis de insulina que se va a suministrar cuando la medición del nivel de azúcar actual esté en zona segura entre 3 y 7 unidades.
Entradas	Lectura del azúcar actual (r2), las dos lecturas previas (r0 y r1).
Fuente	Lectura del azúcar actual del sensor. Otras lecturas de la memoria.
Salidas	CompDose: la dosis de insulina a administrar.
Destino	Ciclo de control principal.
Acción	CompDose es cero si es estable el nivel de azúcar, o cae o si aumenta el nivel pero disminuye la tasa de aumento. Si el nivel se eleva y la tasa de aumento crece, CompDose se calcula entonces al dividir la diferencia entre el nivel de azúcar actual y el nivel previo entre 4 y redondear el resultado. Si la suma se redondea a cero, en tal caso CompDose se establece en la dosis mínima que puede entregarse.
Requerimientos	Dos lecturas previas, de modo que puede calcularse la tasa de cambio del nivel de azúcar.
Precondición	El depósito de insulina contiene al menos la dosis individual de insulina máxima permitida.
Postcondición	r0 se sustituye con r1, luego r1 se sustituye con r2.
Efectos colaterales	Ninguno.

¿ESTO PASA SIEMPRE? - NO DEBO OMITIR NADA “NI UNA COSITA”

Ingeniería del conocimiento

[Ir a la navegación](#) - [Ir a la búsqueda](#)

La **ingeniería del conocimiento** es aquella disciplina moderna que forma parte de la [Inteligencia Artificial](#) y cuyo fin es el diseño y desarrollo de [Sistemas Expertos](#). Para esto, se apoya en metodologías instruccionales y en las ciencias de la [computación](#) y de las tecnologías de la información, intentando representar el conocimiento y razonamiento humanos en un determinado dominio, dentro de un sistema artificial.

El trabajo de los ingenieros del conocimiento consiste en extraer el conocimiento de los expertos humanos en una determinada área, y en codificar dicho conocimiento de manera que pueda ser procesado por un sistema.

El problema es que el ingeniero del conocimiento no es un experto en el campo que intenta modelar, mientras que el experto en el tema no tiene experiencia modelando su conocimiento (basado en la heurística) de forma que pueda ser representado de forma genérica en un sistema.

La ingeniería del conocimiento engloba a los científicos, tecnología y metodología necesarios para procesar el conocimiento. Su objetivo es extraer, articular e informatizar el conocimiento de un experto.

Especificaciones estructuradas

Aunque este enfoque conserva la mayoría de la expresividad y comprensibilidad del lenguaje natural, asegura que haya cierta uniformidad sobre la especificación.

Las anotaciones en lenguaje estructurado emplean plantillas para especificar requerimientos del sistema. La especificación utiliza constructos de lenguaje de programación para mostrar alternativas e iteración, y destaca elementos clave con el uso de sombreado o de fuentes distintas.

Los Robertson (Robertson y Robertson, 1999), en su libro del método de ingeniería de requerimientos VOLERE, recomiendan que se escriban los requerimientos del usuario inicialmente en tarjetas, un requerimiento por tarjeta.

Condición	Acción
Nivel de azúcar en descenso ($r_2 < r_1$)	CompDose = 0
Nivel de azúcar estable ($r_2 = r_1$)	CompDose = 0
Nivel de azúcar creciente y tasa de incremento decreciente ($(r_2 - r_1) < (r_1 - r_0)$)	CompDose = 0
Nivel de azúcar creciente y tasa de incremento estable o creciente ($(r_2 - r_1) \geq (r_1 - r_0)$)	CompDose = round $((r_2 - r_1)/4)$ If resultado redondeado = 0 then CompDose = MinimumDose

Procesos de ingeniería de requerimientos

Los procesos de ingeniería de requerimientos incluyen **cuatro actividades de alto nivel**.

