



## **Contenido Temático**

### **Tema 4: Ingeniería de Requerimiento**

#### **Educir requisitos**

El proceso de educir requisitos, según la Real Academia de la Lengua, significa sacar algo de algo. En este caso, lo que se educa son los requisitos de diversas fuentes, que más tarde se distinguirán. El proceso de educir requisitos puede descomponerse en las actividades de: hallazgo, de hechos, reunión e integración de la información.

Un procedimiento para obtener la información directamente de la gente que usará el sistema consta de los siguientes pasos:

- Identificación de las fuentes relevantes de los requisitos, es decir, que usuarios pueden proporcionar información acerca de los requisitos.
- Preguntar cuestiones apropiadas para obtener y comprender sus necesidades. Analizar la información reunida buscando implicaciones, inconsistencia o aspectos no resueltos.
- Confirmar la comprensión de los requisitos con los usuarios.
- Sintetizar las especificaciones adecuadas para los requisitos.

#### **¿Cómo se pueden construir las técnicas de educación?**

Las técnicas de educación han evolucionado desde este procedimiento general:

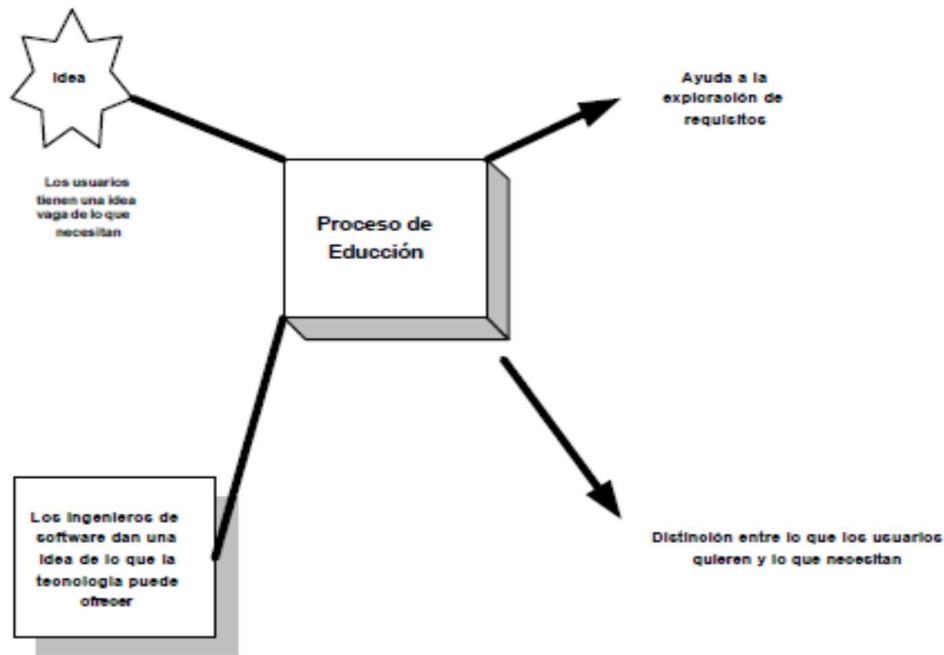
1. De finiendo procesos detallados
2. Proponiendo preguntas o categorías de preguntas
3. Estableciendo formatos estructurado para las reuniones
4. Describiendo comportamiento ideales, individuales o de grupo.
5. Construyendo patrones para organizar y registrar la información

El resultado tangible de educir requisitos es un conjunto de requisitos que puede ser utilizado por el equipo de desarrollo de software. Sin embargo, hay muchas conductas intangibles del proceso que pueden afectar al completo éxito del proyecto. Estas conductas difieren, dependiendo de si el proceso de educación fue conducido bien o pobremente.

#### **Conductas en un buen proceso**



El proceso de educir parte de una idea que refleja una necesidad de automatización que surge de los usuarios y que, gracias a la tecnología proporcionada por los ingenieros del software, se distingue con claridad lo que realmente necesitan los usuarios.



Figura~3.1: Entradas y salidas del proceso de educir

### Conductas en un proceso pobre

Cuando la educación es pobre, lo que ocurre con cierta frecuencia, la insatisfacción producida en los usuarios conduce a un grave impedimento para el desarrollo normal del proyecto software.

### Problemas de la educación de requisitos

Se pueden agrupar en tres categorías:

1. Problemas de alcance, de forma que los requisitos pueden implicar demasiada o muy poca información.

Las técnicas de educación tienen que ser lo suficientemente amplias como para establecer las condiciones de los límites del sistema objetivo.

La determinación de los límites y objetivos del sistema (con los que se comprobará que los requisitos obtenidos son correctos) se puede realizar a través de un análisis de contexto y de la organización.

Hay al menos tres contextos amplios que afectan al proceso de ingeniería de requisitos para un sistema propuesto:



- Organización: los requisitos de organización proporcionan una comprensión de la propia organización y del papel que representa el sistema en el contexto de la organización.
- Los factores de organización incluyen: las presentaciones de las entradas al sistema los usuarios de las salidas del sistema las formas en las que el sistema cambiará el modo de trabajar en la organización.
- Entorno: para educir requisitos del usuario es necesario disponer de una descripción precisa de los usuarios y de su entorno. Los factores de entorno incluyen: restricciones de hardware y software; madurez del dominio del sistema; certeza de las interface del sistema con el sistema que le embebe papel del sistema dentro de otro sistema más grande.
- Proyecto: los factores del contexto del proyecto incluyen: atributos de los diferentes grupos, tales como usuarios finales, patrocinadores, desarrolladores o analistas de requisitos.
  - Atributos tales como el estilo, la jerarquía en ese grupo, la experiencia en el dominio, la experiencia en computadoras.
  - restricciones impuestas por la gente implicada. Por ejemplo, la dirección puede imponer restricciones de tiempo, coste, calidad deseada.

2.- Problemas de comprensión, dentro de grupos de implicados o entre grupos, como por ejemplo, entre usuarios y desarrolladores. Como consecuencia de una pobre comunicación entre usuario y analista, los requisitos son ambiguos, incompletos, inconsistentes e incorrectos, porque no responden a las verdaderas necesidades de los usuarios.

Puede ser dividido en varios aspectos:

- Las comunidades implicadas en la educación (patrocinadores y clientes, desarrolladores, personas de control de calidad, analistas) tienen distintas formaciones y experiencia. La información reunida por una de las comunidades esta sesgado por el nivel de abstracción con el que este grupo ve el problema, objetivos, responsabilidades Por ejemplo:
  - Los desarrolladores del sistema tienen un conocimiento limitado del dominio de la aplicación
  - Los usuarios no conocen métodos de diseño para desarrollar sistemas con componentes software
  - Los clientes puede que no comprendan la dificultad para resolver un problema dado.
- El lenguaje usado para expresar requisitos puede ser demasiado formal o informal para reunir las necesidades de cada uno del grupo. Inicialmente, se usa lenguaje natural lo que provoca ambigüedad y mala interpretación.
- La gran cantidad de información que se educa tiene que organizarse de alguna forma. Cualquier metodología requiere una gran cantidad de documentación que hay que gestionar. La forma en la que se expresan los requisitos, y el tamaño del sistema descrito, afecta a la comprensión.



3.- Problema de volatilidad, ya que los requisitos evolucionan con el tiempo.

Las necesidades del usuario durante el desarrollo de un sistema pueden madurar a causa del conocimiento adicional fruto del desarrollo o de presiones del entorno o de la organización que no son previstas. Si no se incorporan estos cambios, los requisitos originales serán incompletos e inconsistentes con la nueva situación.

Algunas de las causas de volatilidad de los requisitos son:

1. Las necesidades del usuario evolucionan. El proceso de educación, análisis, especificación y validación se debe realizar más de una vez. Y las especificaciones obtenidas se van refinando de acuerdo al nuevo conocimiento obtenido.
2. Los requisitos son resultado de muchas contribuciones con distintas, y a veces contrapuestas, necesidades y objetivos. La priorización de las necesidades de las comunidades implicadas en la educación puede corregir la preponderancia de alguna de ellas durante esta fase de educación.
3. La complejidad de la organización. Los objetivos, políticas, funciones de los usuarios, pueden cambiar a lo largo del proyecto, sobre todo si los usuarios afectados son muchos. El proceso iterativo de desarrollo de requisitos resuelve este problema.
4. Las técnicas desarrolladas y usadas por los ingenieros de software han sido diseñadas para superar las dificultades descritas anteriormente (alcance, comprensión y volatilidad).

Se clasifican en dos grupos:

- Algunas técnicas son de alto nivel, es decir, amplios entornos que permiten albergar procesos que educen requisitos.
- Otras técnicas son de bajo nivel, que proporcionan tácticas específicas para educir detalles de una parte determinada de un sistema o de un usuario particular.
- 

En la tabla siguiente se aprecian algunas de las técnicas más utilizadas

Técnicas de Alto Nivel	Técnicas Detalladas
JAD	Entrevistas
Entorno de Bucles adaptativos	Cuestionarios
Prototipos	Arreglos-Q
	Baintorming
	PIECES
	Análisis de Mercado
	Análisis de factores Críticos
	STROBE

Tabla 3.1: Técnicas de educación de requisitos



En sentido genérico, también se pueden describir las técnicas en las siguientes categorías, basándose en la actividad del analista:

- Preguntar: Una vez identificada la persona se preguntan los requisitos
- Observar e inferir: Se observa el comportamiento de un sistema existente y a los usuarios del mismo y se infieren sus necesidades desde ese comportamiento.
- Discutir y formular: Se discute, con los usuarios, sus necesidades, y juntos, se formula una comprensión común de los requisitos.
- Negociar con respecto de un conjunto estándar: Se comienza con un conjunto existente de requisitos, y se negocia con los usuarios si cada una de estas características se incluye, excluye o modifica.
- Estudiar e identificar problemas: Se realizan investigaciones de problemas que identifiquen requisitos que mejoren el sistema.
- Descubrir a través de procesos creativos: Cuando no hay soluciones obvias, en problemas complejos
- Postular: Si no hay acceso al usuario o cliente, o si se está creando un producto que no tiene precedente.

## Técnicas de educación de requisitos

### El muestreo

El muestreo es el proceso por el cual se seleccionan de manera sistemática elementos representativos de una población. No es en sí mismo una técnica de educación, sino que a partir del muestreo se seleccionan, por ejemplo, las personas a quienes se les aplicará una estrategia de educación.

Por ejemplo si quisiera aplicar una encuesta para conocer el conocimiento sobre cierto rendimiento del sistema actual, podría aplicar una muestra para seleccionar a las personas que son mejores representantes para dicha estrategia de educación.

Existen numerosos motivos por los cuales un analista de sistema querrá seleccionar muestras representativas de datos para examinar o personajes que entrevistar, interrogar u observar. El muestreo agiliza el proceso, por medio de la recopilación de datos seleccionados, y no de todos los datos de la población.

El muestreo reduce la parcialidad de los datos recopilados. Se deben seguir pasos para lograr un buen diseño del muestreo.

1. Definir con precisión los datos que se van a recopilar o a describir. Se debe saber que se harán con los datos antes de recopilarlos. Se deben identificar las variables, atributos y los datos asociados a los artículos que serán recopilados en el muestreo. Se debe considerar el método de estudio (entrevista, cuestionario, etc.), el objetivo y las características de la información que se va a investigar.
2. Delimitar la población sujeta a selección de muestras. Se decide sobre las siguientes cuestiones: a) que tiempo serán suficientes tomar?; b) Se incluirá un solo nivel de la organización ?
3. Elegir el tipo de muestra. Existen cuatro tipos básicos de muestras:



- a. De oportunidad: son determinísticas y no tienen restricciones ni soporte probabilístico
- b. Dirigidas: se elige a un grupo de individuos que conozca y esté interesado en el nuevo sistema de información.
- c. Aleatorias simples: lista numerada de la población donde cada uno de los integrantes de la lista tienen la misma probabilidad de ser elegidos.
- d. Aleatorias complejas:
  - Sistemático: enésima persona de una lista
  - Estratificado: se identifican subpoblaciones (estratos), para luego seleccionar a sujetos de estas subpoblaciones.
  - Por grupos: selecciona a un grupo de gentes o documentos
  - Decidir el tamaño de la muestra. Debe ser un número absoluto (no un porcentaje).
  - Muchas veces se limita por el costo que involucra, o por el tiempo disponible del analista de sistemas, o de los integrantes de la organización.

Pasos a seguir para determinar el tamaño de muestra:

1. Establecer el atributo que se va a tomar como muestra
2. Localizar la base de datos o el informe donde se encuentran los atributos
3. Examinar el atributo y estimar "p" (proporción de la población) que cuente con el atributo
4. Determinar el intervalo estimado aceptable "i" .
5. Elegir el nivel de confianza y encontrar el coeficiente de confiabilidad en un tabla
6. Calcular el error estándar de la muestra "Vp",  $vp = i/z$  .-
7. Determinar el tamaño de la muestra, "n", mediante:  $n = ( p( 1 - p) / Vp^2 ) + 1$

### **Tipos de información que se obtiene durante la investigación**

Conforme el analista se va adentrando en la organización y en sus requerimientos de información, se torna importante examinar los diferentes tipos de datos que aportan información, que de otra manera no podría obtenerse. Los datos concretos revelan la trayectoria de la organización y hacia dónde se dirige según sus miembros. El analista debe examinar los datos cualitativos y cuantitativos.

#### **Análisis de datos cuantitativos**

- Informes corporativos. La ley exige por mandato y regulación que las firmas proporcionen informes corporativos o informes anuales para la emisión de acciones públicas. El estado de resultado es otra forma de estos informes.
- Informes de desempeño. Comparan los resultados reales con los planeados. Sirve para determinar si la tendencia de la diferencia se acorta o se amplía.
- Registros. Contienen actualizaciones periódicas de lo que ocurre en la empresa.



- Formas para la captura de datos. Antes de que se disponga a modificar el flujo de información dentro de la organización, debe comprender la operación del sistema vigente, para ello recopilara y catalogara copias en blanco de cada una de las formas (oficiales y extraoficiales) que se utilizan.

Análisis de documento cualitativo.

Estos documentos no siguen lineamientos preestablecidos y su análisis se vuelve fundamental para comprender como los integrantes de la organización están involucrados en el proceso de la organización. Incluyen:

- Memorándum. Se debe considerar quienes son los emisores y quienes los reciben. Normalmente fluyen hacia abajo. El análisis de su contenido revela valores, actitudes y creencias de los miembros de la organización.
- Avisos pegados en tableros y áreas de trabajo. Reflejan la cultura oficial de la organización.
- Manuales de procedimientos y de políticas de la empresa. Deben analizarse siguiendo lineamientos preestablecidos.

Gran parte de la información, tanto cuantitativa como cualitativa, no es de uso corriente; más bien, se encuentran almacenadas en archiveros.

## **Técnicas de educación**

### **La entrevista**

Es una conversación dirigida con un propósito específico, que se basa en un formato de preguntas y respuestas. En la entrevista desea conocer tanto las opiniones como los sentimientos del entrevistado acerca del estado actual de los sistemas, sus metas personales, de la organización y de los procedimientos informales. En la entrevista necesitara crear un clima de confianza y entendimiento, sin perder en control de la misma.

### **Planeación de la entrevista**

Son cinco los pasos para la planeación de la entrevista.

1. Lectura de antecedentes. Consulte y comprenda el mayor número posible de antecedentes de los entrevistados y de la organización, esto le ayudara a elaborar un vocabulario, que eventualmente, lo capacitará para redactar las preguntas de la entrevista, de manera tal que estas sean comprendidas por su entrevistado y a aprovechar al máximo el tiempo de la entrevista no haciendo preguntas generales sobre antecedentes de la organización.
2. Establecimiento de los objetivos de la entrevista. Se establecen con base a los antecedentes que consulte y a su experiencia particular. Deben existir aspectos fundamentales referentes al procesamiento de la información y al proceso de la toma de decisiones sobre las cuales se deseara hacer preguntas, están incluyen:





fuentes de información, formatos de la información, la frecuencia de la toma de decisiones, la calidad de la información y el estilo de la toma de decisiones.

