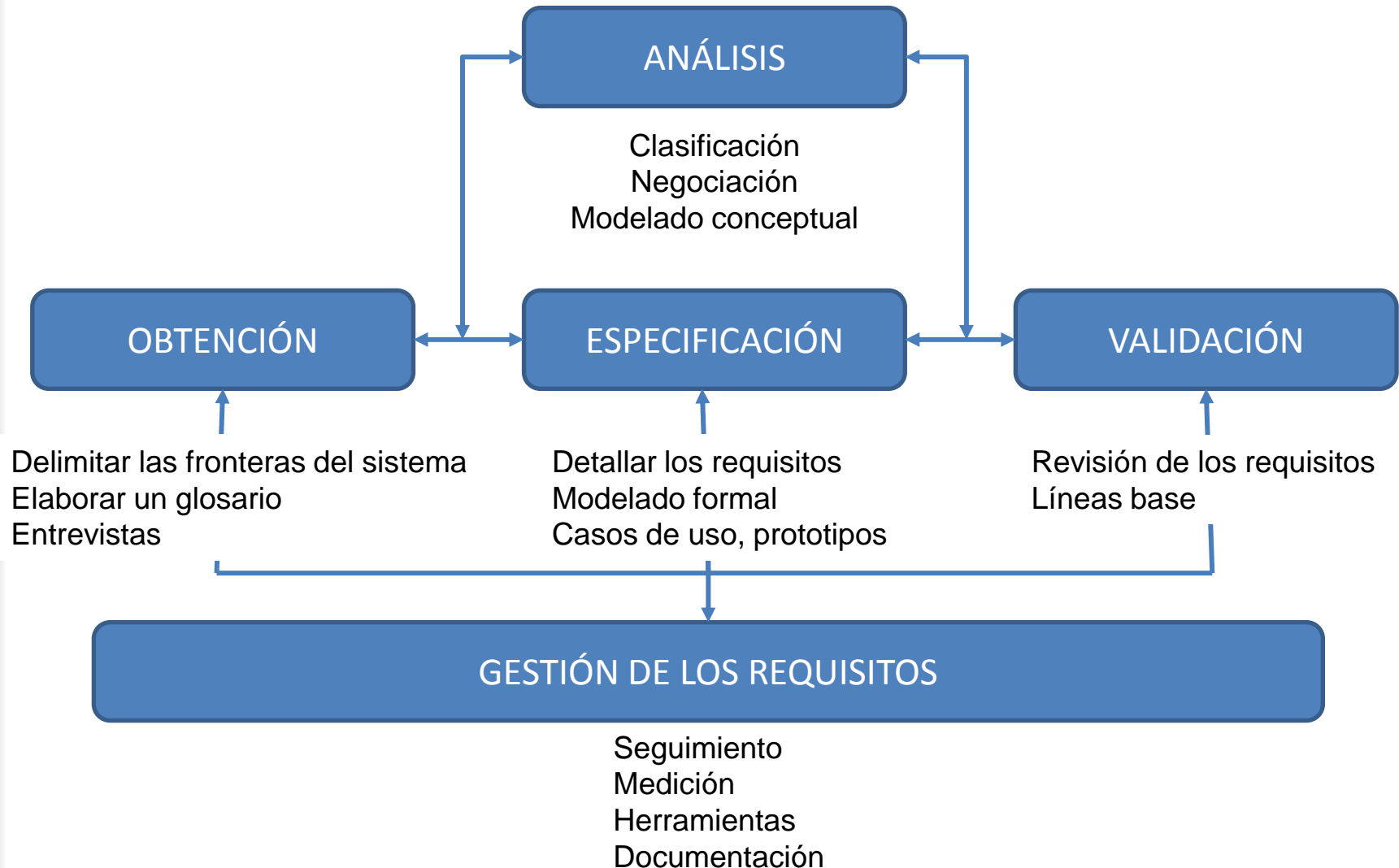


## **2.2 Análisis y especificación de requisitos**

# Indice

- Ingeniería de requisitos
- Análisis de requisitos
  - Requisitos no funcionales
  - Requisitos funcionales
- Especificación de requisitos
- Documento de requisitos
- Organización de requisitos
- Estructura de una ERS
  - Características de una ERS
- Validación de requisitos
- Gestión de requisitos

