



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®



Especialidad Análisis, diseño y desarrollo de software

Análisis Avanzados de Software

D1.3 Ingeniería de requerimientos

Asesor: M.T.I.C. Leonardo Enriquez

Ingeniero electrónico, sistemas digitales



Estudio de factibilidad

1. Estudio de factibilidad.



Estudios de factibilidad

Un estudio de factibilidad es un breve estudio enfocado que debe realizarse con oportunidad en el proceso de IR. Debe responder tres preguntas clave: a) ¿El sistema contribuye con los objetivos globales de la organización? b) ¿El sistema puede implementarse dentro de la fecha y el presupuesto usando la tecnología actual? c) ¿El sistema puede integrarse con otros sistemas que se utilicen?

Si la respuesta a cualquiera de estas preguntas es negativa, probablemente no sea conveniente continuar con el proyecto.

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



Las actividades del proceso son:

-
- ```
graph TD; 1[1. Descubrimiento de requerimientos] --> 2[2. Clasificación y organización de requerimientos]; 2 --> 3[3. Priorización y negociación de requerimientos]; 3 --> 4[4. Especificación de requerimientos]; 4 --> 1;
```

## Proceso de adquisición y análisis de requerimientos



# Especificación de requerimientos

---

## 3. Especificación de requerimientos

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

Los requerimientos del usuario para un sistema deben describir los requerimientos funcionales y no funcionales, de forma que sean comprensibles para los usuarios del sistema que no cuentan con un conocimiento técnico detallado (Tal como se vio en paginas anteriores).



# Validación de requerimientos

## 4. Validación de requerimientos

Es el proceso de verificar que los requerimientos realmente defina el sistema que quiere el cliente. Dichas comprobaciones incluyen:

- a) **Comprobaciones de validez** Un usuario quizá crea que necesita un sistema para realizar ciertas funciones, sin embargo, con mayor consideración y análisis se logra identificar las funciones adicionales o diferentes que se requieran. Los sistemas tienen diversos **participantes con diferentes necesidades**, y cualquier conjunto de requerimientos es inevitablemente un compromiso a través de la comunidad de participantes.
- a) **Comprobaciones de consistencia** Los requerimientos en el documento **no deben estar en conflicto**, es decir no debe haber restricciones contradictorias o descripciones diferentes de la misma función del sistema.
- a) **Comprobaciones de totalidad** El documento de requerimientos debe incluir **requerimientos que definan todas las funciones** y las restricciones pretendidas por el usuario del sistema.
- a) **Comprobaciones de realismo** Al usar el conocimiento de la tecnología existente, los requerimientos deben **comprobarset para garantizar** que en realidad pueden implementarse. Dichas comprobaciones también tienen que considerar el presupuesto y la fecha para el desarrollo del sistema.
- a) **Verificabilidad** Para reducir el potencial de **disputas entre cliente y contratista**, los requerimientos del sistema deben escribirse siempre de manera que sean verificables. Esto significa que usted debe ser capaz de escribir un conjunto de pruebas que demuestren que el sistema entregado cumpla cada requerimiento especificado.





# Requerimiento funcionales

---

- Se definen como su nombre lo indica las funciones de sistema de un software o sus componentes, estos describen **las entradas, salidas y el comportamiento del software**.
- Son enunciados acerca de **servicios que el sistema debe proveer**, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse el sistema en situaciones específicas. En algunos casos, los requerimientos funcionales también explican lo que no debe hacer el sistema.
- Describen lo que el sistema debe hacer. Los cuales dependen del software a desarrollar, los cuales en su mayor parte se los redactan de una forma abstracta. Pero estos describen con detalle la función de este, sus entradas y salidas, excepciones, etcétera.
- Para un desarrollador de sistemas es natural dar interpretaciones de un requerimiento ambiguo con el fin de simplificar su implementación





# Requerimiento no funcionales

- *Requerimientos **no funcionales*** Son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del sistema.
- Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y la representación de datos que se utiliza en la interface del sistema.