3. Selección de los entrevistados. Se deben incluir a gente clave en cada nivel.
4. Preparación del entrevistado. Prepare a las personas que entrevistara. El tiempo que ellos le dedican a Ud. no se lo dedican a la Organización.
5. Selección del tipo y estructura de las preguntas. La esencia de la entrevista se basa en uso de técnicas inquisitivas adecuadas. Las preguntas tienen cierto tipo de estructura. Los dos tipos básicos son cerradas y abiertas. Es posible estructurar su entrevista bajo tres patrones diferentes: el de estructura de pirámide, de estructura de embudo y la estructura de diamante. Cada uno de ellos adecuado para condiciones particulares.

### **Preguntas abiertas**

Incluyen a aquellas tales como:

¿Que opina acerca del uso de redes informáticas en las pequeñas organizaciones? y  
¿Podría explicar el uso de prototipos?

Abiertas son las opciones que tiene el entrevistado para responder. Algunas de las ventajas de utilizar preguntas abiertas son: simplifican las cosas para el entrevistado; permiten al entrevistador, seleccionar el vocabulario del entrevistado, lo que refleja su educación, valores y creencias.; proporcionan una riqueza de detalla; revelan nuevas alternativas sobre preguntas no consideradas; hace más interesante la entrevista, permite una mayor espontaneidad; facilita el estilo del entrevistado; se suele utilizar como alternativa cuando el entrevistado no se encuentra preparado.

Algunas de las desventajas de las preguntas abiertas son: permiten preguntas que pueden generar demasiada información irrelevante; la posible pérdida del control de la entrevista; respuesta con demasiado tiempo para la cantidad de información que aportan; pueden dar la apariencia de que el entrevistador no se prepara o que el mismo se encuentra expedicionando sin tener claros los objetivos.

### **Preguntas cerradas**

Podría ser ¿Cuántos especialistas conforman su equipo de trabajo ?. Las posibles respuestas se hallan limitadas. Un tipo especial de pregunta cerrada es la bipolar. Es muy limitada al tener solo dos respuestas alternativas. Algunas ventajas de las preguntas cerradas son: ahorran tiempo; facilitan la comparación entre entrevistas; llegan al punto de interés; mantienen el control de la entrevista; cubren rápidamente diversos aspectos; obtienen datos de relevancia. Sin embargo las desventajas de la utilización de preguntas cerradas son importantes, entre otras, se citan las siguientes: aburren al entrevistado; pierden la riqueza del detalle; se pueden perder ideas centrales por el punto anterior y no favorecen un clima de armonía entre el entrevistado y el entrevistador.

### **Sondeos**

Es un tercer tipo de pregunta. La pregunta más simple es el ¿Por qué ?El propósito del sondeo es ir más allá de la respuesta inicial para obtener un mayor significado, y aclarar





o ampliar los puntos del entrevistado. El sondeo se puede realizar mediante preguntas cerradas o abiertas.

La mayoría de los entrevistadores principiantes son reticentes acerca del sondeo, y en consecuencia, aceptan respuestas superficiales. Si el sondeo se hace de una manera sistemática y determinada, le servirá para mostrar a su interlocutor que escucha lo que dice, pensando al respecto y respondiendo adecuadamente.

### **Errores en las preguntas**

Al formular de antemano las preguntas podemos corregirlas. Dentro de los tipos de preguntas deficientes tenemos las tendenciosas y las preguntas dobles. Las preguntas tendenciosas tienden a dirigir al entrevistado hacia la respuesta que Ud. quisiera escuchar.

Las preguntas dobles son aquellas que en una sola contienen, de hecho, dos preguntas diferentes. Normalmente no deben utilizarse por que se corre el peligro que el entrevistado solo conteste una de ellas.

### **Orden de las preguntas en una secuencia lógica.**

Existen tres formas de organizar sus preguntas:

- inductivo (piramidal)
- deductivo (embudo)
- combinación de ambas (diamante)

Muchos entrevistadores consideran que las entrevistas se asemejan a conversaciones, y por lo tanto, deberían de carecer de una estructura formal, tanto las preguntas como su secuencia. Para una entrevista íntegramente estructurada todo se planea de antemano y tal plan se sigue de manera estricta... Las preguntas cerradas son la esencia de una entrevista completamente estructurada.

### **Registro de la Entrevista**

Registre los aspectos más importantes de su entrevista. Puede usar grabadora, cuaderno de notas, etc. , lo importante es que se lleve un registro permanente durante la entrevista. De avisar al entrevistado del medio de registración que utilizara, como también el objetivo de dicha registración. Sea honesto y explícito en sus declaraciones y garantice la confiabilidad de la registración. La grabación posee algunas ventajas: proporciona un registro preciso y completo; libera al entrevistador para escuchar y responder con mayor rapidez; permite reproducir la entrevista, permite un mayor contacto visual, posee las siguientes desventajas: puede poner nervioso al entrevistado; puede reducir la capacidad de atención del entrevistador; inconvenientes de búsquedas en la cinta de pasajes de la entrevista.

El tomar notas también tiene sus ventajas: mantiene alerta al entrevistador; sirve para recordar preguntas importantes; muestra el interés del entrevistador en la entrevista y de su preparación para la misma: También las notas no dejan de tener inconvenientes:



perdida de contacto visual; perdida de la continuidad en la conversación, interrupciones, etc...

Antes de la entrevista comuníquese con su entrevistado para confirmar el lugar y la hora. Antes de la entrevistas todo tenga en perfectas condiciones y listo.

#### Realización de la entrevista

Comience en el momento establecido. Saludo a su entrevistado. Veri...que que todo este en perfectas condiciones. Recuerde a su interlocutor el grado de detalles que quiere en las respuestas (si fuesen abiertas). Controle el uso del tiempo. Manéjelo adecuadamente. Existen algunos problemas que se pueden presentar durante la entrevista, por ejemplo: el entrevistado percibe que su autoestima se encuentra amenazada; puede llegar a tener reacciones emotivas a temas conflictivos, puede llegar a malentender las preguntas respecto a la sucesión de los acontecimientos; puede apegarse a formas tradicionales, puede llegar a equivocarse en el momento de inferir sobre lo observado; puede llegar a competir con Ud. Sobre el tiempo de la entrevista. ; puede olvidarse e inclusive mentir sobre hechos importantes.

Trate de cubrir todo el material de la entrevista en no más de 45 minutos (o una hora). Concluya su entrevista en el momento prometido. Solicite al entrevistado si desea acotar algún otro detalle al finalizar la misma. Informe al entrevistado sobre los pasos posteriores.

Luego necesita redactar un informe escrito de la entrevista que rescate la esencia de la misma. De esta manera se asegura la calidad de los datos.

#### Cuestionarios

El Cuestionario es un instrumento de investigación. Este instrumento se utiliza, de un modo preferente, en el desarrollo de búsqueda de información en la etapa de educación, es decir, como estrategia de reducción de requisitos. Es una técnica ampliamente aplicada en el desarrollo de software, y muchas veces, es la única probable a utilizar.

No obstante lo anterior, su construcción, aplicación y tabulación poseen un alto grado científico y objetivo. Elaborar un Cuestionario válido no es una cuestión fácil; implica controlar una serie de variables. Por ello es necesario que el Ingeniero de requerimiento tenga una acaba información al respecto de su creación, uso, valorización y obtención de resultado.

El Cuestionario es "un medio útil y eficaz para recoger información en un tiempo relativamente breve".

En su construcción pueden considerarse preguntas cerradas, abiertas o mixtas.

Recopila información que permite a los analistas recoger opiniones, posturas, conductas y características de las diversas personas de una organización.



Las respuestas (con preguntas cerradas) puede cuantificarse las de preguntas abiertas se pueden analizar e interpretar de maneras distintas.

Sirven para determinar qué tan difundido o limitado se encuentra un sentimiento.

Forma rápida de recopilar cantidades masivas de datos acerca de la opinión de los usuarios.

Lineamientos:

1. Las personas a quienes se interrogara se encuentran dispersas
2. Los involucrados en el proyecto son muchos y se necesita conocer un porcentaje de aprobación o no del mismo.
3. Se lleva a cabo un estudio exploratorio y deseamos medir la opinión general.
4. Deseamos sondear el problema que presenta el sistema actual.

## CARACTERÍSTICAS

- 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 responda 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.

## Cuestionario Restringido o Cerrado

- Es aquel que solicita respuestas breves, específicas y delimitadas.
- "Para poder formular preguntas cerradas es necesario anticipar las posibles alternativas de respuestas".
- Estas respuestas piden ser contestadas con:
  - Dos alternativas de respuestas (respuestas dicotómicas): Si o No.
  - Varias alternativas de respuestas: donde se señala uno o más ítems (opción o categoría) en una lista de respuestas sugeridas. Como no es posible prever todas las posibles respuestas, conviene agregar la categoría Otros o Ninguna



de las Anteriores, según sea el caso. En otras ocasiones, el encuestado tiene que jerarquizar opciones o asignar un puntaje a una o diversas cuestiones.

- Ventajas:
  - Requiere de un menor esfuerzo por parte de los encuestados.
  - Limitan las respuestas de la muestra.
  - Es fácil de llenar.
  - Mantiene al sujeto en el tema.
  - Es relativamente objetivo.
  - Es fácil de clasificar y analizar.

### Cuestionario No Restringido o Abierto

- Las preguntas abiertas no delimitan de antemano las alternativas de respuesta.
- "Las preguntas abiertas son particularmente útiles cuando no tenemos información sobre las posibles respuestas de las personas o cuando esta información es insuficiente".
- Es aquel que solicita una respuesta libre.
- Esta respuesta es redactada por el propio sujeto.
- Proporciona respuestas de mayor profundidad.
- Es de difícil tabulación, resumen e interpretación.

### Cuestionario Mixto

- Es aquél que considera en su construcción tanto preguntas cerradas como abiertas.

### Requerimientos para la Construcción de 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 sólo 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").
- "La elección de tipo de preguntas que contenga el cuestionario depende del grado en que se puedan anticipar las posibles respuestas, los tiempos de que se disponga para codificar y si se quiere una respuesta más precisa o profundizar en alguna cuestión".

### **Construcción de cuestionarios**

- Busca solamente la información que se puede obtener de otras fuentes.
- Es tan breve como sea posible y sólo lo bastante extenso para obtener los datos esenciales.
- Tiene un aspecto atractivo.
- Las instrucciones son claras y completas. Los términos importantes se hallan definidos; cada pregunta implica una sola idea; todas ellas están expresadas tan sencilla y claramente como sea posible, de manera que permite respuestas fáciles, exactas y sin ambigüedad.
- La importancia del tema al cual se refiere, debe ser expuesta clara y cuidadosamente en el cuestionario. Las personas estarán más dispuestas a responder si saben cómo serán utilizadas sus respuestas.
- Las preguntas son objetivas, sin sugerencias hacia lo que se desea como respuesta.
- Las preguntas están presentadas en un orden psicológico correcto, precediendo las de tipo general a las específicas. Deben evitarse las preguntas molestas.
- Es fácil de clasificar o interpretar.
- Antes de aplicar un cuestionario a un grupo numeroso, conviene experimentarlo en un grupo reducido de características lo más semejantes a las personas a las que se va a encuestar. Esta aplicación previa tiene por objeto detectar preguntas e instrucciones ambiguas que posteriormente



pueden restar validez al instrumento. Es lo que se denomina cuestionario piloto de la prueba.

- Al elaborar el cuestionario es necesario establecer la forma en que será tabulado e interpretado. Para este objeto, es de gran utilidad la aplicación experimental que permite prever la dispersión que tendrán las respuestas. Una de las formas más sencillas de tabular un cuestionario es construir una tabla de doble entrada, en uno de cuyos ejes se registra a los encuestados o el número de formulario si se aplicó en forma anónima, y en su otro eje se colocan las preguntas o el número que las representa. De este modo es posible obtener rápidamente una visión global de las respuestas dadas por los individuos encuestados.

### Uso de escalas en los cuestionarios

Escala es el proceso de asignar números u otros símbolos a un atributo o características con el fin de poder medirlo.

Con frecuencias las escalas son arbitrarias y no llegan a ser únicas.

Formas de escalas de medición:

- Nominal: se utilizan para clasificar objetos.
- Ordinal: También permiten la clasificación, sin embargo, la diferencia es que la escala ordinal implica además un arreglo por categorías.
- De intervalo: tienen la característica que la diferencia que existe es la misma entre los intervalos de cada uno de los números.

Proporcional: Son similares a las de intervalos porque se supone que los intervalos entre los números son iguales: sin embargo, las escalas proporcionales cuentan con un cero absoluto.

### La Escala de Likert

- Es un tipo de instrumento de medición o de recolección de datos que disponemos en la investigación social.
- Es una escala para medir las actitudes.
- Consiste en un conjunto de ítems bajo la forma de afirmaciones o juicios ante los cuales se solicita la reacción (favorable o desfavorable, positiva o negativa) de los individuos.
- Alternativas o puntos en la escala de Likert:

Alternativa A: (5) Muy de acuerdo; (4) De acuerdo; (3) Ni de acuerdo, ni en desacuerdo (Afirmación); (2) En desacuerdo; (1) Muy en desacuerdo

Alternativa B: (5) Totalmente de acuerdo; (4) De acuerdo; (3) Neutral (Afirmación); (2) En desacuerdo; (1) Totalmente en desacuerdo



Alternativa C: (5) Definitivamente sí; (4) Probablemente sí; (3) Indeciso (Afirmación); (2) Probablemente no; (1) Definitivamente no;

Alternativa D: (5) Completamente verdadero; (4) Verdadero; (3) Ni falso, ni verdadero (Afirmación); (2) Falso; (1) Completamente falso

- Para obtener las puntuaciones de la escala de Likert, se suman los valores obtenidos respecto de cada frase. El puntaje mínimo resulta de la multiplicación del número de ítems por 1. Una puntuación se considera alta o baja respecto del puntaje total (PT); este último está dado por el número de ítems o afirmaciones multiplicado por 5.

### Parámetros de Desempeño

La validez es el grado con el que la pregunta determina lo que el analista intenta medir. La confiabilidad es un parámetro de consistencia. Si el cuestionario da el mismo resultado al repetirlo bajo las mismas condiciones, se dice que el instrumento cuenta con una consistencia externa. Si el cuestionario contiene fragmentos, y estos generan resultados equivalentes, se dice que tiene una consistencia interna. Ambas son importantes.



### Elaboración de Escalas

Escalas arbitrarias: mide los que se intenta medir

Escala por consenso: involucra a un grupo de jueces, quienes ayudan a evaluar los tópicos que el analista eligió para el cuestionario.

Análisis de temas: involucra la evaluación de los tópicos por un pequeño grupo de quienes responderán, similar al grupo que recibirá el cuestionario definitivo. Cada tópico se analiza en función de su conveniencia.

Factorización: es el procedimiento estadístico por medio del cual se agrupan objetos similares.

Las escalas mal echas origina problemas de:

- Indulgencia: los que responden los cuestionarios son pocos evaluadores





- Tendencia central: cuando el que responde califica todo como promedio.
- Efecto halo: la impresión que deja una pregunta se acarrea en la siguiente.

### Diseño y aplicación de cuestionarios

Elimina cierta resistencia para responder.

#### Lineamientos

- Formato: disponga de suficiente espacio en blanco, disponga de suficiente espacio para las respuestas, utilice círculos para las respuestas, utilice los objetivos como ayuda para establecer el formato y mantenga un estilo consistente.
- Orden de las preguntas: No existe una forma única pero deberá prever que las preguntas de importancia para quien contesta van primero, se deben agrupar las preguntas del mismo tema, use las tendencias asociativas y no olvide de plantear primero los temas de menor controversia.



La decisión acerca de a quien se dará el cuestionario se toma al momento de establecer los objetivos de la información que se busca.

1. Reunir a todas las personas en un solo sitio
2. Entregar personalmente los cuestionarios en blanco y recogerlos luego.
3. Enviar por correo el cuestionario a aquellos empleados de sucursales remotas, estableciendo fechas límites, proporcionando instrucciones y reembolso postal.