# Ingeniería de requisitos



# Análisis de requisitos

- Es una actividad situada entre dos actividades extremadamente importantes: la obtención de requisitos, donde se identifican los requisitos y la especificación, donde los requisitos se plasman en un documento
- El **objetivo principal** del análisis de requisitos es **asegurar la calidad** de los requisitos obtenidos, antes de que formen parte del documento de especificación

# Análisis de requisitos

- Para asegurar la calidad de los requisitos éstos deben cumplir ciertas características:
  1. **Independencia del diseño.** No deben indicar cómo debe implementarse el sistema
  2. **Redundancia.** Dos o más requisitos no deben hacer referencia al mismo aspecto del SW
  3. **Concisión.** Deben expresarse de la forma más simple posible
  4. **Realizabilidad.** Se puede implementar con la tecnología disponible
  5. **Internamente consistente.** No existen contradicciones entre requisitos

# Análisis de requisitos

- 6. **Externamente consistente.** No existen contradicciones entre requisitos y documentos externos como políticas u objetivos de empresa
- 7. **Ambigüedad.** No deben haber requisitos que puedan tener distintas interpretaciones
- 8. **Verificabilidad.** Deben existir pruebas para determinar su existencia en el producto final

# Análisis de requisitos

- Objetivos secundarios:
  1. Definir los límites del sistema (alcance) y su interacción con el entorno
  2. Transformar requisitos de usuario en requisitos del SW
    - Ejemplo:
    - RU: El sistema deberá generar facturas
    - RSW: El sistema deberá generar facturas
      - Las facturas se generarán mensualmente
      - El formato de la factura será el indicado en el apéndice X
      - Los pedidos facturados serán marcados como “Facturado”, para evitar su refacturación posterior
      - Por cada factura generada, se creará un registro de cobro en estado “pendiente”

# Análisis de requisitos

## 3. Anotar requisitos. Asignarles ciertos valores

- Identificación: identificador único
- Versión: facilitar el control de versiones, trazabilidad...

Además, también se puede añadir otro tipo de información:

- Importancia: permite seleccionar los requisitos más importantes
- Prioridad: cuánto de necesario cree el cliente que es un requisito
- Estabilidad: probabilidad de que el requisito cambie en el tiempo



# Análisis de requisitos

- Clasificación de los requisitos
  - Establecer un **conjunto de categorías** y situar cada requisito en ellas
  - Existen diferentes **criterios de clasificación** de requisitos
    - Funcionales vs no funcionales
    - Proceso vs producto
    - Prioridad

# Análisis de requisitos

## Requisitos funcionales (capacidades)

- Son declaraciones de los servicios que debe proporcionar el sistema, la forma en que debe reaccionar a las entradas y cómo se debe comportar en situaciones particulares
- Especifica una **función** que un sistema o componente de un sistema debe ser capaz de llevar a cabo
- También pueden declarar explícitamente lo que el sistema no debe hacer

## Requisitos no funcionales (restricciones)

- Son restricciones de los servicios o funciones ofrecidos por el sistema
- Especifican **aspectos técnicos** que debe incluir el sistema
- Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares
- Normalmente se aplican al sistema en su totalidad

# Requisitos no funcionales

- Requisitos no relacionados directamente con la funcionalidad del sistema
- Pueden estar relacionados con propiedades emergentes del sistema: fiabilidad, tiempo de respuesta, capacidad de almacenamiento, escalabilidad, etc
- Pueden describir restricciones al producto a desarrollar
- Pueden describir restricciones externas del sistema
- Definen las cualidades globales que el sistema ha de exhibir
- Suelen hacer referencia al sistema considerado de forma global
- Suelen ser requisitos más críticos que los requisitos funcionales
- Suelen ser difíciles de verificar