Éstas se enfocan en valorar si el sistema es útil para la empresa (**estudio de factibilidad**),
Tiempo -Costo - Recursos: materiales - maquinas, servidores

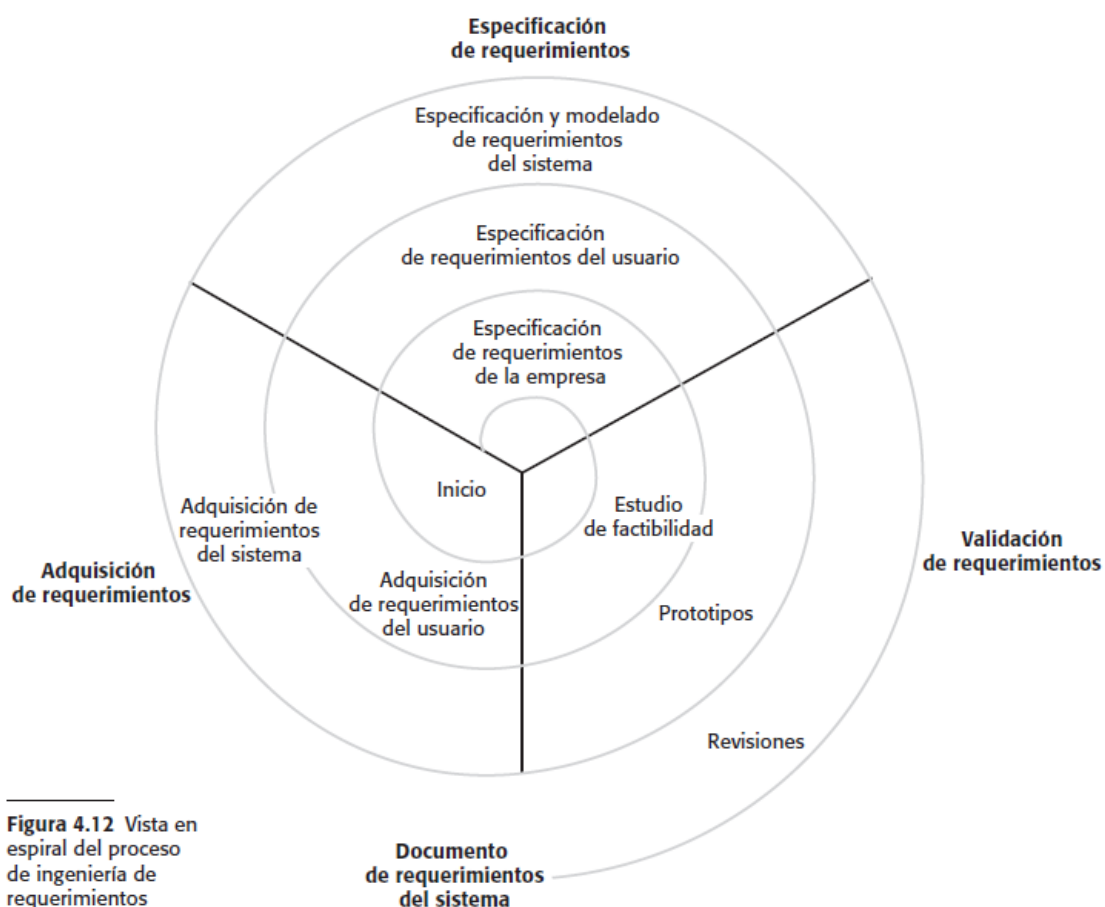
CONOCIMIENTO

Descubrir requerimientos (**adquisición y análisis**),

Convertir dichos requerimientos en alguna forma estándar (**especificación**)

y comprobar que los requerimientos definan realmente el sistema que quiere el cliente (**validación**) de los requerimientos.

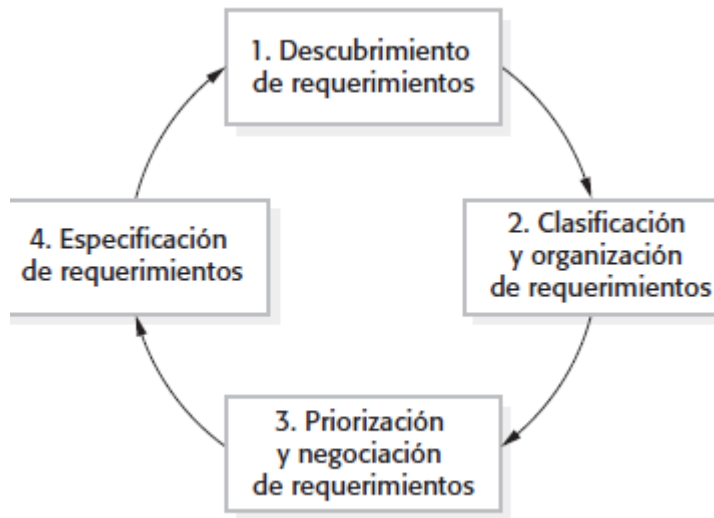
En la práctica, la ingeniería de requerimientos es un proceso iterativo donde las actividades están entrelazadas.