### Uso de arreglos Q

Esta metodología consiste en la estructuración de un arreglo-Q, el cual fuerza a que las respuestas se apeguen a una distribución normal, que es adecuada para agrupar a los que responden, con base en sus opiniones sobre un tópico particular.

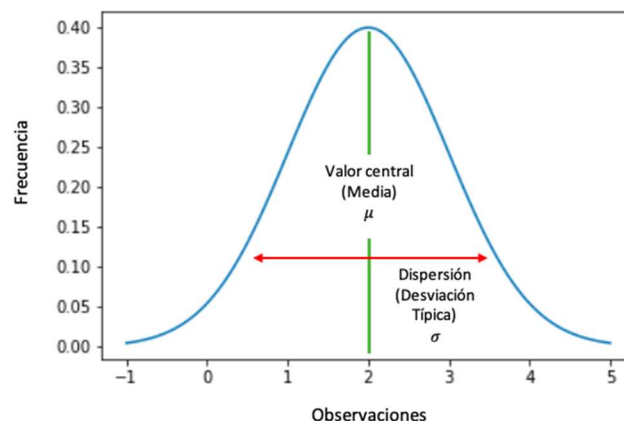
A partir de las entrevistas, las observaciones y el muestreo de datos de archivos, el analista crea una serie de tarjetas con postulados o conceptos para que los que responden las ordenen.



Cada tarjeta contiene por separado un postulado; en teoría, al agrupar las tarjetas, se tiene todo el dominio posible de actitudes sobre el tópic en cuestión.

Se instruye a quien responde para tomar un paquete de tarjetas y separarlas en varias pilas. Para facilitar lo anterior, el analista prepara entre ocho y diez tableros de arreglo-Q, de tal forma que la selección de varios individuos puede procesarse simultáneamente. El número de pilas que se van a ordenar depende del número de postulados; cuanto mayor sea el número de postulados, más pilas debe haber.

Se le pide a quien responde que coloque un número específico de tarjetas en cada pila. El número de tarjetas requerido en cada pila tiene que corresponder a un perfil discreto de distribución normal. (Ver Fig.)



En la gráfica de la distribución “normal” se aprecia la imagen que deberá tener nuestra ordenación de las pilas de respuestas. Los que están por la negativa, los que tienen una respuesta promedio, y finalmente los que están por la afirmativa. Nos damos cuenta que cualquier interrogante que tengamos para responder, podremos pre clasificar las respuestas con la presente distribución normal.

### Brainstorming

Es la aportación de ideas que varias personas ponen en común como punto de partida para un proyecto. Es una técnica de grupo sencilla para la generación de ideas.

Una sesión de brainstorming funciona bien con 4 a 10 personas. Una de ellas es el líder, cuyo papel se reduce a, prácticamente, abrir la reunión más que a restringirla o controlarla.

Es útil en:

- ☐ La generación de múltiples puntos de vistas del problema
- ☐ La formulación del problema en forma diferentes

Ayuda en:

- ☐ Estimula la imaginación contribuyendo a que los usuarios sean conscientes de sus necesidades
- ☐ Ayuda a construir una más completa figura de las necesidades de usuario
- ☐ Evita la tendencia de centrarse en áreas estrechas demasiado pronto



- ☐ Es una técnica fácil de aprender

A continuación se presenta las fases de una sesión. En ella se deberá llevar a cabo distintas fases:

**Fase de preparación:** La preparación para una sesión de brainstorming requiere:

- ☐ La identificación de participantes: clientes, usuarios, e ingenieros de software
- ☐ La designación del líder.
- ☐ La planificación de la sesión con los participantes
- ☐ La preparación de la sala

**Fase de generación:** La sesión es iniciada por el líder, que expresa una instrucción general del problema.

Los participantes expresan libremente, en turno o espontáneamente, dependiendo del criterio del líder.

No hay críticas sobre las ideas, se fomentan las ideas no convencionales, así como embellecer o combinar las ideas de otros. Las ideas deben estar visibles durante la sesión.

**Fase de consolidación:** Durante esta fase las ideas se organizan según criterios de uso, en cuatro pasos:

- 1.- Se revisan las ideas para clasificarlas.
- 2.- Se descartan las ideas que no sean utilizables
- 3.- Se discuten las ideas restantes con el objetivo de ordenarlas imponiendo prioridades.
- 4.- se registran todas estas ideas, con sus prioridades u otros comentarios.

## Entorno PIECES

Es otra estrategia que su uso está poco experimentado por el grupo de analistas. Consta de seis categorías de aspectos que se deben explorar, y que pueden ser ajustados según el dominio de la aplicación.

PIECES es un acrónimo formado por las iniciales de las seis categorías que propone: Rendimiento, información y datos, economía, control, eficiencia y servicios.

**Rendimiento del sistema (Throughput,** o número de tareas completadas en unidad de tiempo.

**Tiempo de respuesta,** o tiempo requerido para ejecutar una única tarea).

**Información y datos a obtener del sistema.** Datos útiles para la toma de decisión

Acceso a la clase correcta de información, en el momento adecuado, en la cantidad justa y en forma utilizable.

**Economía.** Factores de coste relacionados con el uso de un sistema (Nivel de servicio, Exceso de capacidad, que puede implicar procesadores adicionales, redes, estructuras de datos internos, ... que mantengan un servicio estable de rendimiento, cuando la demanda



suele venir a picos. Carga esperada sobre el sistema y el nivel de servicio apropiado se podrá equilibrar el nivel de servicio y el exceso de capacidad.

**Control.** Tipos de control: Seguridad, y Auditoria.

**Eficiencia.** Es una medida de recursos dedicados a realizar una tarea.

**Servicios proporcionados por el Sistema.** Para los usuarios, para los clientes y para otros sistemas.

### **Análisis de mercado**

El análisis de mercado es una actividad común de la mayoría de empresas que venden productos. Las empresas necesitan conocer el mercado antes de construir el producto.

La pregunta es: ¿quiénes hacen el análisis de mercado?

En las grandes empresas, los especialistas de análisis de mercado. En las pequeñas, se contratan consultores.

Hay diversos aspectos en el análisis de mercado:

- ☐ El *análisis de mercado* observa cuidadosamente los productos similares de los competidores. Se identifica lo bueno a copiar y lo malo a evitar.
- ☐ La *investigación de mercado* implica recoger datos estadísticos sobre los productos comprados y se identifican las tendencias en esos datos para predecir la necesidad de futuros productos.
- ☐ *Cuestionarios de clientes*, diseñados cuidadosamente por expertos, pueden educir información muy detallada de las necesidades de compradores y usuarios potenciales de un paquete de software.

### **Análisis de factores críticos**

Esta aproximación de educación consiste en identificar y concentrarse en un pequeño conjunto de factores críticos de los que depende la efectividad del sistema.

Pasos:

- ☐ Comprender la operación del sistema
- ☐ Identificar factores críticos para que el sistema sea efectivo
- ☐ Identificar la fuerza y la debilidad del sistema respecto de cada uno de estos factores
- ☐ Identificar áreas de problemas y oportunidades
- ☐ Reunir los detalles relevantes de estos factores que mejoren el rendimiento del sistema. Para educir detalles se puede usar un proceso estructurado orientado a objetivos
- ☐ Formular requisitos usando estos detalles

### **JAD - Joint Application Design**

El método fue desarrollado para el diseño de sistemas basados en computadoras, centrándose principalmente en el trabajo conjunto entre el usuario demandante del



servicio y el analista en sistema, es decir, la persona que dará la solución a un problema planteado, con la finalidad de que todos los involucrados participen en el desarrollo del trabajo a ejecutar. Este método permite medir, la evolución de las fases del proyecto a ejecutar, es decir, ayudan a tener una mejor comprensión del proceso a seguir, e ir mejorando el mismo durante su ejecución, permitiendo así conocer la complejidad del mismo, para así poder ir tomando las decisiones más exactas se ir reduciendo los márgenes de errores que puedan ir presentándose durante la elaboración del sistema a implantar, y por ende, ayuda a los actores del proceso, saber si están alcanzando los objetivos propuestos, incrementando la satisfacción del cliente.

Promueve la cooperación, comprensión, y equipo de trabajo entre compradores, usuarios y desarrolladores. Proporciona un proceso que facilita la creación de una visión compartida de lo que el sistema debería ser.

### Ventajas

Permite a los usuarios, diseñar sistemas en forma conjunta, en sesiones grupales. Promueve la cooperación, el entendimiento y el trabajo grupal entre distintos grupos de usuarios. Ahorra tiempo en el análisis. Prioriza objetivos. Amplia la transparencia en los procesos. Acelera la toma de decisiones. Mejora la definición de los requerimientos. Comunicar requerimientos funcionales. Estimar esfuerzo, agenda y costos basado en requerimientos. Evaluar la factibilidad de un proyecto. Administrar los cambios. Mejora el mantenimiento y soporte. Medir la productividad. Verificar la completitud. Es independiente de la tecnología usada para el desarrollo e implementación.

### Desventajas

Es más costoso. Puede ser incómodo si el grupo es demasiado grande concerniente al tamaño del proyecto. Resulta arduo formar al personal en su utilización, y mantener las reuniones.

Carece de precisión cuando se trata de proyectos pequeños. Para resultar útil una organización de desarrollo y mantenimiento de software debe tener recontada la mayor parte de su base instalada, pero hacerlo resulta muy costoso, especialmente si se mantiene software de terceros. El factor de ajuste calculado a partir de las características generales de sistema resulta de dudosa utilidad.

### *Principios sobre los que se apoya el JAD*

- ☐ Dinámica de grupo, en la que se enriquecen, complementan e integran las visiones parciales que tienen los participantes en el proceso de analizar el problema.
- ☐ Uso, en las reuniones, de ayudas visuales para mejorar la comunicación y la comprensión
- ☐ Mantenimiento de un proceso organizado racional, en el que los papeles de cada uno de los participantes están perfectamente definidos.



- ❑ Filosofía de documentación "lo que ves es lo que obtienes" con formularios de documentos estándares que se rellenan y se endosan por todos los participantes en una sesión.

Se distingue cinco fases en un plan JAD:

- ❑ *Definición del proyecto*
- ❑ *Investigación*
- ❑ *Preparación*
- ❑ *La sesión JAD*
- ❑ *El documento final*

### Entorno de bucles adaptativos

Es una estrategia donde interactúa durante toda la etapa de ingeniería en requisitos el usuario, el desarrollador y lo que se va realizando y comprendiendo del sistema.

La figura siguiente refiere a la interacción de los actores en la presente estrategia de educación.

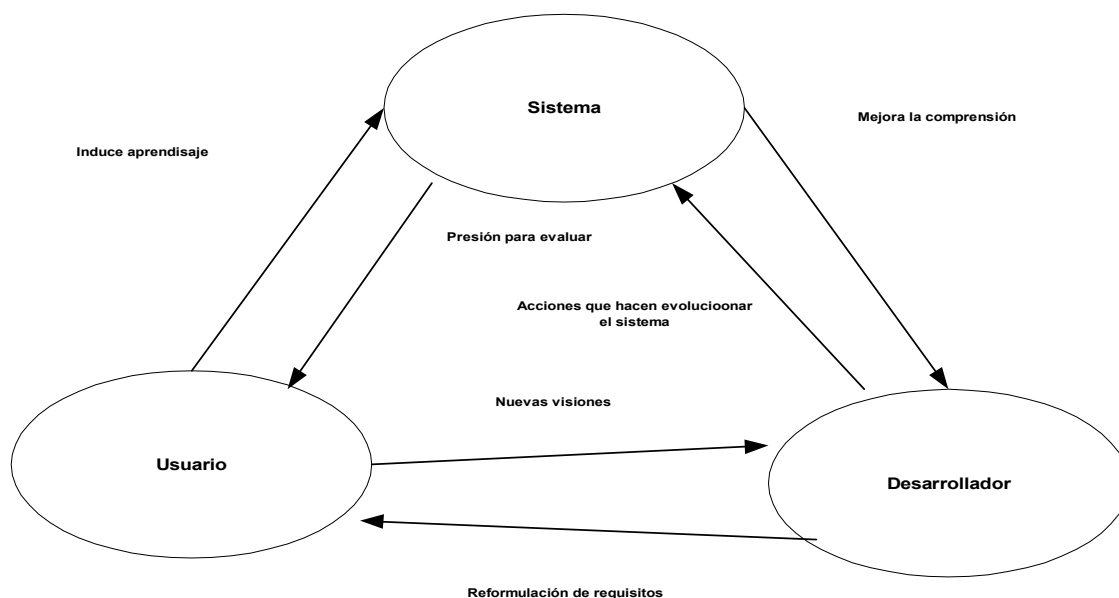


Fig. 03-02: Ciclos de aprendizaje en el entorno de bucles adaptativos

### Prototipos

Consiste en capturar un conjunto de necesidades e implementarlas rápidamente con la intención declarada de expandirlas y refinarlas iterativamente al ir aumentando la comprensión que del sistema tienen los usuarios y quien lo desarrolla. La definición del sistema se realiza mediante el descubrimiento evolutivo y gradual.



El desarrollo de prototipos es una metodología valiosa para identificar con rapidez, las necesidades particulares de información del usuario.

### Enfoques para el desarrollo de los prototipos

- Prototipos de remiendo. Tiene que ver con la construcción de un sistema que si bien funciona, se encuentra remendado o parchado.
- Modelo a escala no funcional. Se construyen a escala, con el objeto de evaluar ciertos aspectos de diseño. (Ej.: Modelo de avión).
- Primer modelo a escala completa. Comúnmente llamado piloto. (Ej.: Primer automóvil de una serie).
- Características esenciales. Incluye algunas, pero no todas las características que tendrá el sistema final.

### Desarrollo de Prototipos

- ☐ Trabajar con módulos manipulables. Un módulo manipulable es aquel que nos permite relacionarnos con sus características, y además su construcción es independiente de otros módulos del sistema.
- ☐ Construir el prototipo con rapidez. Una vez realizado un breve análisis sobre los requerimientos de información por medio de métodos tradicionales (entrevista, cuestionarios, observación), se construye los módulos funcionales de prototipos.
- ☐ Modificaciones en el prototipo. Los módulos deben tener baja dependencia para que sean fácilmente modificables.
- ☐ Enfatizar la interfaz con el usuario. Lo que se trata de obtener es que el usuario planteen más allá sus requerimientos de información.

### Desventaja

Su administración llega a ser difícil como un proyecto.  
Se lo considera un sistema, tanto por los usuarios como los analistas.

### Ventajas

Modificación del sistema en etapas tempranas de su desarrollo

Oportunidad de detener el desarrollo de un sistema que no sirve y la posibilidad de desarrollar otro sistema que se ajuste mejor a las necesidades y a las expectativas del usuario.

El uso de prototipos se resume en: "**honestidad y compromiso**". Si se carece del compromiso del usuario, pocos son los motivos para desarrollar un prototipo. La conducta del usuario es el pivote del proceso de desarrollo y evaluación del prototipo.

### Formas de apoyar la evaluación del prototipo

- ☐ Experimentar con el prototipo





- ☐ Plantear sin restricciones, reacciones hacia el prototipo
- ☐ Sugerir mejorar o recortes al prototipo

### Despliegue de la Función de Calidad (DFC)

Es una técnica de gestión de calidad que traduce las necesidades del cliente en requisitos técnicos del software. Desarrollada en Japón y aplicada en la Industria Mitsubishi.

DFC hace énfasis en entender lo que resulta importante para el cliente. Identifica tres tipos de requisitos:

- Requisitos normales. Se declaran los objetivos y metas en una reunión con el cliente. Si estos requisitos están presentes el cliente está satisfecho.
- Requisitos esperados. Estos son implícitos al sistema, y el cliente no siempre los declara.
- Requisitos innovadores. Son características que van más allá de las expectativas del cliente y suelen ser muy satisfactorias.

### Escenarios

El escenario comienza con un esbozo de la interacción y, durante la obtención, se agregan detalles para crear una descripción completa de esta interacción.