# Requisitos no funcionales

- Los requisitos no funcionales surgen de las necesidades del usuario, debido a restricciones en el presupuesto, a las políticas de la organización, a la necesidad de interoperabilidad con otros sistemas, etc
- Los clientes o usuarios establecen requisitos no funcionales como **metas generales**, tales como: **facilidad de uso**, capacidad para **recuperarse de los fallos** o la **respuesta rápida al usuario**

# Requisitos no funcionales

- **Meta del sistema (definido por el usuario)**
  - Debe ser fácil para los controladores experimentados utilizar el sistema y se debe organizar de tal modo que se minimicen los errores de usuario
- **Requisito no funcional verificable**
  - Después de una formación de 2 horas, a los controladores experimentados les deberá ser posible utilizar todas las funciones del sistema. Después de esta formación, la media de errores cometidos por los usuarios experimentados no excederá de 2 por día

# Requisitos no funcionales

- Clasificación de los requisitos no funcionales [Sommerville, 2005]
- **Requisitos de producto**
  - Especifican el comportamiento del producto
    - Tiempo de respuesta, memoria requerida, fiabilidad, portabilidad, usabilidad...
- **Requisitos organizacionales**
  - Se derivan de las políticas y procedimientos existentes en la organización del cliente y en la del desarrollador
    - Estándares de proceso, herramientas a usar, métodos de diseño, estándares de documentación...

# Requisitos no funcionales

- **Requisitos externos**

- Factores externos al sistema y de su proceso de desarrollo
  - Interoperabilidad con otros sistemas, éticos, legislativos, privacidad, seguridad...