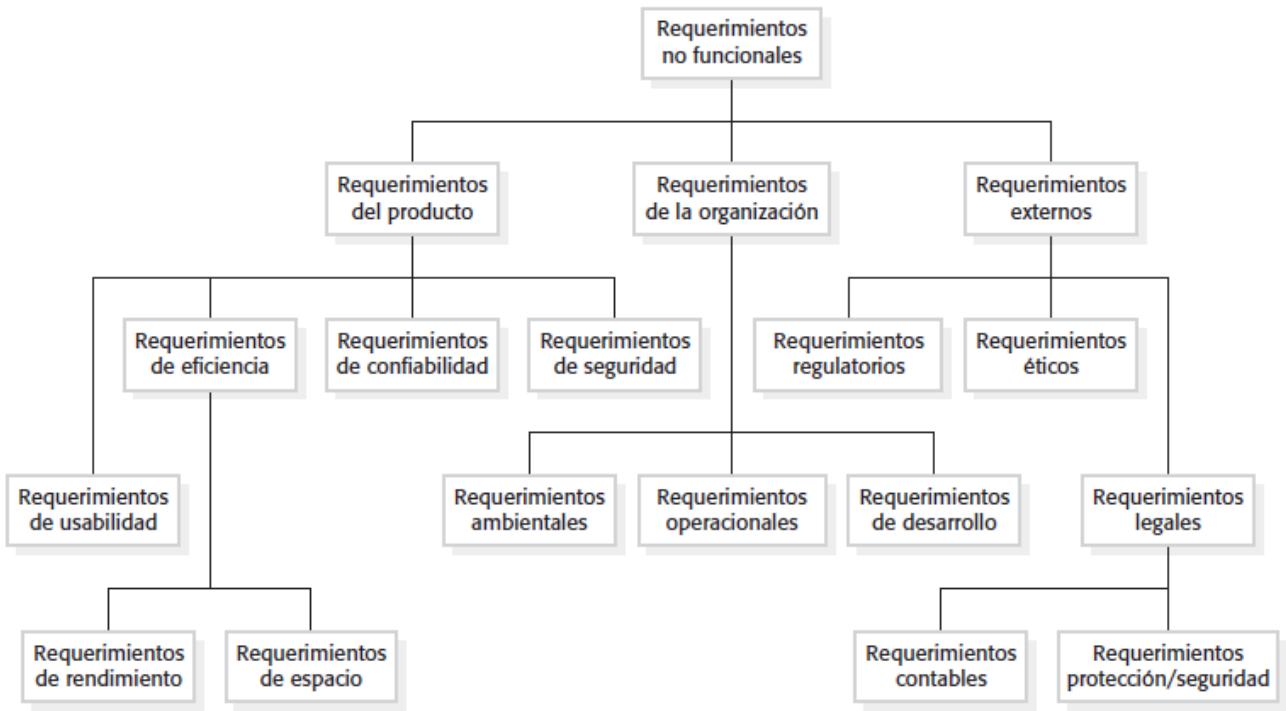
Estos diferentes tipos de requerimientos **se clasifican** de acuerdo con sus implicaciones.

- **Requerimientos del producto.** Especifican el comportamiento del producto; como los requerimientos de desempeño en la rapidez de ejecución del sistema y cuánta memoria se requiere; los de fiabilidad que fijan la tasa de fallas para que el sistema sea aceptable; los de portabilidad y los de usabilidad.
- **Requerimientos organizacionales.** Se derivan de las políticas y procedimientos existentes en la organización del cliente y en la del desarrollador: estándares en los procesos que deben utilizarse; requerimientos de implementación como los lenguajes de programación o el método de diseño a utilizar, y los requerimientos de entrega que especifican cuándo se entregará el producto y su documentación.
- **Requerimientos externos.** Se derivan de los factores externos al sistema y de su proceso de desarrollo. Incluyen los requerimientos de interoperabilidad que definen la manera en que el sistema interactúa con los otros sistemas de la organización; los requerimientos legales que deben seguirse para asegurar que el sistema opere dentro de la ley, y los requerimientos éticos. Estos últimos son impuestos al sistema para asegurar que será aceptado por el usuario.