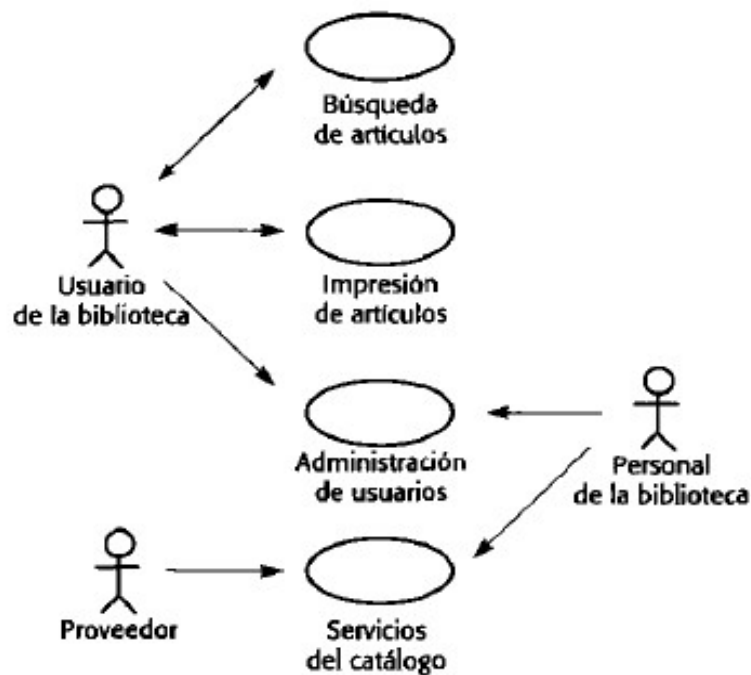
De forma general, un escenario puede incluir:

- Una descripción de lo que esperan el sistema y los usuarios cuando el escenario comienza.
- Una descripción del flujo normal de eventos en el escenario.
- Una descripción de lo que puede ir mal y cómo manejarlo.
- Información de otras actividades que se podrían llevar a cabo al mismo tiempo.
- Una descripción del estado del sistema cuando el escenario termina.

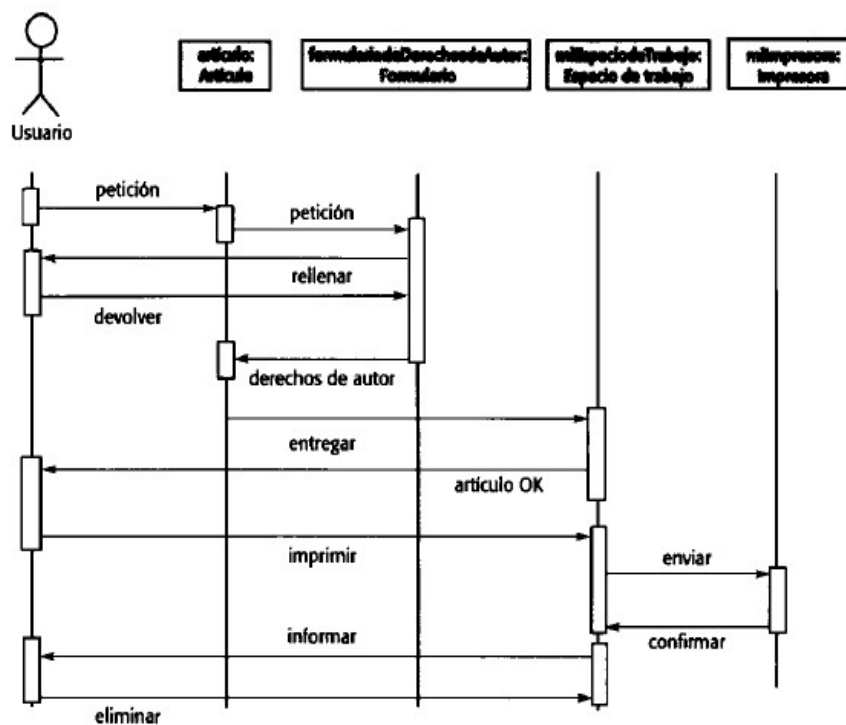
### Casos de uso

Los casos de uso son una técnica que se basa en escenarios para la obtención de requerimientos que se introdujeron por primera vez en el método Objeto (Jacobsen *et al.*, 1993).

Actualmente se han convertido en una característica fundamental de la notación de UML, que se utiliza para describir modelos de sistemas orientados a objetos.



En su forma más simple, un caso de uso identifica el tipo de interacción y los actores involucrados.



Observación al comportamiento de la toma de decisiones y al ambiente de oficina (STROBE)



Se hace uso de la observación para obtener información de los tomadores de decisión y su ambiente, también auxilia a confirmar lo que las entrevistas y los cuestionarios hubieran detectado, o para rebatir o anular lo que otros métodos encontraron.

- ☐ Observación de las actividades de quien toma decisiones
  - ☐ Observación del lenguaje corporal
  - ☐ Registro del comportamiento
- ☐ Observación del ambiente físico

### Etnografía

Es especialmente efectiva para descubrir dos tipos de requerimientos:

- *Los requerimientos que se derivan de la forma en la que la gente trabaja realmente más que de la forma en la que las definiciones de los procesos establecen que debería trabajar.*
- *Los requerimientos que se derivan de la cooperación y conocimiento de las actividades de la gente.*

Los estudios etnográficos pueden revelar los detalles de los procesos críticos que otras técnicas de obtención de requerimientos a menudo olvidan.

Sin embargo, puesto que se centran en el usuario final, este enfoque no es apropiado para descubrir los requerimientos organizacionales o del dominio.

Los estudios etnográficos no siempre pueden identificar nuevas propiedades que se deban agregar al sistema.

Por lo tanto, la etnografía no es un enfoque completo para la obtención de requerimientos por sí mismo, y debe utilizarse para complementar otros enfoques, como el análisis de casos de uso.

Para complementar el tema de las distintas estrategias de educación se presenta el siguiente cuadro comparativo.



Herramientas	Extracción	Análisis	Especificación	Validación
Entrevistas y cuestionarios	X			
Sistemas existentes	X	X		
Grabaciones de video y de audio	X	X		
Brainstorming (lluvia de ideas)	X	X		
Arqueología de Documentos	X	X		
Observación	X			
Prototipo Throw Away (no funcional)	X	X	X	
Prototipo Evolutionary (funcional)	X		X	X
Análisis DOFA		X		
Cadena de Valor		X		
Modelo Conceptual		X	X	
Diagrama de Pescado	X	X	X	
Glosario	X	X	X	X
Diagrama de Actividad		X	X	
Casos de uso	X	X	X	X
Casa de Calidad o QFD				X
Checklist	X		X	

En el presente cuadro se aprecia de cada una de las herramientas presentadas, cuales cubren las siguientes características de extracción, análisis, especificación y validación de los requisitos. Será importante entonces determinar que es un requisito.

## Requisito

Según Zuloaga el requisito es una condición o capacidad a la que el sistema (siendo construido) debe conformar.

Puede ser definido como: Una capacidad del software necesaria por el usuario para resolver un problema o alcanzar un objetivo. Una capacidad del software que debe ser reunida o poseída por un sistema o componente del sistema para satisfacer un contrato, especificación, estándar, u otra documentación formal”.

Existen algunas dificultades para definir los requerimientos. Entre a jugar lo que el usuario definió y transitar por el camino corto-largo de lo que el analista entendió u observo. Más adelante en el proceso se verá que es lo que el programador codificó (escribió), para luego ver, finalmente, cómo funciona el producto y contrarrestar con lo que realmente lo que “quería” el usuario.

El término *requerimiento* no se utiliza de una forma constante en la industria de software. En algunos casos, un requerimiento es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste. En el otro extremo, es una definición detallada y formal de una función del sistema.

- Los requerimientos del usuario son declaraciones, en lenguaje natural y en diagramas, de los servicios que se espera que el sistema proporcione y de las restricciones bajo las cuales debe funcionar.
- Los requerimientos del sistema establecen con detalle las funciones, servicios y restricciones operativas del sistema. El documento de requerimientos del sistema (algunas veces denominado especificación funcional) debe ser preciso. Debe



definir exactamente qué es lo que se va a implementar. Puede ser parte del contrato entre el comprador del sistema y los desarrolladores del software.

### **Ingeniería de Requerimientos**

El proceso de establecer los servicios que el cliente requiere de un sistema y los límites bajo los cuales opera y se desarrolla. Las malas o ineficientes prácticas de la Ingeniería de Requerimientos llevan invariablemente al fracaso del desarrollo del software, y pueden ser más costosas, dependiendo de qué tan tarde estas son descubiertas en el proceso de desarrollo. Es necesaria una disciplina en el desarrollo de software y en particular en el proceso de Ingeniería de Requerimientos a fin de evitar que el desarrollo de software falle o que sufra de costos excesivos.

El éxito de un sistema de software se mide de acuerdo al grado con que este y su proyecto de desarrollo cumplen con el objetivo para el cual fueron requeridos.

El problema del desarrollo de los sistemas de software es que los requerimientos son inherentemente dinámicos. Los cambios ocurren constantemente y esto se debe a: mejoras, cambios por errores descubiertos, cambios por adopción de nuevas tecnologías, cambios por mejoras en la comprensión del sistema, entre otros.

El proceso de Ingeniería de Requerimientos debe ser preciso y flexible a la vez. Preciso por que debe incluir todos los requerimientos del cliente y del ambiente donde este estará operando y flexible, ya que los requerimientos están sujetos a constantes cambios.

### **Que es un requerimiento?**

Puede variar desde unos estatutos abstractos en alto nivel de un servicio o unas restricciones del sistema hasta una especificación funcional matemática detallada.

Los Requerimientos pueden servir Como una función dual:

- Puede ser la base para la declaración de un contrato, por lo tanto, deber estar abierto a interpretación.
- Puede ser la base para el contrato en sí, por lo tanto, debe ser definido en detalle.
- Ambas declaraciones serán llamadas Requerimientos.

Un requerimiento de software define las funciones, capacidades o atributos de cualquier sistema de software.

También representan:



- Factores de calidad del sistema que permitirán evaluar su utilidad a un cliente o usuario.
- Los datos de entrada al proceso de desarrollo de software y representan lo que se requiere implementar.
- Una descripción de cómo el sistema deberá comportarse, describe información del dominio de la aplicación, describe restricciones de la operación del sistema y especifica atributos o propiedades del sistema.
- Un problema por resolver.

No se deben incluir aspectos de diseño, que especifiquen como deben implementarse tales requerimientos, ni detalles de planeación del proyecto o de las pruebas. Es importante separar lo que se requiere (que se detalla con los requerimientos) de cómo se requiere que el sistema sea diseñado (que se detalla en la etapa del diseño).

Todo software tiene requerimientos que lo definen y quizás la parte más difícil de la construcción del software es la decisión de que es lo que se debe construir.

### **Tipos de requerimientos**

*Requerimientos de software.* Se refiere a lo que debe hacer y lo que no debe hacer el sistema, se pueden subdividir en requerimiento funcionales y requerimientos no funcionales.

*Requerimientos de hardware.* Dependen del volumen de operaciones, cantidad de usuarios y de las aplicaciones que deben de estar en uso cuando se implemente el sistema. Se pueden distinguir tipos de requerimientos de hardware, requerimientos de rendimiento, de interfaz, de ambiente.

### **Dificultades para definir los requerimientos**

Estos no son obvios. Generalmente provienen de muchas fuentes. No es sencillo de expresar en lenguaje natural. Es por esto que en la actualidad se busca que los requerimientos sean expresados “formalmente”. Existen diferentes niveles de detalle en su definición y pueden estar, casi siempre, relacionados unos con otros.

Estas dificultades nos llevan a especificar algunas de las características necesarias en la descripción de un requerimiento. Estas son:

- *Completo:* Cada requerimiento debe describir de manera completa la funcionalidad que debe cumplir. Debe contener toda la información necesaria para que el desarrollador diseñe e implemente tal funcionalidad.



- *Correcto*: Cada requerimiento debe describir de manera precisa la funcionalidad que se debe construir. Un requerimiento correcto no debe entrar en conflicto con otro requerimiento. Sólo los usuarios más representativos del sistema pueden determinar de manera precisa si un requerimiento es correcto o no.
- *Realizable*: Debe ser posible implementar cada requerimiento de acuerdo a las capacidades y limitaciones del sistema y el medio que lo rodea. Para garantizar que no se determinen requerimientos no realizables, se recomienda contar con personal al interior del equipo de analistas de requerimientos que pueda establecer las limitaciones técnicas y de costos.
- *Necesario*: Cada requerimiento debe documentar algo que los clientes realmente necesiten, algo que sea para conformidad de un sistema externo con el que se tenga interacción, o para satisfacer un estándar. Para determinar si un requerimiento es necesario se debe determinar quién lo propuso (conocer su origen).
- *Priorizable*: Es importante asignar una prioridad para cada requerimiento que indique que tan esencial es el mismo para la realización del producto. Se pueden perder elementos de juicio para el desarrollo del sistema si se asigna el mismo grado de prioridad a todos los requerimientos.
- *No Ambiguo*: Todos los lectores de un requerimiento deben llegar a una misma y consistente interpretación del mismo. El lenguaje usado en su definición, no debe causar confusiones al lector.
- *Verificable*: Un requerimiento es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.

De la misma manera, podemos concluir que las características para una buena especificación de los requerimientos se desea que:

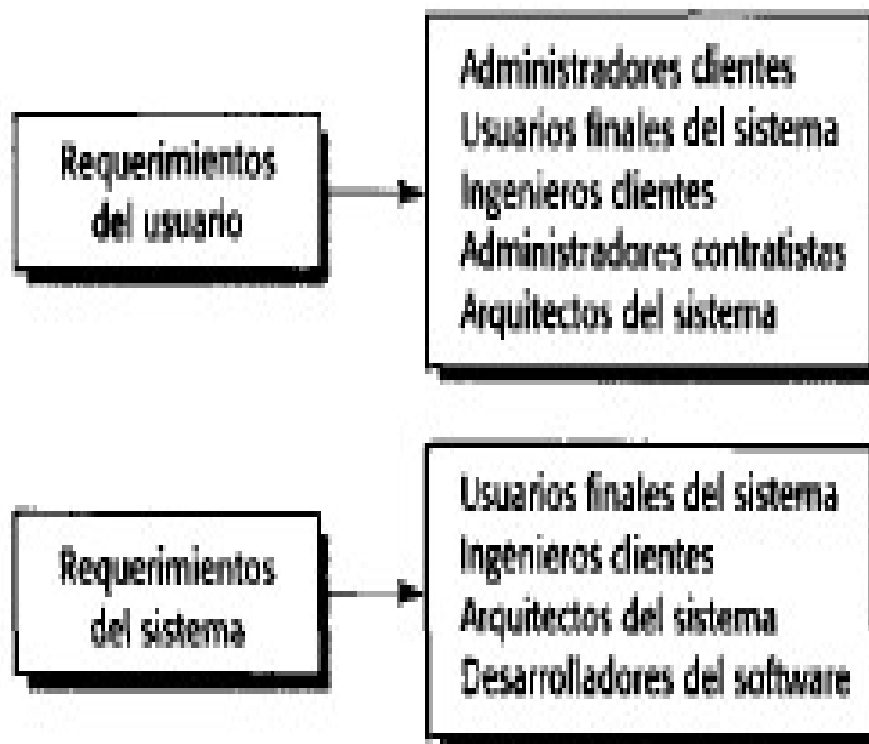
- *Completa*: Una especificación de requerimientos está completa si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.
- *Consistente*: Una especificación de requerimientos es consistente si no existen requerimientos que se contradigan.
- *Modificable*: Una especificación de requerimientos debe permitir ser revisada y mantener un historial de cambios hechos sobre cada requerimiento. Esto requiere que cada requerimiento sea etiquetado de manera única y expresado de manera separada de otros requerimientos para permitir referirse a él de manera no ambigua.