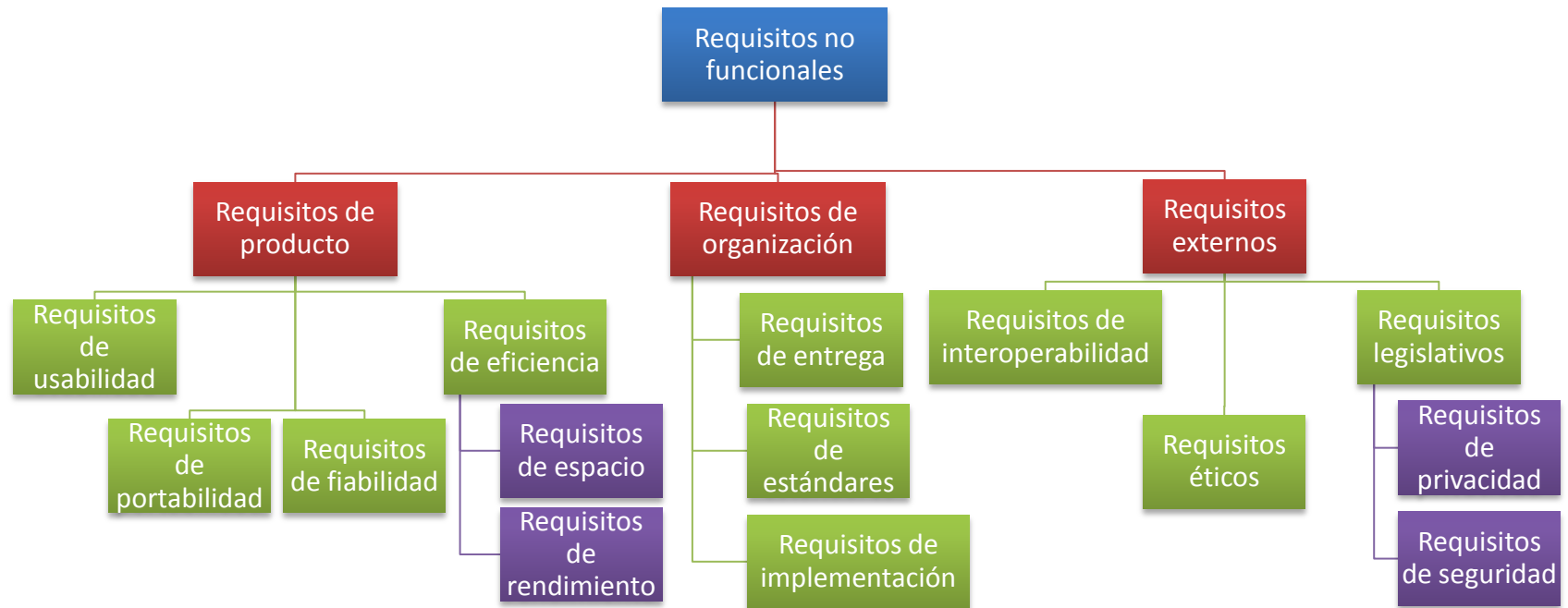


# Requisitos no funcionales

- *“El máximo espacio de almacenamiento ocupado por el sistema debe ser de 8 MB porque el sistema debe alojarse completamente en una memoria de sólo lectura e instalarse en el coche”*
  - **Requisito de producto** que define una restricción en el tamaño del producto
- *“El sistema no debe desvelar ninguna información personal sobre los clientes excepto su nombre y su número de referencia”*
  - **Requisito externo** se deriva de la necesidad del sistema de cumplir la legislación vigente sobre protección de datos



# Requisitos no funcionales



[Sommerville, 2005]

# Requisitos no funcionales

Apariencia	Dependencia de terceros	Fiabilidad
Aspectos culturales y políticos	Disponibilidad	Gestión de fallos
Aspectos económicos	Documentación	Integridad
Aspectos legales y de licencias	Eficacia	Limitación de recursos
Auditoría y control	Eficiencia	Rendimiento
Calidad	Entorno de desarrollo	Resistencia
Certificación	Entorno físico	Robustez
Compatibilidad	Escalabilidad	Seguridad
Comunicaciones	Estabilidad	Soporte
Confidencialidad	Extensibilidad	Tiempo de respuesta
Custodia	Facilidad de mantenimiento	Usabilidad

# Métricas para medir requisitos no funcionales

<b>Rapidez</b>	Transacciones procesadas por segundo Tiempo de respuesta al usuario y a eventos Tiempo de actualización de la pantalla
<b>Tamaño</b>	K Bytes Número de chips de RAM
<b>Facilidad de uso</b>	Tiempo de formación Número de cuadros de ayuda
<b>Fiabilidad</b>	Tiempo medio entre fallos Probabilidad de no disponibilidad Tasa de ocurrencia de fallos Disponibilidad
<b>Robustez</b>	Tiempo de reinicio después de fallos Porcentaje de eventos que provocan fallos Probabilidad de corrupción de los datos después de fallos
<b>Portabilidad</b>	Porcentaje de declaraciones dependientes del objetivo Número de sistemas objetivo

# Requisitos funcionales

- Describen la funcionalidad o los servicios que se espera que el sistema proveerá, sus entradas y salidas, excepciones, etc
- Describen las interacciones entre el sistema y su entorno independientemente de la implementación
- Especifican acciones que el sistema debe realizar
- Especifican el comportamiento esperado del sistema

# Requisitos Funcionales

## Niveles de especificación

- **Requisitos de 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

- **Requisitos del sistema**

- Establecen con detalle las funciones, servicios y restricciones operativas del sistema
- Se debe definir exactamente qué se va a implementar

# Requisitos Funcionales

## Niveles de especificación

- Diferentes niveles de especificación de los requisitos son de utilidad debido a que comunican la información del sistema a diferentes tipos de lectores
- Los requisitos de usuario suelen ser más abstractos mientras que los requisitos del sistema añaden detalles y explican los servicios y funciones que el sistema debe proporcionar

# Requisitos Funcionales

- Ejemplos:
  - El usuario debe ser capaz de buscar entre todo el conjunto de bases de datos o de seleccionar un subconjunto de ellas.
  - Cada pedido tiene un identificador único (pedido\_id).