# Tipos de requerimientos no funcionales

## Tipos de requerimientos no funcionales



Tipos de requerimientos no funcionales

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

Los requerimientos no funcionales, como los requerimientos de fiabilidad, protección y confidencialidad, son en particular importantes para los sistemas fundamentales.



# Tipos de requerimientos no funcionales

---

## Tipos de requerimientos no funcionales

**Comprobabilidad:** Grado en que un sistema, software o servicio de TI permite y facilita que sea probado en un determinado contexto.

•**Disponibilidad:** Corresponde al tiempo total en que un sistema puede ser usado en un período determinado. También puede definirse el grado en que un sistema está en un estado operable definido cada vez que se necesite.

•**Extensibilidad:** Grado en que la implementación del sistema toma en consideración y facilita su crecimiento en el futuro.

•**Escalabilidad:** Capacidad de un sistema o servicio de TI de manejar una creciente carga de trabajo, por ejemplo mayor número de conexiones o usuarios. No debe confundirse con extensibilidad, que mide la capacidad del sistema de crecer en funcionalidades.

•**Mantenibilidad:** Mide la facilidad con que puede darse mantenimiento al producto (en este caso al software o servicio de TI), con la finalidad de: Desarrollar nuevos requerimientos, Aislar los defectos y sus causas, corregir estos defectos y atender las demandas del entorno cambiante.

•**Seguridad:** Grado de protección de los datos, software y plataforma de tecnología de posibles pérdidas, actividades no permitidas o uso para propósitos no establecidos previamente.

•**Usabilidad:** Definido como la facilidad de uso y aprendizaje de un Sistema, Software o Servicio de Tecnología de Información.



# Requisitos comunes

## Requisitos funcionales

| ID   | Requisitos comunes                                         |
|------|------------------------------------------------------------|
| RF1  | Autenticación                                              |
| RF2  | Autorización                                               |
| RF4  | Asignación de funcionalidades                              |
| RF5  | Envío y recepción de archivos                              |
| RF6  | Operaciones CRUD (Create, Read, Update, Delete)            |
| RF7  | Importación de data desde archivos externos                |
| RF8  | Exportación de datos a archivos externos                   |
| RF9  | Envío y recepción de datos de acuerdo a un modelo definido |
| RF10 | Interconexión entre componentes - servicios                |
| RF11 | Interconexión entre componentes - puertos                  |
| RF12 | Abstracción de sistemas de bases de datos relacionales     |

## Requisitos No funcionales

| ID   | Requisitos comunes                              |
|------|-------------------------------------------------|
| RNF1 | Disponibilidad                                  |
| RNF2 | Tiempos de respuesta rápidos (alto rendimiento) |
| RNF3 | Integridad de la información                    |
| RNF4 | Seguridad                                       |
| RNF5 | Mantenibilidad                                  |
| RNF6 | Interoperabilidad                               |



# Ejemplos de requerimiento funcionales de proceso o área de negocio

---

- El sistema enviará un correo electrónico cuando se registre alguna de las siguientes transacciones: pedido de venta de cliente, despacho de mercancía al cliente, emisión de factura a cliente y registro de pago de cliente.
- Se permitirá el registro de pedidos de compra con datos obligatorios incompletos, los cuales podrán completarse posteriormente modificando el pedido. Antes de poder aprobarse los datos del pedido deben estar completos.
- Al aprobar un pedido, la solicitud pasará al siguiente paso del flujo de trabajo (workflow) de aprobación configurado en el sistema.
- El sistema permitirá a los usuarios autorizados el ingresar planes y cronogramas de **proyecto**.
- El sistema permitirá aprobar, cambiar o actualizar planes y cronogramas de **proyecto**.
- El sistema permitirá el envío automatizado de cartas de entrega de órdenes directamente al almacén.
- A cada orden se le asignará un identificador único, que será utilizado para identificarla en todos los procesos subsecuentes que se realicen sobre esta.
- Al ingresar ordenes de entrega, toda orden de entrega estará asociada a un pedido de venta.
- La facturación de pedidos de venta se realizara en lotes, por medio de una pantalla de pedidos pendientes de facturación, la cual mostrará los pedidos no facturados. Una vez facturados los pedidos no se mostrarán en esta lista.