- *Trazable*: Cada requerimiento debe poder permitir trazar una línea del tiempo en la cual indique sus orígenes, y permita ser extendido a otras etapas del desarrollo del producto.

En la figura siguiente se describe a quienes les interesa u usa los requerimientos de usuario y de sistemas.



### Requerimientos del usuario

Deben describir los requerimientos funcionales y no funcionales de tal forma que sean comprensibles por los usuarios del sistema sin conocimiento técnico detallado. Únicamente deben especificar el comportamiento externo del sistema y deben evitar, tanto como sea posible, las características de diseño del sistema. Deben redactarse en un lenguaje sencillo, con tablas y formularios sencillos y diagramas intuitivos.

A continuación se describen algunos de los problemas cuando se redactan con frases del lenguaje natural en un documento de texto:



- ❑ *Falta de claridad.* Algunas veces es difícil utilizar el lenguaje de forma precisa y no ambigua sin hacer el documento poco conciso y difícil de leer.
- ❑ *Confusión de requerimientos.* No se distinguen claramente los *requerimientos funcionales* y no funcionales, las metas del sistema y la información para el diseño.
- ❑ *Conjunción de requerimientos.* Diversos requerimientos diferentes se pueden expresar de forma conjunta como un único requerimiento.

### Requerimientos del sistema

Los requerimientos del sistema son versiones extendidas de los requerimientos del usuario que son utilizados por los ingenieros de software como punto de partida para el diseño del sistema. Agregan detalle y explican cómo el sistema debe proporcionar los requerimientos del usuario. Pueden ser utilizados como parte del contrato para la implementación del sistema y, por lo tanto, deben ser una especificación completa y consistente del sistema entero. Deben describir el comportamiento externo del sistema y sus restricciones operativas. No deben tratar de cómo se debe diseñar o implementar el sistema.

Existen varias razones para esto:

- ❑ Puede tener que diseñar una arquitectura inicial del sistema para ayudar a estructurar la especificación de requerimientos.
- ❑ En muchos casos, los sistemas deben inter operar con otros ya existentes. Esto restringe el diseño, y estas restricciones imponen requerimientos en el sistema nuevo.
- ❑ Es necesario el uso de una arquitectura específica para satisfacer los requerimientos no funcionales

### Requerimientos funcionales y no funcionales

Los requerimientos de sistemas software se clasifican en funcionales y no funcionales, o como requerimientos del dominio:

- ✓ *Requerimientos funcionales.* Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer.
- ✓ *Requerimientos no funcionales.* Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de



desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema.

- ✓ *Requerimientos del dominio. Son requerimientos que provienen del dominio de aplicación del sistema y que reflejan las características y restricciones de ese dominio. Pueden ser funcionales o no funcionales.*

### **Especificación de los Requerimientos**

Se puede utilizar el lenguaje natural para redactar, además de los requerimientos del usuario, las especificaciones de requerimientos del sistema. Sin embargo, debido a que los requerimientos del sistema son más detallados que los requerimientos del usuario, las especificaciones en lenguaje natural pueden ser confusas y difíciles de entender:

- u La comprensión del lenguaje natural depende de que los lectores y redactores de la especificación utilicen las mismas palabras para el mismo concepto.
- u Una especificación de requerimientos en lenguaje natural es demasiado flexible.
- u No existe una forma fácil de modularizar los requerimientos en lenguaje natural.

Debido a estos problemas, las especificaciones de requerimientos redactadas en lenguaje natural son propensas a malas interpretaciones. A menudo éstas no se descubren hasta las fases posteriores del proceso del software, y resolverlas puede resultar muy costoso. Es esencial redactar los requerimientos del usuario en un lenguaje que los no especialistas puedan entender.

La figura siguiente refiere a distintos lenguajes de especificación de requerimientos.

### **Pasos a realizar en la Ingeniería de Requerimientos**

Estos pasos son un proceso iterativo y comprende las siguientes etapas:

Estudio de Factibilidad, el análisis de requerimientos, Definición, que es entender el problema, y finalmente, especificar, o sea, describir el problema. Será necesario también la validación y la negociación con los intervinientes en el proceso de desarrollo.

Un estudio de viabilidad es un estudio corto y orientado a resolver varias cuestiones:

1. ¿Contribuye el sistema a los objetivos generales de la organización?



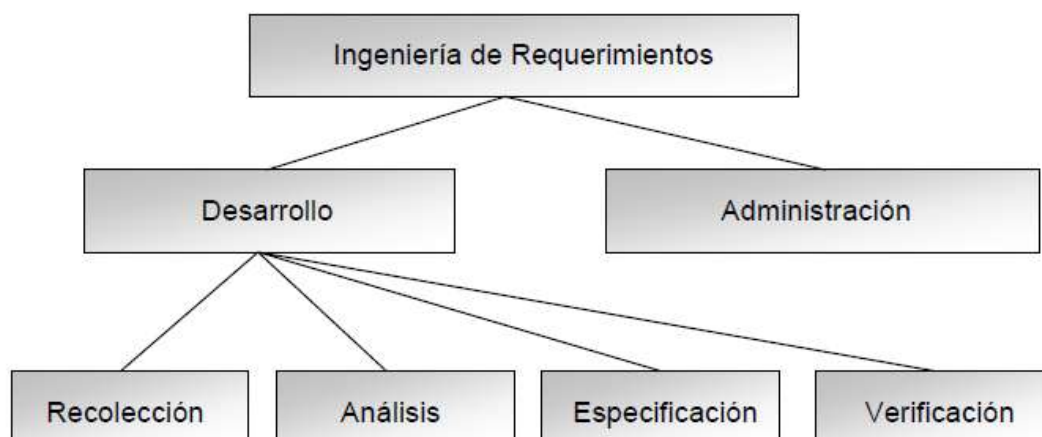
2. ¿Se puede implementar el sistema utilizando la tecnología actual y dentro de las restricciones de coste y tiempo?

3. ¿Puede integrarse el sistema con otros sistemas existentes en la organización?

Comprende la evaluación y recopilación de la información, y la redacción de informes. La fase de evaluación de la información identifica la información requerida para contestar las tres preguntas anteriores. Una vez que dicha información se ha identificado, se debería hablar con las fuentes de información para descubrir las respuestas a estas preguntas. Se pueden consultar las fuentes de información, como los jefes de los departamentos donde se utilizará el sistema, los ingenieros de software que están familiarizados con el tipo de sistema propuesto, los expertos en tecnología y los usuarios finales del sistema. Normalmente, se debería intentar completar un estudio de viabilidad en dos o tres semanas.

Una vez que se tiene la información, se redacta el informe del estudio de viabilidad. Debería hacerse una recomendación sobre si debe continuar o no el desarrollo del sistema. En el informe, se pueden proponer cambios en el alcance, el presupuesto y la confección de agendas del sistema y sugerir requerimientos adicionales de alto nivel para éste.

La figura siguiente refiere la estructura de la Ingeniería de Requerimientos, conformada por dos grandes etapas (desarrollo y administración) y cada una de ellas con sus correspondientes tareas.



La etapa de Desarrollo de Requerimientos consisten en:

- **Recolección (Elicitation):** Es el Proceso a través del cual los clientes (compradores y/o usuarios) y el desarrollador (contratista) de un sistema de software;



descubren, revisan, articulan, y entienden las necesidades de los usuarios del sistema y las restricciones que se dan sobre el software y el desarrollo del mismo.

- **Análisis (Analysis):** Es el proceso de analizar las necesidades de los clientes y los usuarios para llegar a una definición de los requerimientos de software.
- **Especificación (Specification):** Consiste en el desarrollo de un documento que de manera clara y precisa contenga y especifique cada uno de los requerimientos del sistema de software.
- **Verificación (Verification):** Es el proceso de asegurar que la especificación de requerimientos de software sea acorde con los requerimientos del sistema, conforme a los estándares de documentación de la fase de requerimientos, y que a su vez este documento sea una base sólida para la arquitectura y el diseño.

### Problemas en la definición de los requerimientos

Los sistemas de software grandes siempre presentan problemas. Problemas que son tan complejos que puede ser que nunca se comprendan completamente y donde los desarrolladores van comprendiendo el sistema durante su desarrollo. Por lo tanto, los requerimientos son normalmente incompletos e inconsistentes.

Las razones de esa inconsistencia son:

- Los sistemas de software grandes deben permitir una mejora en la situación actual de la empresa. Es difícil anticipar los efectos que el sistema tendrá en la organización.
- Usuarios diferentes tienen requerimientos y prioridades diferentes. Hay constantemente cambios en los requerimientos.
- Los usuarios finales del sistema y la organización que paga por el sistema tienen requerimientos diferentes.
- El prototipado es requerido para clarificar requerimientos.

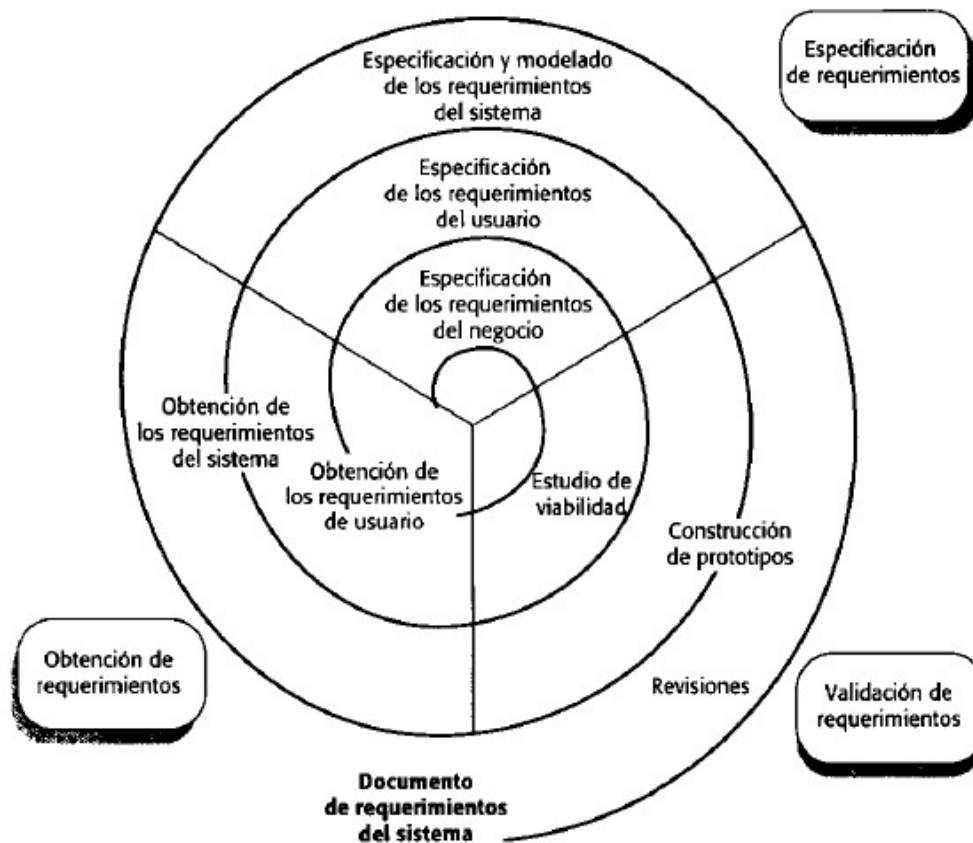
### Administración de los requerimientos

Son muchas y variadas las definiciones de lo que es una “administración de requerimientos” pero lo importante es darnos cuenta que los requerimientos son necesarios gestionarlos.

Podríamos decir entonces que la gestión de un requerimiento es una forma sistemática de obtener, organizar y documentar los requisitos de un sistema, y un proceso que establece



y mantiene el acuerdo entre el cliente y el equipo de proyecto sobre los requisitos cambiantes. (Baufest).



La figura precedente presenta un proceso continuo e iterativo de la ingeniería de requerimiento.

Los requerimientos para sistemas software grandes son siempre cambiantes. Debido a que el problema no puede definirse completamente, es muy probable que los requerimientos del software sean incompletos. Durante el proceso del software, la comprensión del problema por parte de los stakeholders está cambiando constantemente. Estos requerimientos deben entonces evolucionar para reflejar esta perspectiva cambiante del problema. Además, una vez que un sistema se ha instalado, inevitablemente surgen nuevos requerimientos.

Cuando los usuarios finales tienen experiencia con un sistema, descubren nuevas necesidades y prioridades. Normalmente, los sistemas grandes tienen una comunidad de usuarios diversa donde los usuarios tienen diferentes requerimientos y prioridades. Estos pueden contradecirse o estar en conflicto.



Las personas que pagan por el sistema y los usuarios de éste raramente son la misma persona. Los clientes del sistema imponen requerimientos debido a las restricciones organizacionales y de presupuesto. El entorno de negocios y técnico del sistema cambia después de la instalación, y estos cambios se deben reflejar en el sistema.

### **Planificación de la gestión de requerimientos**

La planificación es una primera etapa esencial del proceso de gestión de requerimientos. La gestión de requerimientos tiene un coste elevado. Para cada proyecto, la etapa de planificación establece el nivel de detalle necesario en la gestión de requerimientos.

Durante la etapa de gestión de requerimientos, habrá que decidir sobre:

- *La identificación de requerimientos. Cada requerimiento se debe identificar de forma única de tal forma que puedan ser remitidos por otros requerimientos de manera que pueda utilizarse en las evaluaciones de rastreo.*
- *Un proceso de gestión del cambio. Éste es el conjunto de actividades que evalúan el impacto y coste de los cambios.*
- *Políticas de rastreo. Estas políticas definen las relaciones entre los requerimientos, y entre éstos y el diseño del sistema que se debe registrar y la manera en que estos registros se deben mantener.*
- *Ayuda de herramientas CASE. La gestión de requerimientos comprende el procesamiento de grandes cantidades de información sobre los requerimientos. Las herramientas que se pueden utilizar van desde sistemas de gestión de requerimientos especializados hasta hojas de cálculo y sistemas sencillos de bases de datos.*

Existen tres tipos de información de rastreo que pueden ser mantenidos:

- *La información de rastreo de la fuente vincula los requerimientos con los stakeholders que propusieron los requerimientos y la razón de éstos. Cuando se propone un cambio, esta información se utiliza para encontrar y consultar a los stakeholders sobre el cambio.*
- *La información de rastreo de los requerimientos vincula los requerimientos dependientes en el documento de requerimientos. Esta información se utiliza para evaluar cómo es probable que muchos requerimientos se vean afectados por un cambio propuesto y la magnitud de los cambios consecuentes en los requerimientos.*
- *La información de rastreo del diseño vincula los requerimientos a los módulos del diseño en los cuales son implementados. Esta información se utiliza para evaluar el*





impacto de los cambios de los requerimientos propuestos en el diseño e implementación del sistema.

La gestión de requerimientos necesita ayuda automatizada; las herramientas CASE para esto deben elegirse durante la fase de planificación. Se precisan herramientas de ayuda para:

- *Almacenar requerimientos. Los requerimientos deben mantenerse en un almacén de datos seguro y administrado que sea accesible a todos los que estén implicados en el proceso de ingeniería de requerimientos.*
- *Gestionar el cambio. Este proceso se simplifica si está disponible una herramienta de ayuda.*
- *Gestionar el rastreo. Como se indicó anteriormente, las herramientas de ayuda para el rastreo permiten que se descubran requerimientos relacionados.*

### Gestión del cambio de los requerimientos

La gestión del cambio de los requerimientos se debe aplicar a todos los cambios propuestos en los requerimientos. La ventaja de utilizar un proceso formal para gestionar el cambio es que todos los cambios propuestos son tratados de forma consistente y que los cambios en el documento de requerimientos se hacen de forma controlada. Existen tres etapas principales en un proceso de gestión de cambio:

- *Análisis del problema y especificación del cambio. El proceso empieza con la identificación de un problema en los requerimientos o, algunas veces, con una propuesta de cambio específica.*
- *Análisis del cambio y cálculo de costes. El efecto de un cambio propuesto se valora utilizando la información de rastreo y el conocimiento general de los requerimientos del sistema.*
- *Implementación del cambio. Se modifica el documento de requerimientos y, en su caso, el diseño e implementación del sistema.*

### El documento de requerimientos del software

El documento de requerimientos del software (algunas veces denominado especificación de requerimientos del software o SRS) es la declaración oficial de qué deben implementar los desarrolladores del sistema.