Adquisición y análisis de requerimientos

Después de un estudio de factibilidad inicial, la siguiente etapa del proceso de ingeniería de requerimientos es la adquisición y el análisis de requerimientos.

En esta actividad, los ingenieros de software trabajan con clientes y usuarios finales del sistema para **descubrir el dominio de aplicación**, qué servicios debe proporcionar el sistema, el desempeño requerido de éste, las restricciones de hardware, etcétera.



OPTIMIZAR

Lo mejor para uno no siempre es lo mejor para otros.

Entonces buscamos lo mejor para todos.

Un participante en el sistema (**stakeholders**) es quien debe tener alguna influencia directa o indirecta sobre los requerimientos de este.

Las actividades del proceso son:

1. **Descubrimiento de requerimientos** Éste es el proceso de interactuar con los participantes del sistema para descubrir sus requerimientos. También son los requerimientos de dominio de los **participantes** (stakeholders) y la documentación se descubren durante esta actividad.

Clasificar como no funcional y funcional.

2. **Clasificación y organización de requerimientos** Esta actividad toma la compilación no estructurada de requerimientos, **agrupa requerimientos relacionados y los organiza en grupos coherentes**. La forma más común de agrupar requerimientos es usar un modelo de la arquitectura del sistema, para identificar subsistemas y asociar los requerimientos con cada subsistema. En la práctica, la ingeniería de requerimientos y el diseño arquitectónico no son actividades separadas completamente.

3. **Priorización y negociación de requerimientos** Inevitablemente, cuando intervienen diversos participantes, los requerimientos entrarán en conflicto. Esta actividad se

preocupa por priorizar los requerimientos, así como por encontrar y resolver conflictos de requerimientos mediante la negociación. Por lo general, los participantes tienen que reunirse para resolver las diferencias y estar de acuerdo con el compromiso de los requerimientos.

4. **Especificación de requerimientos** Los requerimientos se documentan e ingresan en la siguiente ronda de la espiral. Pueden producirse documentos de requerimientos formales o informales.

La adquisición y la comprensión de los requerimientos por parte de los participantes del sistema es un proceso difícil por diferentes razones:

1. Los participantes con frecuencia no saben lo que quieren de un sistema de cómputo, excepto en términos muy generales; pueden encontrar difícil articular qué quieren que haga el sistema; pueden hacer peticiones inalcanzables porque no saben qué es factible y qué no lo es.
2. Los participantes en un sistema expresan naturalmente los requerimientos con sus términos y conocimientos implícitos de su trabajo. Los ingenieros de requerimientos, sin experiencia en el dominio del cliente, podrían no entender dichos requerimientos.
3. Diferentes participantes tienen distintos requerimientos y pueden expresarlos en variadas formas. Los ingenieros de requerimientos deben descubrir todas las fuentes potenciales de requerimientos e identificar similitudes y conflictos.
4. Factores políticos llegan a influir en los requerimientos de un sistema. Los administradores pueden solicitar requerimientos específicos del sistema, porque éstos les permitirán aumentar su influencia en la organización.
5. El ambiente económico y empresarial donde ocurre el análisis es dinámico. Inevitablemente cambia durante el proceso de análisis. Puede cambiar la importancia de requerimientos particulares; o bien, tal vez surjan nuevos requerimientos de nuevos participantes a quienes no se consultó originalmente.

Descubrimiento de requerimientos

El descubrimiento de requerimientos (llamado a veces adquisición de requerimientos) es el proceso de recopilar información sobre el sistema requerido y los sistemas existentes, así como de separar, a partir de esta información, los requerimientos del usuario y del sistema.

Las fuentes de información durante la fase de descubrimiento de requerimientos incluyen documentación, participantes del sistema y especificaciones de sistemas similares. La interacción con los participantes es a través de entrevistas y observaciones, y pueden usarse escenarios y prototipos para ayudar a los participantes a entender cómo será el sistema.

Los participantes varían desde administradores y usuarios finales de un sistema hasta participantes externos como los reguladores, quienes certifican la aceptabilidad del sistema.