# Ejemplos de requerimiento funcionales de proceso o área de negocio

---

- El sistema también permitirá el registro de facturas manuales no asociadas a pedidos, sin embargo, estas requerirán autorización por parte del grupo de Gerentes antes de ser contabilizadas.
- El proceso de compras en el sistema abarcará los siguientes pasos y transacciones: Ingreso de la requisición, emisión de la solicitud de cotización y emisión de la orden de compra.
- Los elementos de la solicitud de cotización serán los mismos de la requisición asociada, al igual que los de la orden de compra. El sistema permitirá la emisión de solicitudes de cotización y órdenes de compra parciales.
- La contabilización de transacciones de facturas de venta y facturas de compra podrá configurarse para realizarse de forma automatizada a su registro, o manualmente en lotes (Proceso Batch).
- El software debe poder emitir los siguientes estados financieros: Balance general, Estado de ganancias y pérdidas, Estado de flujos de efectivo. Además, debe poder emitir un listado de mayor general y mayor analítico.
- Los pedidos de compra que excedan los montos establecidos en el flujo de liberaciones de pedidos configurados, deberán pasar por las aprobaciones establecidas en dicho flujo de aprobación.



# Ejemplos de requerimientos funcionales de interfaz grafica

- La solución validara automáticamente el cliente asociado a una orden con el sistema de gestión de contactos.
- El campo de monto acepta únicamente valores numéricos con dos decimales.
- El campo fecha de transacción acepta únicamente fechas anteriores al día de hoy (día actual).
- El campo nombre acepta caracteres alfabéticos únicamente.
- El campo dirección acepta caracteres alfabéticos, numéricos y especiales.
- El campo país consistirá en una lista de preselección. El país asociado a una dirección debe ser previamente registrado en el sistema.
- El campo estado, provincia o departamento consistirá en una lista de preselección. A los usuarios se les presentará únicamente los estados asociados al país seleccionado previamente. El departamento o provincia a seleccionar deberá ser registrado en la funcionalidad correspondiente.
- El campo material de elemento de la pantalla de requisiciones de compra será una lista de preselección, que mostrará únicamente los materiales registrados en el maestro de materiales.
- El campo fecha contable acepta únicamente fechas que correspondan con periodos contables que estén abiertos en el sistema.
- La pantalla de registro de pago puede imprimir los datos en pantalla a la impresora.
- Se mostrará el nombre, tamaño total, espacio disponible y formato de un pen drive o flash drive conectado al puerto USB del computador.



- El sistema controlará el acceso y lo permitirá solamente a usuarios autorizados.
- La base de datos será implementada con trazas de auditoría.
- Las hojas de cálculo asegurarán los datos usando firmas electrónicas.
- El sistema permitirá elaborar y emitir el reporte regulatorio XX, según los requerimientos establecidos en el reglamento y ley aplicable.
- Los libros de venta y de compras serán emitidos en el formato establecido por las autoridades tributarias de dicha materia.





# Ejemplos de requerimiento de seguridad

---

- El sistema controlará el acceso y lo permitirá solamente a usuarios autorizados. Los usuarios deben ingresar al sistema con un nombre de usuario y contraseña.
- El sistema enviará una alerta al administrador del sistema cuando ocurra alguno de los siguientes eventos: Registro de nueva cuenta, ingreso al sistema por parte del cliente, 2 o más intentos fallidos en el ingreso de la contraseña de usuario y cambio de contraseña de usuario.
- Los integrantes del grupo de usuarios de analistas pueden ingresar solicitudes pero no pueden aprobarlas o borrarlas.
- Los integrantes del grupo de usuarios de gerentes pueden ingresar y aprobar solicitudes, pero no pueden borrarlas.
- Los integrantes del grupo de usuario de administradores no pueden ingresar o aprobar solicitudes, pero si pueden borrarlas.
- Cualquier intercambio de datos vía internet que realice el software se realizará por medio del protocolo encriptado https.