Debe incluir tanto los requerimientos del usuario para el sistema como una especificación detallada de los requerimientos del sistema.

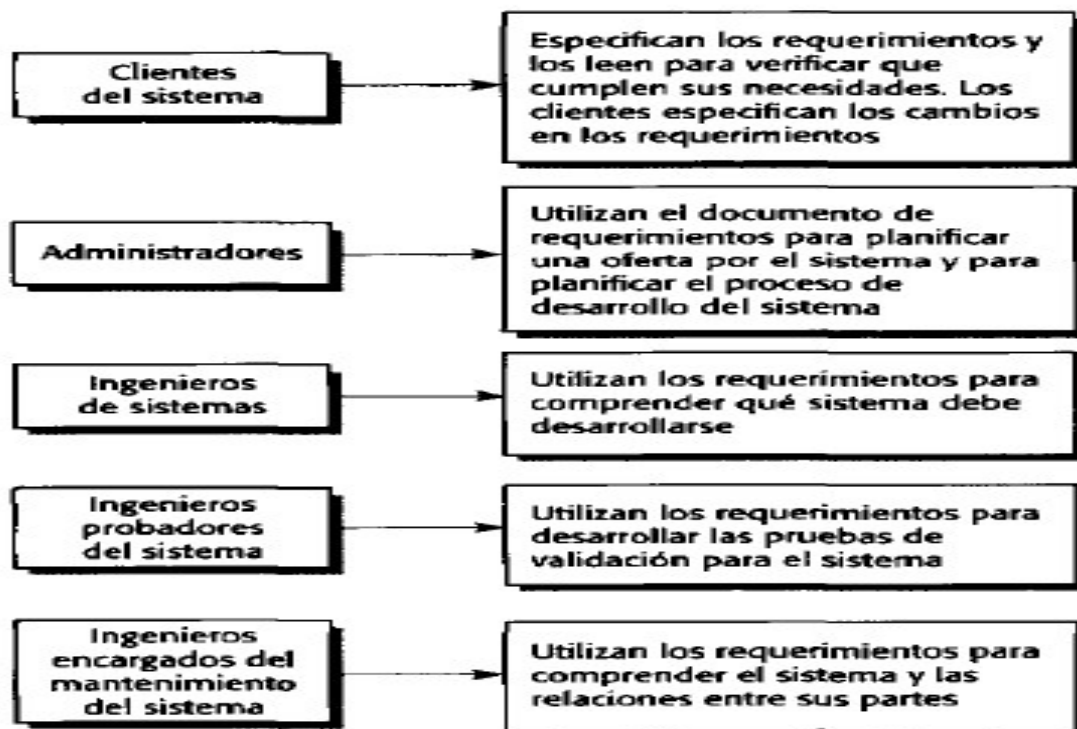


El documento de requerimientos tiene un conjunto de usuarios del documento que lo utilizan.

La diversidad de posibles usuarios significa que el documento de requerimientos tiene que presentar un equilibrio entre la comunicación de los requerimientos a los clientes, la definición de los requerimientos en el detalle exacto para los desarrolladores y probadores, y la inclusión de información sobre la posible evolución del sistema.

El nivel de detalle que se debe incluir en un documento de requerimientos depende del tipo de sistema que se desarrolle y del proceso de desarrollo utilizado.

La figura siguiente refiere a lo que observan de dicho documento los intervinientes en el desarrollo del software.



### Estructura del Documento de Requerimientos

- **Introducción.** Describe la necesidad de crear el sistema y cuáles son sus objetivos de negocios.
- **Glosario.** Define los términos técnicos usados.
- **Modelos del Sistema.** Define los modelos Mediante los cuales se muestran los componentes del sistema y las relaciones entre ellos.
- **Definición de Requerimientos Funcionales.** Define los servicios que serán proporcionados.



- Definición de Requerimientos No-funcionales. Definir las restricciones del sistema y el proceso de desarrollo.
- Evolución del Sistema. Definir las suposiciones fundamentales en las cuales el sistema se basa y los cambios que preveen.
- Especificación de Requerimientos. Especificación detallada de los requerimientos funcionales del sistema.
- Apéndices. Descripción de la plataforma de Hardware del Sistema. Requerimientos de la base de Datos (quizá como un modelo Entidad Relacion)
- Índice.

### IEEE-STD-830-1998 : ESPECIFICACIONES DE LOS REQUISITOS DEL SOFTWARE

El formato de Especificación de Requisitos Software (ERS) según la última versión del estándar IEEE 830. Según IEEE, un buen Documento de Requisitos, pese a no ser obligatorio que siga estrictamente la organización y el formato dados en el estándar 830, se deberá incluir, de una forma o de otra, toda la información presentada en dicho estándar. El estándar de IEEE 830 no está libre de defectos ni de prejuicios, y por ello ha sido justamente criticado por múltiples autores y desde varios puntos de vista, llegándose a cuestionar incluso si es realmente un estándar en el sentido habitual que tiene el término en otras ingenieras.

A continuación se detallan las partes de documento IEEE830

- u 1. Introducción
- u 1.1. Propósito.
- u 1.2. Ambito del Sistema
- u 1.3. Definiciones, Acrónimos y Abreviaturas
- u 1.4. Referencias
- u 1.5. Vision General del Documento
- u 2. Descripción General
- u 2.1. Perspectiva del Producto
- u 2.2. Funciones del Producto
- u 2.3. Características de los Usuarios
- u 2.4. Restricciones
- u 2.5. Suposiciones y Dependencias
- u 2.6. Requisitos Futuros
- u 3. Requisitos Específicos
- u 3.1. Interfaces Externas
- u 3.2. Funciones
- u 3.3. Requisitos de Rendimiento
- u 3.4. Restricciones de Diseño
- u 3.5. Atributos del Sistema
- u 3.6. Otros Requisitos
- u 4. Apéndices

### Formulario IEEE830



**Introducción. Propósito.** En esta subsección se definirá el propósito del documento ERS y se especificará a quien va dirigido el documento.

**Introducción. Ámbito del sistema**

- u Se podrá dar un nombre al futuro sistema (p.ej. MiSistema)
- u Se explicara lo que el sistema hará y lo que no hará.
- u Se describirán los beneficios, objetivos y metas que se espera alcanzar con el futuro sistema.
- u Se referenciaran todos aquellos documentos de nivel superior (Ej.: en Ingeniera de Sistemas, que incluyen Hardware y Software, deberá mantenerse la consistencia con el documento de especificación de requisitos globales del sistema, si existe).

**Introducción. Definiciones.** En esta subsección se definirán todos los términos acrónimos y abreviaturas utilizadas en la ERS.

**Referencia.** En esta subsección se mostrara una lista completa de todos los documentos referenciados en la ERS.

**Visión general.** Esta subsección describe brevemente los contenidos y la organización del resto de la ERS.

### Descripción General

En esta sección se describen todos aquellos factores que afectan al producto y a sus requisitos. No se describen los requisitos, sino su contexto. Esto permitirá definir con detalle los requisitos en la sección 3, haciendo que sean más fáciles de entender.

Incluye

- ☐ Perspectiva del Producto
- ☐ Funciones del Producto
- ☐ Características de los Usuarios
- ☐ Restricciones
- ☐ Suposiciones y Dependencias
- ☐ Requisitos Futuros

### Perspectiva del Producto

- u Esta subsección debe relacionar el futuro sistema (producto software) con otros productos.
- u Si el producto es totalmente independiente de otros productos, también debe especificarse aquí.
- u Si la ERS define un producto que es parte de un sistema mayor, esta subsección relacionara los requisitos del sistema mayor con la funcionalidad del producto descrito en la ERS, y se identificaran las interfaces entre el producto mayor y el producto aquí descrito.
- u Se recomienda utilizar diagramas de bloques.

**Funciones del Producto.** En esta subsección de la ERS se mostrara un resumen, a grandes rasgos, de las funciones del futuro sistema. (ej. En una ERS para un programa de contabilidad, esta subsección mostrara que el sistema soportara el mantenimiento de



cuentas, mostrara el estado de las cuentas y facilitara la facturación, sin mencionar el enorme detalle que cada una de estas funciones requiere).

Las funciones deberán mostrarse de forma organizada, y pueden utilizarse gráficos, siempre y cuando dichos gráficos reflejen las relaciones entre funciones y no el diseño del sistema.

### Características de los Usuarios

Esta subsección describirá las características generales de los usuarios del producto, incluyendo nivel educacional, experiencia y experiencia técnica.

**Restricciones.** Esta subsección describirá aquellas limitaciones que se imponen sobre los:

- u desarrolladores del producto
- u Políticas de la empresa
- u Limitaciones del hardware
- u Interfaces con otras aplicaciones
- u Operaciones paralelas
- u Funciones de auditora
- u Funciones de control
- u Lenguaje(s) de programación
- u Protocolos de comunicación
- u Requisitos de habilidad
- u Criticalidad de la aplicación
- u Consideraciones acerca de la seguridad

**Suposiciones y Dependencias.** Esta subsección de la ERS describirá aquellos factores que, si cambian, pueden afectar a los requisitos. (Ej.: los requisitos pueden presuponer una cierta organización de ciertas unidades de la empresa, o pueden presuponer que el sistema correrá sobre cierto sistema operativo. Si cambian dichos detalles en la organización de la empresa, o si cambian ciertos detalles técnicos, como el sistema operativo, puede ser necesario revisar y cambiar los requisitos).

**Requisitos Futuros.** Esta subsección esbozara futuras mejoras al sistema, que podrán analizarse e implementarse en un futuro.

**Requisitos Específicos.** Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas. Esta es la sección más larga e importante de la ERS.

### Requisitos Específicos

Deberán aplicarse los siguientes principios:

- ☐ El documento deberá ser perfectamente legible por personas de muy distintas formaciones e intereses.



- ☐ Deberán referenciarse aquellos documentos relevantes que poseen alguna influencia sobre los requisitos.
- ☐ Todo requisito deberá ser unívocamente identificable mediante algún código o sistema de numeración adecuado.
- ☐ Lo ideal, aunque en la práctica no siempre realizable, es que los requisitos posean las siguientes características: Corrección; No ambiguos; Completos; consistentes; clasificados; verificables; modificables; trazables.

Incluye:

- ☐ Interfaces Externas
- ☐ Funciones
- ☐ Requisitos de Rendimiento
- ☐ Restricciones de Diseño
- ☐ Atributos del Sistema
- ☐ Otros Requisitos

**Requisitos Específicos. Interfaces Externas.** Se describirán los requisitos que afecten a la interfaz de usuario, interfaz con otros sistemas (hardware y software) e interfaces de comunicaciones.

**Requisitos Específicos. Funciones.**

Esta subsección (quizá la más larga del documento) deberá especificar todas aquellas acciones (funciones) que deberá llevar a cabo el software. Normalmente (aunque no siempre), son aquellas acciones expresables como "el sistema deberá. . .". Si se considera necesario, podrán utilizarse notaciones gráficas y tablas, pero siempre supeditadas al lenguaje natural, y no al revés.

Es importante tener en cuenta que, en 1983, el Estándar de IEEE 830 establece que las funciones deberán expresarse como una jerarquía funcional (en paralelo con los DFDs propuestos por el análisis estructurado).

El Estándar de IEEE 830, en sus últimas versiones, permite organizar esta subsección de múltiples formas, sugiere, entre otras, las siguientes:

- ☐ Por tipos de usuario: Distintos usuarios poseen distintos requisitos. Para cada clase de usuario que exista en la organización, se especificaran los requisitos funcionales.
- ☐ Por objetos: Los objetos son entidades del mundo real que serán reflejadas en el sistema. Para cada objeto, se detallaran sus atributos y sus funciones.
- ☐ Por objetivos: Un objetivo es un servicio que se desea que ofrezca el sistema y que requiere una determinada entrada para obtener su resultado. Para cada objetivo o su objetivo que se persiga con el sistema, se detallaran las funciones que permitan llevarlo a cabo.
- ☐ Por estímulos: Se especificaran los posibles estímulos que recibe el sistema y las funciones relacionadas con dicho estímulo.



- ❑ Por jerarquía funcional: Si ninguna de las anteriores alternativas resulta de ayuda, la funcionalidad del sistema se especificara como una jerarquía de funciones que comparten entradas, salidas o datos internos. Se detallaran las funciones (entrada, proceso, salida) y las sub funciones del sistema.

Requisitos Específicos.

Requisitos de rendimiento.

Se detallaran los requisitos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema, etc.

También, si es necesario, se especificaran lo requisitos de datos, es decir, aquellos requisitos que afecten a la información que se guardara en la base de datos. Por ejemplo, la frecuencia de uso, las capacidades de acceso y la cantidad de registros que se espera almacenar (decenas, cientos, miles o millones).

Restricciones de diseño.

Todo aquello que restrinja las decisiones relativas al diseño de la aplicación: Restricciones de otros estándares, limitaciones del hardware, etc.

Requisitos Específicos. Atributos del Sistema.

Se detallaran los atributos de calidad del sistema: Fiabilidad, mantenibilidad, portabilidad, y, muy importante, la seguridad. Deberá especificarse que tipos de usuario está autorizados, o no, a realizar ciertas tareas, y como s implementaran los mecanismos de seguridad (por ejemplo, por medio de un login y una password).

Otros.

Cualquier otro requisito que no encaje en otra sección.

Apéndices.

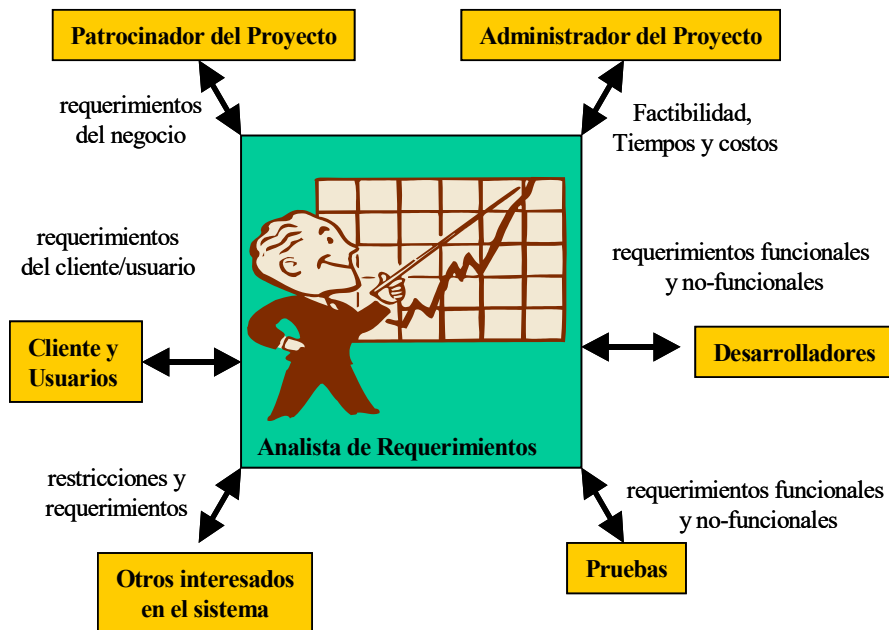
Pueden contener todo tipo de información relevante para la ERS pero que, propiamente, no forme parte de la ERS. Por ejemplo:

- Formatos de entrada/salida de datos, por pantalla o en listados.
- Resultados de análisis de costes.
- Restricciones acerca del lenguaje de programación.

## Los actores del proceso de la Ingeniería de requerimientos

### El Analista de requerimiento

Algunas de las actividades del analista de requerimiento se ven reflejadas en la siguiente figura.