Por ejemplo, los participantes que se incluyen para el sistema de información de pacientes en atención a la salud mental son:

1. Pacientes cuya información se registra en el sistema.
2. Médicos que son responsables de valorar y tratar a los pacientes.
3. Enfermeros que coordinan, junto con los médicos, las consultas y suministran algunos tratamientos.
4. Recepcionistas que administran las citas médicas de los pacientes.
5. Personal de TI que es responsable de instalar y mantener el sistema.
6. Un director de ética médica que debe garantizar que el sistema cumpla con los lineamientos éticos actuales de la atención al paciente.
7. Encargados de atención a la salud que obtienen información administrativa del sistema.
8. Personal de archivo médico que es responsable de garantizar que la información del sistema se conserve, y se implementen de manera adecuada los procedimientos de mantenimiento del archivo.

Además de los participantes del sistema, se observa que los requerimientos también pueden venir del dominio de aplicación y de otros sistemas que interactúan con el sistema a especificar. Todos ellos deben considerarse durante el proceso de adquisición de requerimientos.

Todas estas diferentes fuentes de requerimientos (participantes, dominio, sistemas) se representan como puntos de vista del sistema, y cada visión muestra un subconjunto de los requerimientos para el sistema.

Diferentes puntos de vista de un problema enfocan el problema de diferentes formas. Sin embargo, sus perspectivas no son totalmente independientes, sino que por lo general se traslapan, de manera que tienen requerimientos comunes. Usted puede usar estos puntos de vista para estructurar tanto el descubrimiento como la documentación de los requerimientos del sistema.

Descubrir: Técnica Entrevistas

Los requerimientos que se derivan de entrevistas pueden ser de dos tipos:

- 1. Entrevistas cerradas**, donde los participantes responden a un conjunto de preguntas preestablecidas.
- 2. Entrevistas abiertas**, en las cuales no hay agenda predefinida. El equipo de ingeniería de requerimientos explora un rango de conflictos con los participantes del sistema y, como resultado, desarrolla una mejor comprensión de sus necesidades.

A las personas les gusta hablar acerca de sus trabajos, así que por lo general están muy dispuestas a participar en entrevistas. Sin embargo, las entrevistas no son tan útiles para comprender los requerimientos desde el dominio de la aplicación.

Por dos razones resulta difícil asimilar el conocimiento de dominio a través de entrevistas:

1. Todos los especialistas en la aplicación usan terminología y jerga que son específicos de un dominio. Es imposible que ellos discutan los requerimientos de dominio sin usar este tipo de lenguaje. Por lo general, usan la terminología en una forma precisa y sutil, que para los ingenieros de requerimientos es fácil de malinterpretar.
2. Cierta conocimiento del dominio es tan familiar a los participantes que encuentran difícil de explicarlo, o bien, creen que es tan fundamental que no vale la pena mencionarlo. Por ejemplo, para un bibliotecario no es necesario decir que todas las adquisiciones deben catalogarse antes de agregarlas al acervo. Sin embargo, esto quizá no sea obvio para el entrevistador y, por lo tanto, es posible que no lo tome en cuenta en los requerimientos.

Descubrir: Técnica Escenarios

Por lo general, las personas encuentran más sencillo vincularse con ejemplos reales que con descripciones abstractas. Pueden comprender y criticar un escenario sobre cómo interactuar con un sistema de software. Los ingenieros de requerimientos usan la información obtenida de esta discusión para formular los verdaderos requerimientos del sistema.

Los escenarios son particularmente útiles para detallar un bosquejo de descripción de requerimientos. Se trata de ejemplos sobre descripciones de sesiones de interacción. Cada escenario abarca comúnmente una interacción o un número pequeño de interacciones posibles.

Se desarrollan diferentes formas de escenarios y se ofrecen varios tipos de información con diversos niveles de detalle acerca del sistema. Las historias que se usan en programación extrema, estudiadas en el capítulo 3, son un tipo de escenario de requerimientos.

Un escenario comienza con un bosquejo de la interacción. Durante el proceso de adquisición, se suman detalles a éste para crear una representación completa de dicha interacción. En su forma más general, un escenario puede incluir:

1. Una descripción de qué esperan el sistema y los usuarios cuando inicia el escenario.
2. Una descripción en el escenario del flujo normal de los eventos.
3. Una descripción de qué puede salir mal y cómo se manejaría.
4. Información de otras actividades que estén en marcha al mismo tiempo.
5. Una descripción del estado del sistema cuando termina el escenario.