# Ejemplos de requerimiento de interfaces externas (hardware y software)

---

- El software podrá ser utilizado en los sistemas operativos Windows, Linux y OSX.
- La aplicación debe poder utilizarse sin necesidad de instalar ningún software adicional además de un navegador web.
- La aplicación debe poder utilizarse con los navegadores web Chrome, Firefox e Internet Explorer.



# Ejemplos de requerimiento no funcionales de producto: Eficiencia

---

- El sistema debe ser capaz de procesar N transacciones por segundo. Esto se medirá por medio de la herramienta SoapUI aplicada al Software Testing de servicios web.
- Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.
- El sistema debe ser capaz de operar adecuadamente con hasta 100.000 usuarios con sesiones concurrentes.
- Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.



# Ejemplos de requerimiento no funcionales de producto: Seguridad lógica y de datos

---

- Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.
- El nuevo sistema debe desarrollarse aplicando **patrones y recomendaciones de programación que incrementen la seguridad de datos.**
- Todos los sistemas deben respaldarse cada 24 horas. Los respaldos deben ser almacenados en una localidad segura ubicada en un edificio distinto al que reside el sistema.
- Todas las comunicaciones externas entre servidores de datos, aplicación y cliente del sistema deben estar encriptadas utilizando el algoritmo RSA.
- Si se identifican ataques de seguridad o brecha del sistema, el mismo no continuará operando hasta ser desbloqueado por un administrador de seguridad.



- El sistema no continuará operando si la temperatura externa es menor a 4 grados Celsius.
- El sistema no continuará operando en caso de fuego. (Ej. Un ascensor).



- El sistema debe tener una disponibilidad del 99,99% de las veces en que un usuario intente accederlo.
- El tiempo para iniciar o reiniciar el sistema no podrá ser mayor a 5 minutos.
- La tasa de tiempos de falla del sistema no podrá ser mayor al 0,5% del tiempo de operación total.
- El promedio de duración de fallas no podrá ser mayor a 15 minutos.
- La probabilidad de falla del Sistema no podrá ser mayor a 0,05.



# Ejemplos de requerimiento no funcionales: Mantenibilidad

---

- El sistema debe tener una disponibilidad del 99,99% de las veces en que un usuario intente accederlo.
- El tiempo para iniciar o reiniciar el sistema no podrá ser mayor a 5 minutos.
- La tasa de tiempos de falla del sistema no podrá ser mayor al 0,5% del tiempo de operación total.
- El promedio de duración de fallas no podrá ser mayor a 15 minutos.
- La probabilidad de falla del Sistema no podrá ser mayor a 0,05.



- El sistema debe tener una disponibilidad del 99,99% de las veces en que un usuario intente accederlo.
- El tiempo para iniciar o reiniciar el sistema no podrá ser mayor a 5 minutos.
- La tasa de tiempos de falla del sistema no podrá ser mayor al 0,5% del tiempo de operación total.
- El promedio de duración de fallas no podrá ser mayor a 15 minutos.
- La probabilidad de falla del Sistema no podrá ser mayor a 0,05.





# Ejemplos de requerimiento no organizacionales

- El procedimiento de desarrollo de software a usar debe estar definido explícitamente (en manuales de procedimientos) y debe cumplir con los estándares ISO 9000.
- La metodología de desarrollo de software será **Behaviour Driven Development (BDD) apoyada en Cucumber.**
- El sistema debe ser desarrollado utilizando las herramientas CASE XYZ.
- El proceso de desarrollo se gestionará por medio de una determinada **herramienta web para gestionar el proceso de desarrollo de software.**
- Debe especificarse un plan de recuperación ante desastres para el sistema a ser desarrollado.
- Cada dos semanas deberán producirse reportes gerenciales en los cuales se muestre el esfuerzo invertido en cada uno de los componentes del nuevo sistema.
- Las pruebas de software se gestionaran con una **herramienta de gestión de software testing.**
- Las pruebas de software se ejecutarán utilizando **Selenium y Ruby** como herramienta y lenguaje Scripting para automatización de software testing.