# Ejercicio

- Clasificar los siguientes requisitos en funcionales o no funcionales (de producto, de la organización o externo):
  1. La interfaz debe seguir la normativa de colores e imagen corporativa de la empresa
  2. La web debe seguir la normativa internacional en las áreas de protocolos, contenidos, herramientas, formatos y lenguajes relacionados con la accesibilidad a internet
  3. El área de trabajo debe poder ampliarse o reducirse mediante los correspondientes botones de acceso directo en la barra de herramientas



# Ejercicio

4. La aplicación permitirá consultar las facturas por diversos criterios, tales como facturas de pronto vencimiento, por cliente, por importe, etc
5. El acceso a la información personal sobre clientes no podrá ser consultada por los usuarios, excepto aquellos a los que les hayan sido concedidos privilegios especiales para ello
6. El programa cliente que realiza las búsquedas en la base de datos debe poder estar seguro de que la respuesta del sistema no ha sido falsificada o alterada

# Especificación de requisitos

- Objetivos diferentes según el punto de vista:
  - Para el **usuario o cliente**
    - Definir la **necesidad** (lo que se quiere y lo que no se quiere)
  - Para el **desarrollador**
    - **Comunicación** entre clientes, usuarios y desarrolladores
    - Permitir **iniciar** la actividad de **diseño**
    - Soportar las actividades de **prueba** del sistema
    - **Controlar** la evolución del sistema

# Especificación de requisitos

- Contenido de un documento de especificación:
  - Información acerca del problema
  - Interfaz externa del sistema con su entorno (SW, HW, usuarios, puertos de comunicación)
  - Propiedades y comportamiento del sistema
  - Restricciones de diseño y fabricación del producto
  - Descripciones acerca de cómo el sistema ayudará a los usuarios a realizar mejor sus tareas
  - Restricciones acerca de la tecnología que será utilizada en la construcción del sistema (protocolos, SO, etc)
  - Restricciones acerca de las propiedades emergentes del sistema

# Especificación de requisitos

- El documento de especificación no debe contener:
  - Requisitos del proyecto: planificación, costes, fases, hitos...
  - Diseño
  - Planes de garantía del producto

# Especificación de requisitos

- ¿Cómo escribir los requisitos?
  - Utilizar expresiones del tipo:
    - “El sistema hará X...”
    - “Se facilitará Y...”
    - Etc
  - Se pueden complementar con diagramas o notaciones formales
  - Deberán escribirse en un lenguaje que el usuario o cliente pueda entender

# Documento de requisitos

- Al menos utilizar dos documentos a distinto nivel de detalle:
  - Documento de Requisitos de **Usuario** (**DRU**, User Requirements Document)
  - Especificación de Requisitos **Software** (**ERS**, Software Requirements Specification)

# Documento de requisitos

- ¿En qué se diferencian los requisitos de usuario de los requisitos del software?
  - El DRU se escribe desde el punto de vista del usuario/cliente/interesado. Normalmente no poseen demasiado nivel de detalle
  - EL ERS desarrolla mucho más los contenidos del DRU
- La diferencia radica en el **nivel de detalle** de los requisitos

# Documento de requisitos

- Ejemplo: Diferencia entre DRU y ERS
  - **DRU**
    1. El software debe acceder a ficheros externos creados por otras herramientas
  - **ERS**
    - 1.1 El usuario debe poder elegir el tipo de fichero externo
    - 1.2 Cada tipo de fichero externo se puede asociar con la herramienta que produce ese tipo de fichero
    - 1.3 Cada tipo de fichero externo puede ser representado por un icono en la barra de herramientas
    - 1.4 Este icono puede ser definido por el usuario
    - 1.5 Cuando el usuario selecciona uno de estos iconos el efecto es aplicar la herramienta asociada a ese tipo de fichero