SUPOSICIÓN INICIAL:

El paciente observa a un auxiliar médico que elabora un registro en el sistema y recaba información personal de aquél (nombre, dirección, edad, etcétera). Una enfermera ingresa en el sistema y obtiene la historia médica.

NORMAL:

La enfermera busca al paciente por su nombre completo. Si hay más de un paciente con el mismo apellido, para identificarlo se usa el nombre y la fecha de nacimiento.

La enfermera elige la opción de menú y añade la historia médica.

Inmediatamente la enfermera sigue una serie de indicadores (*prompt*) del sistema para ingresar información de consultas en otras instituciones, sobre problemas de salud mental (entrada libre de texto), condiciones médicas existentes (la enfermera selecciona las condiciones del menú), medicamentos administrados actualmente (seleccionados del menú), alergias (texto libre) y vida familiar (formato).

QUÉ PUEDE SALIR MAL:

Si no existe el registro del paciente o no puede encontrarse, la enfermera debe crear un nuevo registro e ingresar información personal.

Las condiciones o los medicamentos del paciente no se ingresan en el menú. La enfermera debe elegir la opción “otro” e ingresar texto libre que describa la condición/medicamento.

El paciente no puede/no proporciona información acerca de su historia médica.

La enfermera tiene que ingresar a texto libre que registre la incapacidad/renuencia a brindar información. El sistema debe imprimir el formato de exclusión estándar que menciona que la falta de información podría significar que el tratamiento esté limitado o demorado. Esto tiene que firmarlo el paciente.

OTRAS ACTIVIDADES:

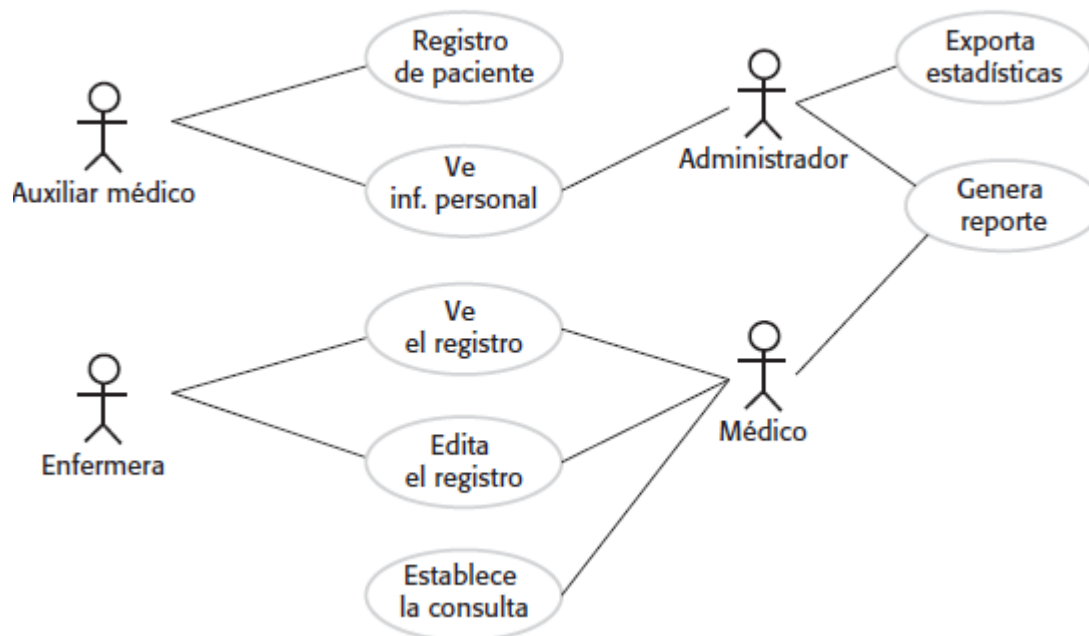
Mientras se ingresa la información, otros miembros del personal pueden consultar los registros, pero no editarlos.

ESTADO DEL SISTEMA A COMPLETAR:

Ingreso del usuario. El registro del paciente, incluida su historia médica, se integra en la base de datos, se agrega un registro a la bitácora (log) del sistema que indica el tiempo de inicio y terminación de la sesión y la enfermera a cargo.