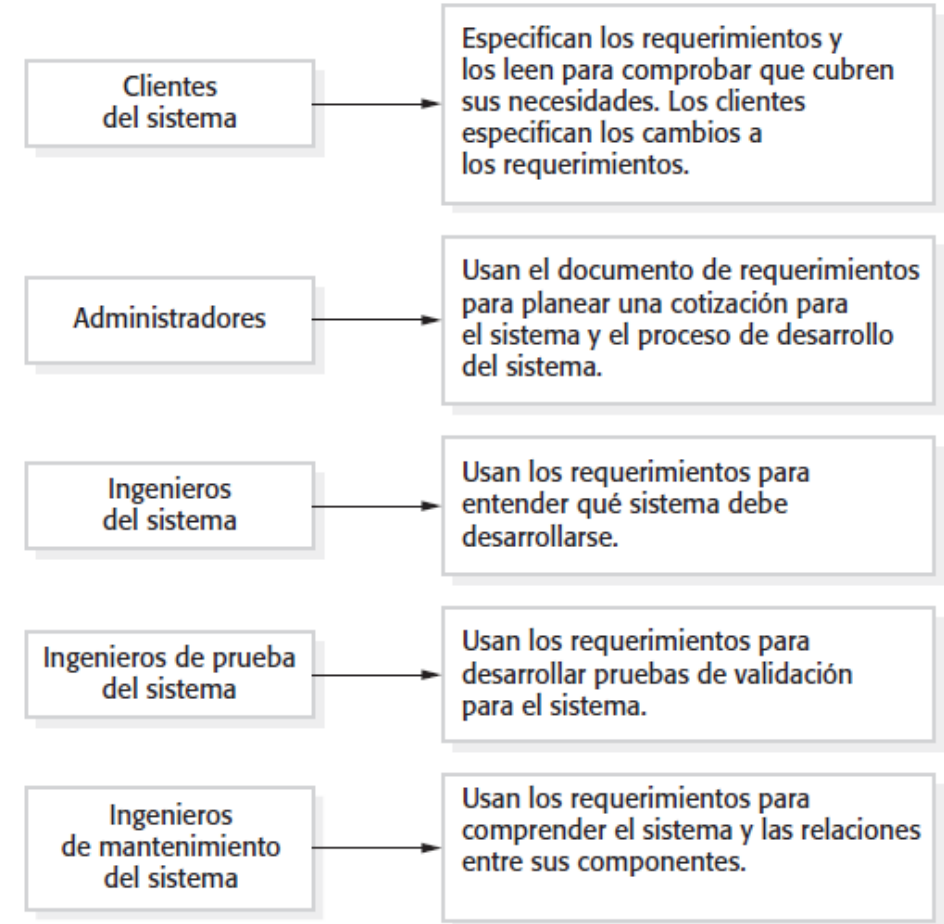
# Documento de requerimiento de software

El **documento de requerimientos de software** (llamado algunas veces especificación de requerimientos de software o SRS) es un comunicado oficial de lo que deben implementar los desarrolladores del sistema.

El documento incluye:

- **Requerimientos del usuario** para un sistema,
- **Especificación detallada de los requerimientos** del sistema.

En ocasiones, los requerimientos del usuario y del sistema se integran en una sola descripción. En otros casos, los requerimientos del usuario se definen en una introducción a la especificación de requerimientos del sistema. Si hay un gran número de requerimientos, los requerimientos del sistema detallados podrían presentarse en un documento aparte.