# Organización de requisitos

- Un conjunto de requisitos se puede agrupar si hacen referencia a:
  - **El mismo estímulo externo**
    - En un sistema automático de aterrizaje se pueden agrupar los requisitos relativos al sensor de altitud, al nivel de combustible, etc
  - **La misma característica del sistema**
    - En un sistema de telefonía se pueden agrupar los requisitos relativos a las llamadas locales, internacionales, etc
  - **La misma respuesta del sistema**
    - En un sistema de nóminas se pueden agrupar los requisitos relativos a generar pagos, generar listas de empleados actuales...
  - **El mismo objeto del mundo real**
    - En un sistema de biblioteca automatizada se pueden agrupar los requisitos relativos a libros, usuarios, etc
  - **La misma clase de usuarios**
    - En el sistema de la biblioteca se pueden separar los requisitos del bibliotecario, del usuario, etc
  - **La misma clase de funciones**
    - Datos de entrada,, interfaz de usuario, generación de informes, etc

# Estructura de una ERS

- Existen diferentes estándares para documentar requisitos
  - IEEE Std. 830
  - PSS-05 de la Agencia Espacial Europea (ESA)
- El estándar IEEE 830 propone diversas estructuras para el documento dependiendo de la forma de agrupar los requisitos

# Estructura de una ERS

- Estructura del documento de especificación de requisitos según el estándar IEEE 830 cuando los requisitos funcionales se agrupan por tipo de usuario

1. INTRODUCCIÓN
1.1. Propósito
1.2. Ámbito
1.3. Definiciones, acrónimos y abreviaturas
1.4. Referencias
1.5. Visión general del resto del documento
2. DESCRIPCIÓN GENERAL
2.1. Perspectiva del producto
2.2. Funciones del sistema
2.3. Características de los usuarios
2.4. Restricciones generales
2.5. Suposiciones y dependencias
3. REQUISITOS ESPECÍFICOS
3.1. Requisitos de interfaces externos
3.1.1. Interfaces de usuario
3.1.2. Interfaces hardware
3.1.3. Interfaces software
3.1.4. Interfaces de comunicaciones
3.2. Requisitos funcionales
3.2.1. Usuario 1
3.2.1.1. Requisito funcional 1.1.
...
3.2.1.n. Requisito funcional 1.n.
...
3.2.2. Usuario 2
3.2.2.1. Requisito funcional 2.1.
...
3.2.2.n. Requisito funcional 2.n.
3.3. Requisitos de rendimiento
3.4. Requisitos de diseño
3.5. Atributos del sistema
3.6. Otros requisitos
APÉNDICES
ÍNDICE

# Estructura de una ERS

- 1. Introducción:
  - 1.1 Propósito:
    - Propósito del documento y a quién va dirigido
  - 1.2 Ámbito
    - Se le da un nombre al futuro sistema. ¿Qué hace y qué no hace el producto SW? Beneficios, objetivos y metas
  - 1.3 Definiciones, siglas, y abreviaturas
    - En forma de apéndices o referencias a otros documentos
  - 1.4 Referencias
    - Lista completa de todas las referencias de los documentos en otra parte de la SRS
  - 1.5 Visión Global
    - Descripción de contenidos y cómo se organiza el resto de la SRS

# Estructura de una ERS

- 2. Descripción General:
  - 2.1 Perspectiva del Producto
    - Relación con otros Productos SW del Sistema
      - Interfaces Sistema; HW; SW; Comunicaciones ..
  - 2.2 Funciones del Producto
    - Resumen de las funciones principales (texto o gráficos)
  - 2.3 Características del Usuario
    - Tipo de usuarios al que está destinado el producto, experiencia técnica, nivel de conocimientos
  - 2.4 Restricciones Generales
    - Estándares, Limitaciones HW, Interfaces con otras aplicaciones
  - 2.5 Asunciones y Dependencias
    - Factores que pueden afectar a los requisitos especificados
  - 2.6 Requisitos Futuros

# Estructura de una ERS

- **3. Requisitos Específicos:**
- Contiene todos los requisitos software.
- Para cada requisito, se debe incluir:
  - Identificador único
  - Descripción de cada entrada (el estímulo) en el sistema,
  - Cada salida (la contestación) del sistema, y
  - Todas las funciones realizadas por el sistema en la salida a una entrada o en el apoyo de la salida.
- Los requisitos se pueden organizar y ordenar de varias maneras (ver anexo A del estándar).