Se pueden resumir en las siguientes:

- Definir los objetivos del proyecto y los beneficios al negocio.
- Identificar el problema a resolver y obtener los requerimientos.
- Identificar a los involucrados en el desarrollo del proyecto así como a las clases de clientes y usuarios.
- Identificar el ambiente del dominio a desarrollar y estar preparado para desarrollar el sistema requerido.
- Administrar los requerimientos utilizando un proceso y un plan de requerimientos.
- Modelar los requerimientos.
- Realizar control de cambios en los requerimientos.

Tendrá que tener las siguientes habilidades: a) Capacidad de comunicación; b) Capacidad de análisis y observación; c) Capacidad de organización; d) Analizar los riesgos del desarrollo del software.

## El Cliente

Sus actividades y responsabilidades deberán ser:

- Educar al analista de requerimientos acerca del negocio y sus objetivos.
- Ser claro y preciso acerca del problema que se quiere resolver.
- Colaborar con el analista en la definición de los requerimientos.
- Revisar los documentos de requerimientos y el avance del proyecto.
- Comunicar a los analistas sobre cambios en los requerimientos.
- Plantear costos y tiempos esperados de desarrollo y estar abierto a discutir cambios en los costos y tiempos de entrega.
- Estar siempre dispuesto a reunirse con los desarrolladores para discutir distintos aspectos del proyecto.





- Respetar los procesos que implementarán los desarrolladores para implementar el producto.

### El Usuario

#### Clasificación de los usuarios:

- La frecuencia con la que usan el sistema.
- Las funciones que usan del sistema y su frecuencia.
- La experiencia en el dominio de la aplicación y su experiencia con otros sistemas similares.
- El tipo de uso que le dan al sistema (operación, administración, mantenimiento, supervisión).
- Las tareas que desempeñan en soporte de los procesos de la organización.
- Sus privilegios de acceso o niveles de seguridad (tales como usuario invitado, administrador o usuario de nivel interno).
- Tipo de usuarios necesarios para operar el sistema (persona, grupo de personas, robot, u otra computadora).

#### Requerimientos de acuerdo a su característica

- **Requerimientos de dominio.** Los requerimientos de dominio son requerimientos que provienen del dominio de aplicación del sistema y reflejan las características de este dominio.
- **Requerimientos de Datos.** Los requerimientos de datos definen las estructuras de datos requeridas en el sistema.
- **Requerimiento de Interfaz.** Definen las características y parámetros de la comunicación del sistema a desarrollar con otros sistemas dentro de la empresa, o incluso de los subsistemas.
- **Requerimientos Perdurables.** Rrequerimientos estables derivados de las actividades de la organización Del cliente. Por ejemplo, un hospital siempre tendrá doctores, enfermeras, etc. Puede ser derivado de modelos de dominio.
- **Requerimientos Volátiles.** Los requerimientos cambian durante el desarrollo o cuando el sistema está en uso. En un hospital, los requerimientos se derivan de las políticas salud-cuidados.
- **Requerimientos Cambiantes.** Los requerimientos que cambian por el ambiente del sistema.
- **Requerimientos Emergentes.** Requerimientos que surgen como una comprensión del desarrollo del sistema.
- **Requerimientos de Consecuencias.** Requerimientos que resultan de la introducción del sistema computacional.
- **Requerimientos de Compatibilidad.** Requerimientos que dependen de otros sistemas o de otros procesos de la organización.

#### Medidas en los requerimientos NO funcionales



En el cuadro siguiente se resume las propiedades a tener en cuenta en los requerimientos NO funcionales y que se deberá medir en cada una de ellas.

Propiedad	Medida
Velocidad	Transacciones por segundo Tiempo de respuesta a eventos
Tamaño	Número de líneas de código Numero de Bytes de Memoria disponible
Facilidad de uso	Tiempo de entrenamiento Numero de ayudas
Confiabilidad	Errores permitidos por unidad de tiempo Media de tiempo por fallo Disponibilidad en tiempo
Robustez	Tiempo para restablecer después de fallo. Porcentaje de fallos que causan caída del sistema.
Portabilidad	Facilidad de transportar a otro S.O o lenguaje.

### Trazabilidad

Es la capacidad de describir y de seguir la vida de un requisito, tanto en dirección hacia adelante y hacia atrás, es decir, desde sus orígenes, a través de su desarrollo y especificación, a su despliegue y uso subsecuentes, y a través de períodos de refinamiento y de la iteración en curso en cualquiera de estas fases. (Gotel).

La trazabilidad es la medida en la cual se puede establecer una relación entre dos productos o más del proceso de desarrollo.

La trazabilidad de requerimientos se refiere a la habilidad de seguir el ciclo de un requerimiento, tanto para atrás como hacia adelante.



## TIPOS

Permite que los usuarios rastreen como se implementa o prueba un requerimiento (o como se lo hará). Permite ver donde se originó un requerimiento.

En otras palabras, todos los documentos de requerimientos, documentos de diseño, código y guiones de prueba deberán a su fuente de origen.

### Trazabilidad Bidireccional

Contar con un mecanismo integral de trazabilidad facilita la finalización de las tareas relacionadas con pruebas y documentación.

Cuando ocurren cambios en el software, la trazabilidad hace que sea relativamente más fácil evaluar el impacto de que los cambios podrían tener en otras partes del proceso de desarrollo.

La trazabilidad vertical garantiza que todos los requerimientos sean diseñados, y que todos los diseños se codifiquen y prueben.

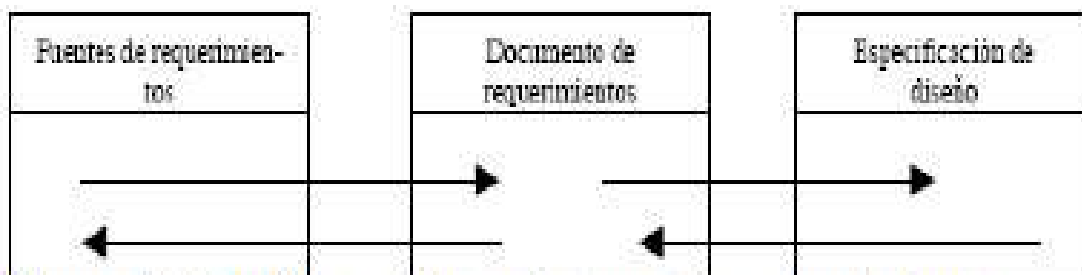
El mecanismo de trazabilidad no solo resalta las pruebas que se deben actualizar o repetir sino que también señala los documentos (análisis de riesgos, especificaciones y manuales del usuario) que se deben revisar.

La trazabilidad horizontal es la habilidad de relacionar las secciones/componentes de la misma fase entre si e identificar las dependencias que haya entre ellas, dependencias/relación de un componente/clase con otros componentes/clases.

Permite fácilmente detectar si hay conflictos entre requerimientos, diseño, lógica de codificación o casos de prueba.

Cuando se genera un control de cambios sobre un requerimiento, la trazabilidad horizontal brinda la opción de identificar si tendrá un impacto sobre algún otro aspecto de los requerimientos.

La figura siguiente muestra la relación que deben tener los requerimientos en las diferentes etapas.



**Figura 5. Relación que deben de tener los requerimientos en las diferentes etapas**



### Matriz de Trazabilidad

Durante el ciclo de vida del desarrollo de software de un proyecto existen dos fases muy importantes: definición de requisitos; y el diseño y ejecución de pruebas.

En proyectos grandes o complejos es difícil poder saber que test ejecutados o diseñados cubren cada una de las especificaciones o requerimientos del proyecto.

Es por este motivo que existe lo que se conoce como la matriz de trazabilidad.

La matriz de trazabilidad es una herramienta que se utiliza para saber que requerimientos quedan cubiertos por una prueba.

Imaginemos que tenemos un proyecto con 5 requerimientos (R1-R5) y hemos diseñado tres casos de prueba (T1-T3).

- El caso de prueba T1 cubre los requerimientos R1 y R4
- El caso de prueba T2 cubre los requerimientos R3 y R5
- El caso de prueba T3 cubre el requerimiento R3

En este caso la matriz resultante será:

	T1	T2	T3
R1	*		
R2			
R3		*	*
R4	*		
R5		*	

### Casos de Uso

Los casos de uso son una excelente herramienta para estimular a que los usuarios potenciales hablen, de un sistema, desde sus propios puntos de vista.

La idea es involucrar a los usuarios en las etapas iniciales del análisis y diseño del sistema. Esto aumenta la probabilidad de que el sistema sea de mayor provecho para la gente a la que supuestamente ayudará, en lugar de ser un manojito de expresiones de computación incomprensibles e inmanejables por los usuarios finales.

Cuando realizamos un análisis de casos de uso, no nos preocupamos por la forma de implementarlo. Tan solo nos preocupamos de cómo el sistema se comportará para alguien que tenga que utilizarlo.

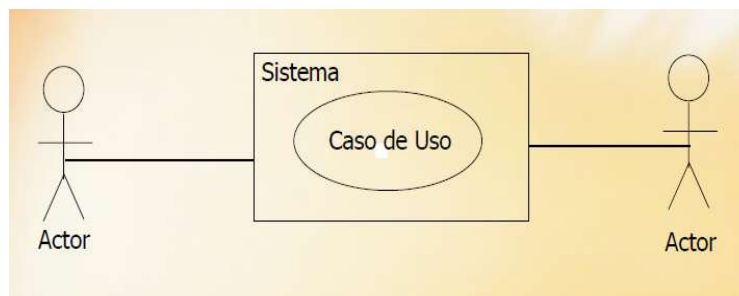
El objetivo es derivar una colección de casos de uso que, finalmente, mostraremos a las personas que diseñan y desarrollan el sistema, de manera que el resultado sea un sistema fácil de utilizar.



Para iniciar los casos de uso seguirá un conjunto de procedimientos:

- Entrevistas con los clientes (y entrevistas con expertos) que nos llevarán a los diagramas de clases.
- Esto nos dará cierta idea del área en la que trabajará y una familiaridad con los términos que utilizará. Posteriormente, contaremos con un fundamento para hablar con los usuarios.
- Seguidamente, entrevistaremos a los usuarios (preferentemente en grupos) y les pediremos que nos indiquen lo que ellos harían con el sistema que usted diseñará. Sus repuestas conformarán un conjunto candidato de casos de uso.
- Es importante describir brevemente cada caso de uso, además, de una lista de todos los actores que iniciarán y se beneficiarán de los casos de uso.
- Los casos de uso aparecerán en varias fases del proceso de desarrollo. Le ayudarán con el diseño de una interfaz del usuario, coadyuvarán con las opciones de desarrollo de los programadores y establecerán las bases para probar el sistema recién generado.

### Representación de un Modelo de Caso de Uso



La figura precedente refiere a un “casos de uso”, estos se representan por una elipse. Una figura agregada al caso de uso representa el actor. El actor que inicia el caso de uso se encuentra a la izquierda mientras que el que recibe la acción se encuentra a la derecha. El nombre del actor aparece justo debajo de él, y el nombre del caso de uso aparece ya sea dentro de la elipse o justo debajo de ella. Una línea asociativa conecta a un actor con el caso de uso, y representa la comunicación entre el actor y el caso de uso. La línea asociativa es sólida, como la que conecta a las clases asociadas.

### Secuencia de Pasos en los Escenarios

Cada caso de uso es una colección de escenarios y cada escenario es una secuencia de pasos.

El uso de los diagramas de casos de uso será, por lo general, parte de un documento de diseño que el cliente y el equipo de diseño tomarán como referencia.

Cada diagrama tendrá su propia página, de igual manera, cada escenario de caso de uso tendrá su propia página. El actor que inicia el caso de uso; Condiciones previas para el caso de uso; Pasos en el escenario; Condiciones posteriores cuando se finaliza el



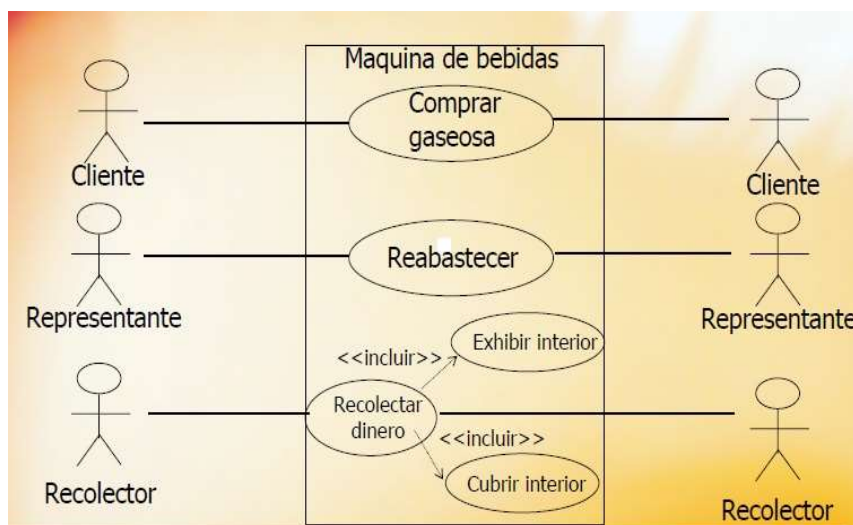
escenario; El actor que se beneficia del caso de uso. También se pueden utilizar conjeturas del escenario.

Los casos de uso se pueden relacionar entre sí como sigue:

- Inclusión. Le permite volver a utilizar los pasos de un caso de uso dentro de otro.
- Extensión. Le permite crear un caso de uso mediante la adición de pasos a uno existente.
- Generalización. Cuenta con un caso de uso que se hereda de otro.
- Agrupamiento. Es una manera de agrupar a los casos de uso.

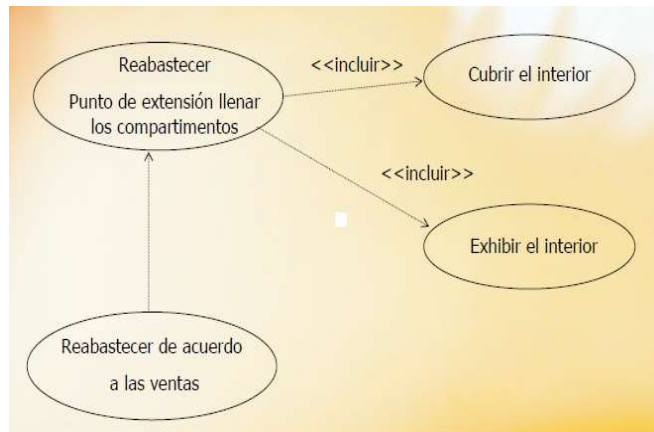
### Inclusión

Indica que un caso de uso está formado por otros. La representación se hace utilizando el símbolo que usó para la dependencia entre clases: una línea discontinua con una punta de flecha que conecta las clases apuntando hacia la clase dependiente. Justo sobre la línea, se agregará un estereotipo: la palabra <<incluir>> bordeada por dos pares de paréntesis angulares. Un caso de uso incluido nunca aparecerá solo, ya que simplemente funciona como parte de un caso de uso que lo incluya.



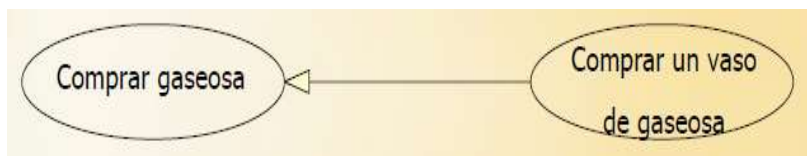
### Extensión

El caso de uso reabastecer, podría ser la base de otro caso de uso denominado “reabastecer de acuerdo a las ventas”. Es decir, reabastecer la máquina con aquellas bebidas que tienen mayor aceptación. Entonces podemos decir que el nuevo caso de uso extiende al original dado que agrega otros pasos a la secuencia del caso de uso original, que se conoce como el caso base. La extensión solo se puede realizar en determinados puntos que son conocidos como puntos de extensión.