Descubrir: Técnica Casos de uso

Los casos de uso son una técnica de descubrimiento de requerimientos que se introdujo por primera vez en el método Objectory (Jacobson *et al.*, 1993). Ahora se ha convertido en una característica fundamental del modelado de lenguaje unificado. UML.



Los casos de uso identifican las interacciones individuales entre el sistema y sus usuarios u otros sistemas. Cada caso de uso debe documentarse con una descripción textual.

Entonces pueden vincularse con otros modelos en el **UML** que desarrollará el escenario con más detalle.

Descubrir: Técnica Etnografía

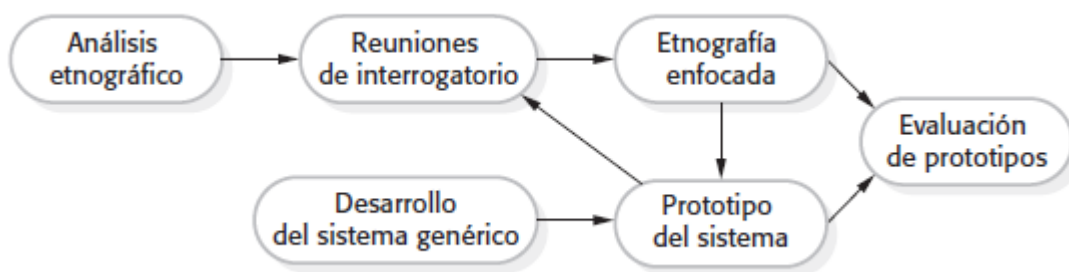
Los sistemas de software no existen aislados. **Se usan en un contexto social y organizacional**, y dicho escenario podría derivar o restringir los requerimientos del sistema

de software. A menudo satisfacer dichos requerimientos sociales y organizacionales es crítico para el éxito del sistema.

La etnografía es una técnica de observación que se usa para entender los procesos operacionales y ayudar a derivar requerimientos de apoyo para dichos procesos.

Un analista se adentra en el ambiente laboral donde se usará el sistema. Observa el trabajo diario

y toma notas acerca de las tareas existentes en que intervienen los participantes. El valor de la etnografía es que ayuda a descubrir requerimientos implícitos del sistema que reflejan las formas actuales en que trabaja la gente, en vez de los procesos formales definidos por la organización.



general, el uso de la etnografía en el diseño de sistemas. El autor ha investigado métodos para integrar la etnografía en el proceso de ingeniería de software, mediante su vinculación con los métodos de la ingeniería de requerimientos.

Descubrir: Técnica Revisar el mercado de software

En el dominio del software que queremos implementar: con costo o sin costo.

Observar requerimientos funcionales del software – para que inventar cosas cuando los usuarios trabajan perfectamente en un determinado ambiente.

Algunos elementos funcionales se convierten en paradigmas.

Que se está haciendo, como se está haciendo – esto implica que el proyecto de desarrollo de software se pueda cancelar, terminando con la compra de software o uso de software libre, siempre y cuando en este último, exista una persona que lo respalde.

Si ya está hecho – y tiene un costo accesible-

-COMO SE TRABAJA EN EL ÁREA – conocer ese mercado

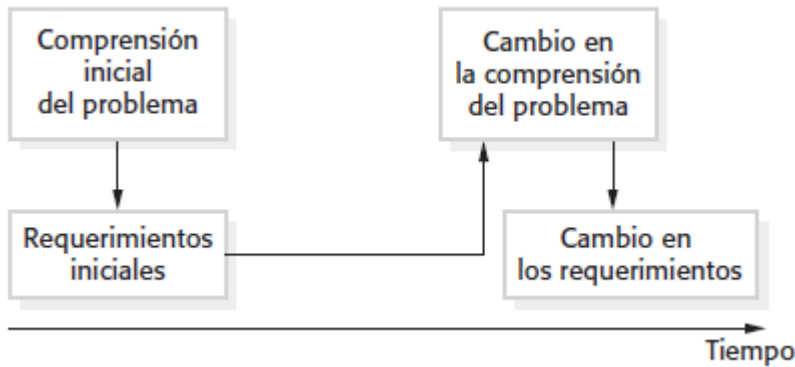
Validación de requerimientos

La validación de requerimientos es el proceso de verificar que los requerimientos definan realmente el sistema que en verdad quiere el cliente.