# Documento de SRS

- Requisitos Específicos: según modo (anexo A)
  - 3.1 Interfaces Externas
    - Descripción detallada de las entradas y salidas del Sistema SW
    - Complementa las descripciones de Interfaz de los apartados anteriores
  - 3.2 Funciones (requisitos funcionales)
  - 3.3 Requisitos de Rendimiento/Ejecución
    - Estáticos y Dinámicos
  - 3.4 Restricciones de Diseño
    - Impuestas por otros estándares (formato informes; convenciones de nombrado elementos; etc..)
    - Limitaciones del HW
  - 3.5 Atributos de Calidad del Software
    - Fiabilidad, Disponibilidad, Mantenibilidad, Seguridad,
  - 3.6 Otros Requisitos

# Características de una ERS

- **No ambigua:**

- Un requisito ambiguo se presta a distintas interpretaciones
- Cada característica del producto final debe ser descrita utilizando un término único
- Si un término tiene distintos significados en distintos contextos se debe incluir un glosario

- **Completa:**

- Incluye todos los requisitos significativos del software
- Define la respuesta del software a todas las posibles clases de datos de entrada y en todas las posibles situaciones
- Está conforme con cualquier estándar de especificación que se deba cumplir.
- Están etiquetadas y referenciadas en el texto todas las figuras, tablas y diagramas.



# Características de una ERS

- **Fácil de Verificar**
  - Existe algún procedimiento finito y efectivo en coste para que una persona o máquina compruebe que el SW satisface cada requisito
- **Consistente**
  - Los Requisitos no entran en conflicto
- **Clasificada por orden de importancia o estabilidad**
  - Facilita la gestión a los desarrolladores

# Características de una ERS

- **Fácil de Modificar**

- Cualquier cambio se puede realizar fácil, de forma completa y consistente
- Implica una organización coherente y manejable (Tabla de Contenidos, Índice y Referencias Cruzadas)
- Es fundamental que la SRS sea No Redundante

- **Facilidad de Traza (Trazabilidad)**

- Debe facilitar las referencias con otros productos del ciclo de vida
- Referencias hacia atrás
- Referencias hacia delante

# Características de una ERS

- En la ERS el estándar IEEE 830 recomienda tratar los siguientes puntos:
  - **Capacidades funcionales:** ¿qué deberá hacer el software?
  - **Interfaces:** ¿cómo interacciona el software con los usuarios, con el hardware sobre el que funcionará y con otros elementos hardware y software externos?
  - **Niveles de rendimiento:** ¿qué se requiere en términos de velocidad, disponibilidad, tiempo de respuesta, etc?
  - **Atributos:** ¿qué consideraciones de seguridad, fiabilidad, protección de datos y privacidad o calidad se deben tener en cuenta?
  - **Restricciones y limitaciones de diseño:** ¿existen estándares que deban seguirse? ¿qué políticas sobre integridad de la BBDD, lenguajes de implementación, etc existen?

# Validación de requisitos

- El objetivo es determinar si los documentos de requisitos definen el software que los usuarios esperan
- Los métodos más comunes para la validación son:
  - Revisión de los requisitos
  - Prototipado
  - Validación del modelo
  - Pruebas de aceptación

# Validación de requisitos

- Revisión de los requisitos
  - Un grupo de personas (a menudo se incluye algún representante del cliente) revisa los documentos de requisitos en busca de inconsistencias, malentendidos, puntos poco claros, conflictos entre requisitos y otros problemas similares
  - La composición ideal del grupo de revisores debería contar con uno o varios usuarios, un responsable del cliente, desarrolladores, algún analista de requisitos y varios expertos funcionales en el problema
  - Como resultado de este proceso se elabora y publica una lista de problemas y posibles soluciones