Usuarios de un documento de requerimientos



# Contenido de un documento de requerimiento de software

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

Formas de escribir una especificación de requerimientos del sistema:

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



# Aspectos a contemplar en una especificación de requerimientos

Según el estándar internacional de Especificación de Requerimientos IEEE830, los documentos de definición y especificación de requerimientos deben contemplar los siguientes aspectos resumidos por [Pfleeger, 2002] como se indica a continuación:

- **Ambiente físico**
  - ¿Dónde esta el equipo que el sistema necesita para funcionar?
  - ¿Existe una localización o varias?
  - ¿Hay restricciones ambientales como temperatura, humedad o interferencia magnética?
- **Interfaces**
  - ¿La entrada proviene de uno o más sistemas?
  - ¿La salida va a uno o más sistemas?
  - ¿Existe una manera preestablecida en que deben formatearse los datos?
- **Usuarios y factores humanos**
  - ¿Quien usará el sistema?
  - ¿Habrá varios tipos de usuario?
  - ¿Cuál es el nivel de habilidad de cada tipo de usuario?
  - ¿Qué clase de entrenamiento requerirá cada tipo de usuario?
  - ¿Cuán fácil le será al usuario comprender y utilizar el sistema?
  - ¿Cuán difícil le resultará al usuario hacer uso indebido del sistema?
- **Funcionalidad**
  - ¿Qué hará el sistema? - ¿Cuándo lo hará?
  - ¿Existen varios modos de operación?
  - ¿Cómo y cuando puede cambiarse o mejorarse un sistema?
  - ¿Existen restricciones de la velocidad de ejecución, tiempo de respuesta o rendimiento?
- **Documentación**
  - ¿Cuánta documentación se requiere?
  - ¿Debe estar en línea, en papel o en ambos?
  - ¿A que audiencia está orientado cada tipo de información?
- **Datos**
  - ¿Cuál será el formato de los datos, tanto para la entrada como para la salida?
  - ¿Cuán a menudo serán recibidos o enviados?
  - ¿Cuán exactos deben ser?
  - ¿Con qué grado de precisión deben hacerse los cálculos?
  - ¿Cuántos datos fluyen a través del sistema?
  - ¿Debe retenerse algún dato por algún período de tiempo?



# Aspectos a contemplar en una especificación de requerimientos

- **Recursos**

- ¿Qué recursos materiales, personales o de otro tipo se requieren para construir, utilizar y mantener el sistema?
- ¿Qué habilidades deben tener los desarrolladores?
- ¿Cuánto espacio físico será ocupado por el sistema?
- ¿Cuáles son los requerimientos de energía, calefacción o acondicionamiento de aire?
- ¿Existe un cronograma prescrito para el desarrollo?
- ¿Existe un límite sobre la cantidad de dinero a gastar en el desarrollo o en hardware y software?

- **Seguridad**

- ¿Debe controlarse el acceso al sistema o a la información?
- ¿Cómo se podrán aislar los datos de un usuario de los de otros?
- ¿Cómo podrán aislarse los programas de usuario de los otros programas y del sistema operativo?
- ¿Con qué frecuencia deben hacerse copias de respaldo?
- ¿Las copias de respaldo deben almacenarse en un lugar diferente?
- ¿Deben tomarse precauciones contra el fuego, el daño provocado por agua o el robo?

- **Aseguramiento de la calidad**

- ¿Cuáles son los requerimientos para la confiabilidad, disponibilidad, facilidad de mantenimiento, seguridad y demás atributos de calidad?
- ¿Cómo deben demostrarse las características del sistema a terceros?
- ¿El sistema debe detectar y aislar defectos?
- ¿Cuál es el promedio de tiempo prescrito entre fallas?
- ¿Existe un tiempo máximo permitido para la recuperación del sistema después de una falla?
- ¿El mantenimiento corregirá los errores, o incluirá también el mejoramiento del sistema?
- ¿Qué medidas de eficiencia se aplicarán al uso de recursos y al tiempo de respuesta?
- ¿Cuán fácil debe ser mover el sistema de una ubicación a otra o de un tipo de computadora a otro?