Dichas comprobaciones incluyen:

1. **Comprobaciones de validez** - Un usuario quizá crea que necesita un sistema para realizar ciertas funciones. Sin embargo, con mayor consideración y análisis se logra identificar las funciones adicionales o diferentes que se requieran. Los sistemas tienen diversos participantes con diferentes necesidades, y cualquier conjunto de requerimientos es inevitablemente un compromiso a través de la comunidad de participantes. Verificar si se hacen las cosas como se plantean – x ej. revisar procedimientos, normas, estándares, leyes vigentes.
2. **Comprobaciones de consistencia** Los requerimientos en el documento no deben estar en conflicto. Esto es, no debe haber restricciones contradictorias o descripciones diferentes de la misma función del sistema.
3. **Comprobaciones de totalidad** El documento de requerimientos debe incluir requerimientos que definan todas las funciones y las restricciones pretendidas por el usuario del sistema. Dentro del DOMINIO – ver TODO lo que va a hacer el sistema (MENUS)
4. **Comprobaciones de realismo** Al usar el conocimiento de la tecnología existente, los requerimientos deben comprobarse para garantizar que en realidad pueden implementarse. objetivos alcanzables computacionalmente hablando.
5. Dichas comprobaciones también tienen que considerar el presupuesto y la fecha para el desarrollo del sistema.
6. **Verificabilidad** Para reducir el potencial de disputas entre cliente y contratista, los requerimientos del sistema deben escribirse siempre de manera que sean verificables.

Esto significa que usted debe ser capaz de escribir un conjunto de pruebas que demuestren que el sistema entregado cumpla cada requerimiento especificado.



Hay algunas técnicas de validación de requerimientos que se usan individualmente o en conjunto con otras:

1. **Revisiones de requerimientos** Los requerimientos se analizan sistemáticamente usando un equipo de revisores que verifican errores e inconsistencias.
2. **Creación de prototipos** En esta aproximación a la validación, se muestra un modelo ejecutable del sistema en cuestión a los usuarios finales y clientes. Así, ellos podrán experimentar con este modelo para constatar si cubre sus necesidades reales.
3. **Generación de casos de prueba** Los requerimientos deben ser comprobables. Si las pruebas para los requerimientos se diseñan como parte del proceso de validación, esto revela con frecuencia problemas en los requerimientos. Si una prueba es difícil o imposible de diseñar, esto generalmente significa que los requerimientos serán difíciles de implementar, por lo que deberían reconsiderarse. El desarrollo de pruebas a partir de los requerimientos del usuario antes de escribir cualquier código es una pieza integral de la programación extrema.

Gestión de requerimientos –

Este es el marco de trabajo en el que se desarrolla todo lo anterior. De aquí para abajo es lo primero en que hay que empezar.

La administración de requerimientos es el proceso de comprender y controlar los cambios en los requerimientos del sistema.

Los requerimientos para los grandes sistemas de software siempre cambian.

Como los problemas no se logran definir por completo, los requerimientos del software están condenados también a estar incompletos. Durante el proceso de software, la comprensión que los participantes tienen

de los problemas cambia constantemente. Entonces, los requerimientos del sistema también deben evolucionar para reflejar esa visión cambiante del problema.

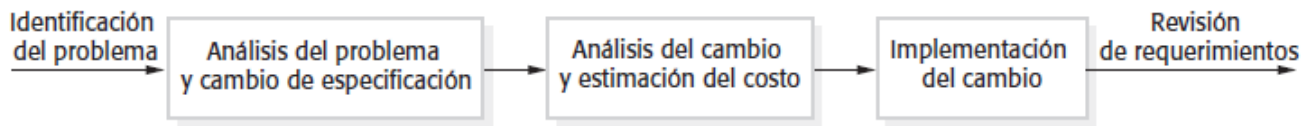
Existen muchas razones por las que es inevitable el cambio:

1. Los ambientes empresarial y técnico del sistema siempre cambian después de la instalación. Puede introducirse nuevo hardware, y quizá sea necesario poner en interfaz el sistema con otros sistemas, cambiar las prioridades de la empresa (con los consecuentes cambios en el sistema de apoyo requerido) e introducir nuevas leyes y regulaciones que el sistema deba cumplir cabalmente.
2. Los individuos que pagan por un sistema y los usuarios de dicho sistema, por lo general, no son los mismos. Los clientes del sistema imponen requerimientos debido a restricciones organizativas y presupuestales. Esto podría estar en conflicto con los requerimientos del usuario final y, después de la entrega, probablemente deban agregarse nuevas características para apoyar al usuario, si el sistema debe cubrir sus metas.
3. Los sistemas grandes tienen regularmente una comunidad de usuarios diversa, en la cual muchos individuos tienen diferentes requerimientos y prioridades que quizás estén en conflicto o sean contradictorios. Los requerimientos finales del sistema inevitablemente tienen un compromiso entre sí y, con la experiencia, a menudo se descubre que el equilibrio de apoyo brindado a diferentes usuarios tiene que cambiar.