# Validación de requisitos

- Prototipado
  - El prototipado permite mostrar el funcionamiento de los requisitos
  - Sirve para descubrir problemas y clarificar a los interesados algunas asunciones realizadas por los ingenieros del software
- Validación del modelo
  - Se verifica si el modelo es consistente y si refleja de forma adecuada los requisitos reales del sistema
  - Si se utiliza alguna notación de modelado es posible automatizar la validación del modelo

# Validación de requisitos

- Pruebas de aceptación
  - Los requisitos han de ser verificables
  - Las pruebas de aceptación consisten en la elaboración de un plan que establece cómo deben ser verificados los diferentes requisitos
  - Es útil para detectar problemas, ya que, para aquellos requisitos que estén descritos de forma ambigua será difícil elaborar un plan de verificación de los mismos

# Gestión de requisitos

- Los requisitos deben ser gestionados
- La Gestión de Requisitos implica la recolección, almacenamiento y mantenimiento de grandes cantidades de información
- Existen herramientas CASE que facilitan la gestión de requisitos:
  - BBDD para almacenar requisitos
  - Facilidades de análisis y generación de documentos
  - Facilidades de gestión de cambios
  - Facilidades de trazabilidad



# Gestión de requisitos

- La identificación y clasificación de requisitos es necesaria para poder gestionarlos de forma eficiente
- Los requisitos deben numerarse usando algún tipo de esquema
- Este esquema debe incluir una clasificación de los requisitos en grupos más manejables
- Existen diferentes técnicas para la identificación y clasificación de requisitos

# Gestión de requisitos

- Técnicas para clasificación e identificación:
  - Identificadores únicos. Numeración secuencial asignada manualmente o por una herramienta CASE de forma automática
  - Numeración según una jerarquía. Numeración asignada según la posición dentro del documento de requisitos. Por ejemplo, el séptimo requisito de la tercera sección del segundo capítulo: 2.3.7
  - Numeración secuencial dentro de cada categoría. Se asigna un identificador a la categoría del requisito al que se le añade la numeración correspondiente.

# Gestión de requisitos

- Los requisitos se pueden agrupar jerárquicamente
- Un requisito padre se compone de una serie de requisitos hijos
- Un requisito hijo es un sub-requisito de su padre
- Las relaciones jerárquicas introducen un nivel adicional a la clasificación de requisitos
- La jerarquía de requisitos permite definir requisitos en diferentes niveles de abstracción

# Gestión de requisitos

- Los requisitos cambian, pueden ser desestimados o pueden aparecer nuevos requisitos en cualquier etapa del ciclo de vida
- Cuanto más avanzado esté el desarrollo más costará introducir un cambio en los requisitos
- La gestión de cambios involucra un rastreo de todos los requisitos relacionados
- Las herramientas CASE que permiten manejar diferentes versiones y rastrear los cambios efectuados en los requisitos son muy útiles en estos casos

# Gestión de requisitos

- La trazabilidad de los requisitos es una parte importante de la gestión de los cambios
- La trazabilidad implica mantener una serie de relaciones entre requisitos para rastrear los efectos de un cambio

# Gestión de requisitos

- Algunos requisitos cambian más que otros
- **Requisitos Estables**
  - Esencia del Sistema y su Dominio de Aplicación
- **Requisitos Volátiles**
  - Específicos del Sistema en un entorno particular para un cliente particular
  - Las causas del cambio son variadas:
    - Corrección de errores y problemas en los requisitos
    - Conocimiento creciente del cliente/usuario
    - Problemas técnicos, de calendario o costes
    - Cambio en las prioridades del cliente

# Bibliografía

- Requirements Analysis and System Design. Leszek A. Maciaszek
- Ingeniería del Software. Ian Sommerville. Séptima edición
- Agile Software Requirements. Dean Leffingwell
- Ingeniería del Software. Un enfoque desde la guía SWEBOK. Salvador Sánchez et al.