Karl Wieggers [Wie03], de la empresa Process Impact Inc., desarrolló un formato útil (disponible en [www.processimpact.com/process\\_assets/srs\\_template.doc](http://www.processimpact.com/process_assets/srs_template.doc)) que sirve como guía para aquellos que deben crear una ERS completa. Su contenido normal es el siguiente:

## Tabla de contenido

### Revisión de la historia

## 1. Introducción

- 1.1 Propósito
- 1.2 Convenciones del documento
- 1.3 Audiencia objetivo y sugerencias de lectura
- 1.4 Alcance del proyecto
- 1.5 Referencias

## 2. Descripción general

- ## 2.1 Perspectiva del producto

- 2.2 Características del producto
- 2.3 Clases y características del usuario
- 2.4 Ambiente de operación
- 2.5 Restricciones de diseño e implementación
- 2.6 Documentación para el usuario
- 2.7 Suposiciones y dependencias

### 3. Características del sistema

- 3.1 Característica 1 del sistema
- 3.2 Característica 2 del sistema (y así sucesivamente)

#### 4. Requerimientos de la interfaz externa

- 4.1 Interfaces de usuario
- 4.2 Interfaces del hardware
- 4.3 Interfaces del software
- 4.4 Interfaces de las comunicaciones

## 5. Otros requerimientos no funcionales

- 5.1 Requerimientos de desempeño
- 5.2 Requerimientos de seguridad
- 5.3 Requerimientos de estabilidad
- 5.4 Atributos de calidad del software

## 6. Otros requerimientos

## Apéndice A: Glosario

## Apéndice B: Modelos de análisis

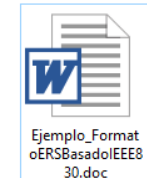
## Apéndice C: Lista de conceptos

Puede obtenerse una descripción detallada de cada ERS si se descarga el formato desde la URL mencionada antes.

## INFORMACIÓN

## Estructura de un documento de requerimientos

Formato ERS-IEE830





# Listado de verificación para validad requerimientos



## *Listado de verificación para validar requerimientos*

Con frecuencia es útil analizar cada requerimiento en comparación con preguntas de verificación. A continuación se presentan algunas:

- ¿Los requerimientos están enunciados con claridad? ¿Podrían interpretarse mal?
- ¿Está identificada la fuente del requerimiento (por ejemplo, una persona, reglamento o documento)? ¿Se ha estudiado el planteamiento final del requerimiento en comparación con la fuente original?
- ¿El requerimiento está acotado en términos cuantitativos?
- ¿Qué otros requerimientos se relacionan con éste? ¿Están comparados con claridad por medio de una matriz de referencia cruzada u otro mecanismo?
- ¿El requerimiento viola algunas restricciones del dominio?
- ¿Puede someterse a prueba el requerimiento? Si es así, ¿es posible especificar las pruebas (en ocasiones se denominan criterios de validación) para ensayar el requerimiento?
- ¿Puede rastrearse el requerimiento hasta cualquier modelo del sistema que se haya creado?
- ¿Es posible seguir el requerimiento hasta los objetivos del sistema o producto?
- ¿La especificación está estructurada en forma que lleva a entenderlo con facilidad, con referencias y traducción fáciles a productos del trabajo más técnicos?
- ¿Se ha creado un índice para la especificación?
- ¿Están enunciadas con claridad las asociaciones de los requerimientos con las características de rendimiento, comportamiento y operación? ¿Cuáles requerimientos parecen ser implícitos?

## INFORMACIÓN



# Bibliografía

---

Pressman, R. S. (2010). Ingeniería de Software, Un enfoque practico Séptima Edición. Ciudad de México: Mc Graw Hill.

Sommerville. (2011). Ingeniería de Software 9 Edición. Estado de México: Pearson.

UNID Universidad Interamericana para el desarrollo. (2018). Ingeniería de software. Ciudad de México: UNID.