

# **Análisis y Diseño de Sistemas**

---

**Tema: Ingeniería de Requerimientos**

**Estándar IEEE 830-1998 Documento ERS**

**- Breve descripción -**

**UNPSJB – 2014**

# Breve introducción....

---

**En sistemas con baja componente no software se generan dos tipos de documentos**

- **Documento de definición del sistema**  
documento de requisitos de usuario, define a alto nivel los requisitos desde la perspectiva del dominio o del negocio.
- **Especificación de los requisitos del software**  
Contrato, que debe hacer el software y que se espera que no haga. (IEEE std 830, std 1465, ISO/IEC 12119), puede complementarse con descripciones o modelos mas o menos formales

# Breve introducción....

---

- El estándar 830-1998 fue generado por un equipo de trabajo del IEEE, su finalidad es la integración de los requerimientos del sistema desde la perspectiva del usuario, cliente y desarrollador.
- Es esencialmente una guía para la redacción del ERS

# IEEE 830

---

- Quién la hizo: Software Engineering Standards Committee, del IEEE Computer Society
- IEEE (Institute of Electric and Electronic Engineers, en E.U.A.)
- Cuándo: 1998
- ¿Es de uso obligatorio? NO

# IEEE 830

---

- Quién la puede usar:
  - Un cliente/usuario que vaya a definir requerimientos (características) de un software que necesite
  - Un desarrollador (interno/externo) que haga software “a la medida” mediante proyecto
  - Un desarrollador que haga software “de paquete” que se venda masivamente

# IEEE 830 sirve para que...

---

- Un cliente describa claramente lo que quiere
- Un proveedor entienda claramente lo que el cliente quiere
- Se establezcan bases para un contrato de desarrollo (o de compra-venta)
- Se reduzca el esfuerzo de análisis, diseño, y programación (evitando re-trabajos)

# IEEE 830 sirve para que...

---

- Se tenga una base o referencia para validar o probar el software solicitado
- Se facilite el traspaso del software a otros clientes/usuarios
- Se le puedan hacer mejoras (o innovaciones) a ese software

# Ambiente del IEEE 830

---

- El ERS es la fuente principal para hacer el plan detallado de un proyecto de software
- Un SRS puede referirse a los requerimientos deseados de todos los componentes de un sistema grande, o a componentes (módulos) individuales del mismo
- Si se hacen ERS por separado para varios módulos, tiene que mantenerse la consistencia en los documentos
- Si un software necesita interactuar con otro, tienen que especificarse los requerimientos de esa interacción (interfaces), definiendo sus funcionalidades y el nivel de desempeño deseado



# Características de un buen ERS

---

- Correcto
- No ambiguo
- Completo
- Consistente
- Ordenado con base en importancia y/o estabilidad
- Verificable
- Modificable
- Rastreadable

# Correcto

---

- El ERS es correcto si los requerimientos escritos son aquellos que el software deberá cumplir
- No hay un método para saber si el ERS es correcto; lo importante es que se pida lo que realmente se necesita

# No ambiguo

---

- Un ERS es *no* ambiguo si cada requerimiento establecido en él tiene una sola interpretación posible, tanto por el cliente/usuario como por el desarrollador
- En casos donde alguna palabra pudiera tener múltiples significados, se debe incluir su definición precisa en un glosario que se adicione al ERS.
  - Ejemplos: planta, obra, maestro, carga, ...

# Completo

---

- El ERS es completo si incluye:
  - Todos los requerimientos significativos sobre funcionalidades, desempeño, restricciones de diseño, atributos, o interfaces externas
  - Las respuestas que debería dar el software a todas las posibles entradas de datos en todas las situaciones posibles (entradas aceptables o no aceptables: validación)
  - Especificación de unidades de medida (si son aplicables)
  - En caso de que el SRS tenga diagramas o tablas informativas, hay que darles número o identificación

# Consistente

---

- Los diversos requerimientos escritos tienen que ser compatibles entre sí; no debe haber contradicciones ni conflictos entre ellos.
- Para lograr la consistencia deben evitarse los siguientes conflictos:
  - En características especificadas de objetos del mundo real
  - De lógica o de tiempo entre dos actividades
  - Referencia a un mismo objeto del mundo real pero usando diferentes palabras para el mismo objeto

# Ordenado por grado de importancia o Estabilidad

---

- Cada requerimiento especificado debe tener alguna identificación (número, letra, secuencia alfanumérica) para indicar su grado de importancia o de estabilidad
- Algunos requerimientos son más importantes que otros
- Al “rankearlos” se puede dar a cada requerimiento la atención que se merece

# Ordenado por grado de importancia o Estabilidad

---

- Grado de estabilidad: número de cambios que se podrían esperar para un requerimiento, debido a eventos futuros que afecten a la organización, las responsabilidades, y las personas que usarán el software
- Grado de necesidad:
  - Esencial
  - Condicional
  - Opcional

# Verificable

---

- Un requerimiento es verificable si existe algún método cuantificable mediante el cual se pueda analizar si el software cumple ese requerimiento.
- Ejemplo:
  - “La respuesta del programa deberá darse dentro de los 20 seg posteriores a la apertura de la válvula VX389, y debe responder correctamente en el 60% de los casos”
- Contraejemplo (algo no verificable):
  - “El programa tiene que funcionar bien”