Planeación de la administración de requerimientos

La planeación es una primera etapa esencial en el proceso de administración de requerimientos.

Esta etapa establece el nivel de detalle que se requiere en la administración de requerimientos. Durante la etapa de administración de requerimientos, usted tiene que decidir sobre:



1. *Identificación de requerimientos* Cada requerimiento debe identificarse de manera exclusiva, de forma que pueda tener referencia cruzada con otros requerimientos y usarse en las evaluaciones de seguimiento.
2. *Un proceso de administración del cambio* Éste es el conjunto de actividades que valoran el efecto y costo de los cambios. En la siguiente sección se estudia con más detalle este proceso.
3. *Políticas de seguimiento* Dichas políticas definen las relaciones entre cada requerimiento, así como entre los requerimientos y el diseño del sistema que debe registrarse. La política de seguimiento también tiene que definir cómo mantener dichos registros.
4. *Herramientas de apoyo* La administración de requerimientos incluye el procesamiento de grandes cantidades de información acerca de los requerimientos. Las herramientas disponibles varían desde sistemas especializados de administración de requerimientos, hasta hojas de cálculo y sistemas de bases de datos simples.

La administración de requerimientos necesita apoyo automatizado y herramientas de software, para lo cual deben seleccionarse durante la fase de planeación. Se necesitan herramientas de apoyo para:

1. *Almacenamiento de requerimientos* Los requerimientos tienen que mantenerse en un almacén de datos administrado y seguro, que sea accesible para todos quienes intervienen en el proceso de ingeniería de requerimientos.
2. *Administración del cambio* El proceso de administración del cambio se simplifica si está disponible la herramienta de apoyo activa.
3. *Administración del seguimiento* Como se estudió anteriormente, la herramienta de apoyo para el seguimiento permite la identificación de requerimientos relacionados. Algunas herramientas que están disponibles usan técnicas de procesamiento en lenguaje natural, para ayudar a descubrir posibles relaciones entre los requerimientos.

Administración del cambio en los requerimientos

La administración del cambio en los requerimientos debe aplicarse a todos los cambios propuestos a los requerimientos de un sistema, después de aprobarse el documento de requerimientos. La administración del cambio es esencial porque es necesario determinar si los beneficios de implementar nuevos requerimientos están justificados por los costos de la implementación. La ventaja de usar un proceso formal para la administración del cambio es que todas las propuestas de cambio se tratan de manera consistente y los cambios al documento de requerimientos se realizan en una forma controlada.

Existen tres etapas principales de un proceso de administración del cambio:

1. *Análisis del problema y especificación del cambio* El proceso comienza con la identificación de un problema en los requerimientos o, en ocasiones, con una propuesta de cambio específica. Durante esta etapa, el problema o la propuesta de cambio se analizan para comprobar que es válida. Este análisis retroalimenta al solicitante del cambio, quien responderá con una propuesta de cambio de requerimientos más específica, o decidirá retirar la petición.
2. *Análisis del cambio y estimación del costo* El efecto del cambio propuesto se valora usando información de seguimiento y conocimiento general de los requerimientos del sistema. El costo por realizar el cambio se estima en términos de modificaciones al documento de requerimientos y, si es adecuado, al diseño y la implementación del sistema. Una vez completado este análisis, se toma una decisión acerca de si se procede o no con el cambio de requerimientos.
3. *Implementación del cambio* Se modifican el documento de requerimientos y, donde sea necesario, el diseño y la implementación del sistema. Hay que organizar el documento de requerimientos de forma que sea posible realizar cambios sin reescritura o reorganización extensos. Conforme a los programas, la variabilidad en los documentos se logra al minimizar las referencias externas y al hacer las secciones del documento tan modulares como sea posible.