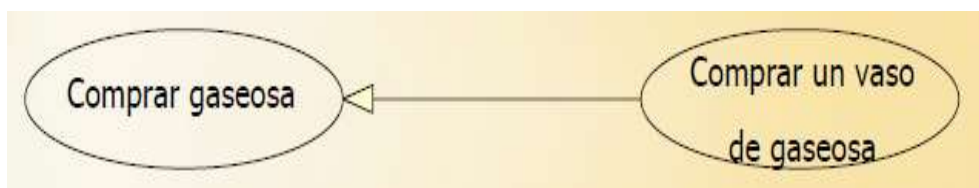
### Generalización

Los casos de uso pueden heredar entre sí. En la herencia de los casos de uso, el caso de uso secundario hereda las acciones y significado del primario, y además agrega sus propias acciones. Podemos aplicar el caso de uso secundario en cualquier lugar donde aplique el primario. Un caso de uso puede heredar el sentido y comportamiento de otro.

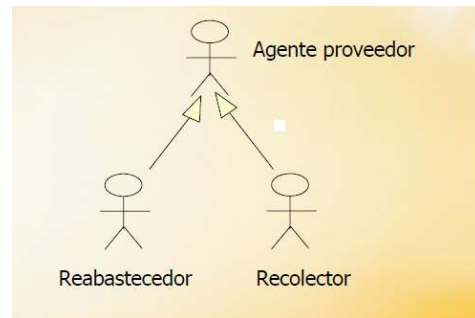


### Generalización

Los casos de uso pueden heredar entre sí. En la herencia de los casos de uso, el caso de uso secundario hereda las acciones y significado del primario, y además agrega sus propias acciones. Podemos aplicar el caso de uso secundario en cualquier lugar donde aplique el primario. Un caso de uso puede heredar el sentido y comportamiento de otro.



Como la muestra la figura siguiente, la herencia también se puede establecer entre actores.



### Agrupamiento

Cuando se dispone de varios casos de uso que se quieran agrupar, se puede hacer mediante paquetes. Por ejemplo, esto puede ocurrir cuando un sistema consta de varios subsistemas. Otra posibilidad es cuando entrevistamos a los usuarios para obtener los requerimientos de un sistema. Cada requerimiento podría ser representado como un caso de uso por separado.

### Diagramas de Casos de Uso en el Proceso de Análisis

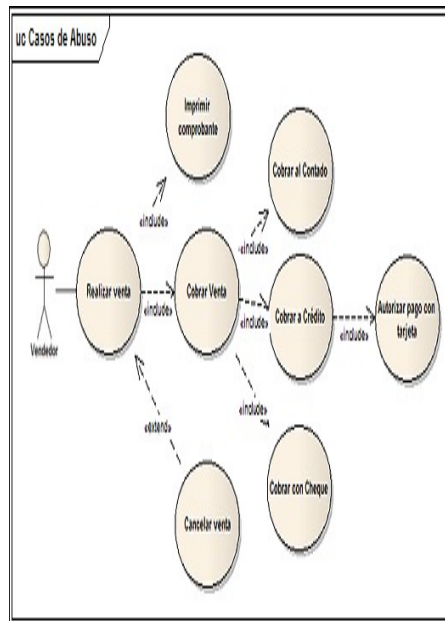
Los pasos son los siguientes:

- Las entrevistas inician el proceso, generando diagramas de clases.
- Una vez que conozcamos la terminología general del área del cliente, estamos listos para hablar con los usuarios.
- Las entrevistas iniciales darán como resultado la revelación de los actores y casos de uso de alto nivel que describirán los requerimientos funcionales en términos generales. Esta información establece los confines y ámbito del sistema.
- Las entrevistas posteriores profundizarán en estos requerimientos, que dará por resultado modelos de casos de uso que mostrarán los escenarios y las secuencias detalladamente.

### Casos de Abuso

Otro abuso de estas relaciones se da cuando queremos representar la unión de casos de uso por una decisión de diseño del sistema, específicamente por una decisión de navegabilidad entre funciones.





(Ej.: “Registrar Préstamo de un Video”. Y estando en dicho caso de uso tienes a la vista en la pantalla, y decides utilizar, un botón que te permitiría iniciar otro caso de uso que tiene poco o nada que ver con el objetivo del caso de uso inicial (digamos, “Consultar Promociones”). Esto no debería de mostrarse como una relación entre estos dos casos de uso en el modelo.

Una de las razones por las cuales deberías de considerar el uso de este tipo de relaciones, es porque identificas que hay pasos que son iguales en dos o más casos de uso.

### Herramientas para la ingeniería de requerimientos

Están orientadas a soportar algunas de las actividades (o todas) de la ingeniería de requerimientos, pero principalmente se orientan a permitir la administración de los requerimientos en relación a estas actividades.

#### RequisitePro

Es una herramienta de administración de requerimientos que puede ser utilizada para administrar los requerimientos de los proyectos, el cual promueve la comunicación y colaboración entre los miembros del equipo de trabajo con el fin de reducir los riesgos del proyecto.

Esta herramienta permite organizar y priorizar los requerimientos, así como hacer la trazabilidad entre estos. Además registra un historial de cambios para cada uno de los requerimientos. Pertenece a IBM y soporta la herramienta Microsoft Word para la creación de los requerimientos. Y para tener en cuenta las métricas de los requerimientos soporta Microsoft Excel.



También provee la opción de crear un proyecto nuevo teniendo en cuenta que metodología se quiere o se está utilizando para el desarrollo del proyecto.

### **Rational DOORS**

DOORS (Dynamic Object Oriented Requirements System) es un software para la gestión de requisitos creado por la empresa sueca Telelogic, posteriormente adquirido por IBM.

Básicamente es una base de datos que permite el almacenamiento estructurado y la administración de requisitos (objetos), que cuenta con atributos, tanto libres como a elegir en un menú. Para diferenciar los requisitos DOORS otorga a cada objeto un número identificador inequívoco. Para permitir el seguimiento de los requisitos durante el transcurso del proyecto se pueden enlazar unos objetos con otros. DOORS se basa en una base de datos propietaria.

Cuenta con una serie de herramientas para el análisis y permite automatizar diferentes pasos en el uso de la herramienta con la ayuda de un lenguaje de script llamado DXL (DOORS eXtended Language).

Los datos almacenados en DOORS se pueden utilizar para crear un modelo basado en UML 2.0.

Es una aplicación de uso habitual en la industria aeronáutica, de automovilística y ferroviaria.

### **CALIBER RM (Borland Caliber Analyst)**

Se trata de un producto que está compuesto por dos aplicaciones desarrolladas por la compañía Borland.

Por un lado está el Caliber DefineIT (la última de las herramientas en cuanto a fecha de lanzamiento) que permite definir los requisitos del sistema así como capturar los diferentes escenarios de negocio a través de diferentes herramientas visuales; es necesario señalar que este software es compatible con gran número de herramientas existentes en el mercado.

Por el otro está Caliber RM que nos permite gestionar dichos requisitos durante el ciclo de vida del producto, si bien no ayudaba al usuario a visualizar los requerimientos y por lo tanto no resultaba tan efectiva como ellos demandaban.

El paquete que incluye ambas aplicaciones nos permitirá realizar las siguientes tareas: representar y especificar los escenarios de manera visual, permitiendo el uso de un lenguaje común; generar diagramas de casos de pruebas y UML, mejorando tanto la velocidad como la exactitud de la definición de los requisitos; rastrear los requisitos



software durante el ciclo de vida del proyecto, respondiendo de manera rápida a cualquier cambio que se produzca.

La compañía Seilevel Inc., una de las más fuertes en cuanto a los servicios relacionados con los requisitos del software, ha seleccionado esta herramienta como la mejor de este tipo. Según palabras de un directivo de la compañía ven características únicas en esta herramienta así como una experiencia de usuario excelente y una oportunidad para mejorar el trabajo de sus clientes en cuanto al análisis y gestión de requisitos se refiere.

### CASE Spec

Esta herramienta está desarrollada por la empresa Goda Software, siendo esta una aplicación comercial de uso exclusivo para el sistema operativo Windows. Las principales características que avalan a esta herramienta son las siguientes:

- Especificación: posibilidad de realizar las especificaciones tanto con las técnicas tradicionales como con los diagramas de casos de uso. Además, nos permite crear diagramas UML y de flujo de datos.
- Seguimiento de los requisitos: a través del uso combinado de un procesador de textos y una hoja de cálculo, el usuario será capaz de realizar el seguimiento de los requisitos así como hacer un informe acerca de los mismos.
- Capacidad de rastreo: mediante la existencia de una matriz se representan de manera sencilla las diferentes relaciones existentes entre los requisitos definidos y otra serie de elementos incidentes en el proyecto.
- Capacidad de importar y exportar archivos.
- Generación automática de la documentación del proyecto así como posibilidad de realizar amplios informes.

### IRQA 4

Herramienta desarrollada por Visure y que tiene la meta de servir como aplicación para proporcionar un soporte integral en la ingeniería de requisitos de un proyecto de informática.

A parte de incluir las tareas más básicas de la ingeniería de requisitos (captura, análisis, modelado, organización y seguimiento), esta aplicación dispone de las siguientes características:

- Reutilización de requisitos: permite que los requisitos definidos en un proyecto puedan ser utilizados en otros proyectos realizados por la organización, a través



del uso de librerías. De esta manera se consigue ofertar una pequeña ventaja a la hora de realizar líneas de productos.

- Vista documental: esta nueva opción ofrece un agrupamiento de los requisitos que permite al usuario observar una diferenciación clara entre los mismos así como facilitar toda labor relacionada con estos.
- Ingeniería de requisitos: además de la gestión de los requisitos, esta aplicación proporciona funcionalidades relacionadas con la ingeniería de requisitos, lo que permite centralizar en una sola herramienta todas las actividades relacionadas con los requisitos (incluyendo las pruebas de validación y aceptación).
- Al ser esta una herramienta integrada, se ofrece al usuario la libertad de seleccionar aquellas otras aplicaciones más adecuadas para la realización de otras tareas relacionadas con el ciclo de vida de un proyecto, lo que hace que no se dependa de un solo proveedor de aplicaciones.

### **Tiger Pro**

Herramienta shareware desarrollada para facilitar al usuario la tarea de redactar los requerimientos de un proyecto. Este aplicativo es capaz de solucionar algunos de los defectos que aparecen a la hora de definir los requisitos de un programa. También ayuda al usuario a aclarar algunos de los requerimientos desde el punto de vista de las pruebas a realizar, señalando aquellos requerimientos cuya verificación pueda resultar complicada.

La herramienta, que se va actualizando con el paso del tiempo, permite exportar el trabajo realizado en archivos bajo el formato CSV. Los usuarios que utilicen esta herramienta podrán trabajar en los requisitos tomando como referencia los siguientes conceptos: palabras claves relacionadas con el mismo (hasta 3 palabras para cada requisito), criterio de aceptación del requisito, seguimiento del mismo (tanto hacia la fuente como hacia otros lugares), prioridad del requerimiento, riesgo que trae consigo el requisito y coste del mismo. Además, a la hora de realizar los informes correspondientes, la herramienta nos proporcionará la opción de redactar los mismos en forma textual o bien nos presentará la información de forma gráfica.

### **GatherSpace**

Esta herramienta de definición y gestión de requisitos utiliza Internet como su lanzadera, ya que no es necesario instalar ningún programa para utilizarla: bastará con crear una cuenta en el sitio web de la misma y comenzar a definir el proyecto que se quiere desarrollar. De esta manera, la aplicación consigue que la colaboración de todos los miembros del grupo de desarrollo sea posible de una manera mucho más eficaz.



Las características más representativas de esta herramienta son las siguientes:

- Creación de una jerarquía de requerimientos: permite crear paquetes funcionales para después relacionarlos con componentes de más alto nivel para después permitir asociar casos de uso más detallados y requisitos del software a dichos componentes.
- Manipular varios proyectos al mismo tiempo, controlando el acceso de los usuarios para que estos puedan ver solo alguno de los proyectos.
- Posibilidad de visualizar la documentación generada a partir de los requisitos en tres formatos diferentes: HTML, PDF y Microsoft Word.
- Además de contar con todas estas opciones, la compañía ha dispuesto un buen sistema de seguridad que protegerá los datos introducidos en la herramienta.

### **RaQuest**

Se trata de la herramienta de gestión de requisitos desarrollada por la empresa Sparx Systems, desarrolladora también de la herramienta de análisis y modelado Enterprise Architect, utilizada en la Escuela.

Las características principales de esta herramienta son las siguientes:

- u **Definición y gestión de los elementos relacionados con los requisitos**, entre los que se encuentran el tipo, el estado, la dificultad del requisito, las relaciones existentes entre diferentes requisitos, etc.
- u **Creación de paquetes** para gestionar de manera más sencilla y completa los requisitos.
- u **Generación de documentación del proyecto** (tanto parcial como total) en los siguientes formatos: HTML, CSV, Word, Excel, RTF

Además de estas características, la herramienta nos ofrece una serie de vistas diferentes, dependiendo de la vista que queramos obtener del proyecto. Estas vistas son: vista del tipo lista (permite ordenar los requisitos, mostrar diferentes listas, filtrar las listas en relación a diferentes palabras y buscar en el proyecto) y vista del tipo árbol (se pueden mostrar los árboles de proyecto y miembro así como mostrar los árboles por el tipo y por el estado).

El cuadro siguiente presenta una serie de características que deberían tener cada una de estas herramientas en el contexto de la ingeniería de requerimientos y expresa cuál de ellas cubre cada una de estas características.



CARACTERÍSTICAS	DOORS	RTM Workshop	Caliber - RM	RequisitePro
Permite la conversión de documentos fuente para cargar los requerimientos a una Base de Datos (BD)	✓	✓	✓	✓
Importa los requerimientos desde tablas de word hacia BD	✓	✓	✓	
Incorpora objetos no textuales como hojas de trabajo de excel e imágenes dentro de BD	✓	✓	✓	
Sincroniza un SRS con el contenido de BD		✓		✓
Define diferentes atributos para diferentes tipos de requerimientos y ajusta atributos para requerimientos individuales	✓	✓	✓	✓
Define directivas para requerimientos	✓	✓		
Notifica a los participantes afectados del proyecto vía e-mail acerca de los cambios en los requerimientos	✓	✓	✓	
Define relaciones de trazabilidad y vínculos entre los requerimientos individuales y entre estos y otros elementos del sistema	✓	✓	✓	✓
Permite definir opciones de usabilidad para requerimientos no funcionales	✓	✓	✓	✓
Incluye elementos para el aprendizaje como tutoriales y proyectos de ejemplo	✓	✓	✓	✓
Se integra con otras herramientas como: prueba, diseño y administración de proyectos	✓	✓	✓	✓
Define usuarios y grupos, con sus respectivos permisos de acceso	✓	✓	✓	✓
Habilita discusiones de requerimientos		✓	✓	✓
Incluye interfaces web para consultas a la BD, discusiones y tal vez la actualización de atributos de requerimientos	✓	✓	✓	✓
Incluye un sistema para desarrollos de sistemas basados en propuestas de cambios	✓	✓		
Incluye una completa documentación de manuales de usuario	✓		✓	✓

## Bibliografía

1. Plantillas de las Clases de Teoría. Mgter. Vallejos, Oscar A.
2. Capítulo VI y VII de: INGENIERÍA DEL SOFTWARE. Séptima edición Ian Sommerville. ISBN: 84-7829-074-5. PEARSON ADDISON WESLEY.
3. Capítulo IV de: INGENIERÍA DEL SOFTWARE. Un enfoque práctico. Séptima edición. Roger S. Pressman. ISBN: 978-607-15-0314-5. MC GRAW HILL
4. PIATTINI, Mario; J. Antonio CalvoManzano; Joaquín Cervera y Luis Fernández, Análisis y diseño detallado de aplicaciones informáticas de gestión, México, coedición Alfa omegaRama, 2004, 728 pp.
5. PFLEEGER, Shari Lawrence, Ingeniería de software, Teoría y práctica, México, Prentice Hall, 2002, 759 pp.
6. PRESSMAN, Roger S., Ingeniería del software, 5a. Edición, México, Mc. GrawHill, 2002, 602 pp.



7. SMITH, Jo Ann, Desarrollo de proyectos con programación orientada a objetos con C++ , México, Thomson, 2001 224 pp

Link

8. <http://www.software-engin.com>