# Modificable

---

- Es modificable si la estructura y estilo de redacción permiten la realización de cambios en forma fácil, completa y consistente
- Para esto es recomendable:
  - Incluir índice, tabla de contenido y tabla de referencias cruzadas
  - Cada requerimiento debe aparecer sólo una vez (la redundancia propicia errores de inconsistencia)
  - Poner cada requerimiento separado de los demás

# Rastreable

---

- Cada requerimiento debe identificarse con algún número, letra o secuencia alfanumérica
- Un SRS es rastreable si el origen de cada requerimiento es claro, y si facilita el seguimiento de cada requerimiento a lo largo del proyecto de software
- Dos tipos de rastreabilidad:
  - **Hacia atrás:** en cada requerimiento se menciona explícitamente su fuente documental
  - **Hacia delante:** los documentos que se deriven del SRS deben usar las identificaciones de requerimientos como fueron dadas en el SRS

# Preparación conjunta de la ERS

---

- Es fundamental que la ERS se escriba de forma conjunta entre el cliente y el equipo de desarrollo de software
- Los clientes no saben lo suficiente de informática para escribirla ellos mismos
- Los desarrolladores no conocen suficientemente los problemas y el campo de trabajo del cliente

# Evolución de la ERS

---

- La ERS debe evolucionar conforme lo hace el proceso de desarrollo de software:
  - Puede ser imposible especificar ciertos detalles al comienzo del proyecto (y cuando esto es así, debe indicarse claramente)
  - Conforme el producto evoluciona pueden aparecer deficiencias, incorrecciones, etc., en los requerimientos no detectadas previamente

# Estructura del ERS.....

---

## **1. Introducción**

- 1.1. Propósito
- 1.2. Alcance
- 1.3. Definiciones, acrónimos y abreviaturas
- 1.4. Rereferencias
- 1.5. Revisiones

## **2. Descripción General**

- 2.1. Perspectiva del producto
- 2.2. Funciones del producto
- 2.3. Características de los usuarios
- 2.4. Restricciones
- 2.5. Dependencias

## **3. Requerimientos Específicos**

## **4. Apéndice**

## **5. Índice**

# Estructura del ERS.....

---

## **1. Introducción**

- 1.1. Propósito
- 1.2. Alcance
- 1.3. Definiciones, acrónimos y abreviaturas
- 1.4. Rereferencias
- 1.5. Revisiones

## **2. Descripción General**

- 2.1. Perspectiva del producto
- 2.2. Funciones del producto
- 2.3. Características de los usuarios
- 2.4. Restricciones
- 2.5. Dependencias

## **3. Requerimientos Específicos**

## **4. Apéndice**

## **5. Índice**

# Sección 3. Requerimientos Específicos

---

## **3. Requerimientos Específicos**

3.1 Interfaces Externas

3.2 Funciones

3.3 Requisitos de Rendimiento

3.4 Restricciones de Diseño

3.5 Atributos del Sistema

3.6 Otros Requisitos

# Contenido del ERS

---

- Sección 1: introducción
  - Debe incluir una descripción general del ERS, mostrando lo siguiente:
    - 1.1 Propósito del documento
    - 1.2 Alcance
    - 1.3 Definiciones, acrónimos, y abreviaturas
    - 1.4 Referencias
    - 1.5 Descripción general del documento
- En esta sección se proporcionará una introducción a todo el documento de Especificación de Requisitos Software



# Sección I: Introducción

---

## 1.1 Propósito del documento: aquí se tiene que:

- Explicar para qué se usará el documento
- Especificar a qué tipo de lectores está destinado (usuarios, clientes, analistas, programadores, etc.)

# Sección I: Introducción

---

## 1.2 Alcance

En esta subsección se pondrá nombre al futuro sistema, se explicará lo que el sistema hará y lo que no hará, se describirán los beneficios, objetivos y metas que se espera alcanzar con el futuro sistema y se mantendrán referencias a los documentos de nivel superior que puedan existir.

# Sección I: Introducción

---

## 1.2 Alcance – Aquí se tiene que:

- Identificar por su nombre genérico (y específico) el producto(s) de software que se estén requiriendo; por ejemplo: sistema de administración de inventario, generador de reportes, sistema manejador de bases de datos marca SúperX, etc.)
- Explicar lo que el software hará, y, si fuera necesario aclararlo, también lo que no se espera que haga

# Sección I: Introducción

---

## Algunas guías...

### 1.1 PROPÓSITO

- La razón de ser del documento
- Audiencia

Es una descripción breve, en dos o tres párrafos, a modo de prólogo.

Se habla del documento en sí, no del producto software

### 1.2 ALCANCE

- En la organización:  
mostrar organigrama sombreando el dpto. a atender.
- Respecto al producto:  
Módulos generales y submódulos, puede ser una tabla, un esquema, un diagrama de contexto.

# Sección I: Introducción

---

## 1.3 Definiciones, acrónimos, y abreviaturas

- Dar las definiciones de todos los términos, acrónimos (siglas) y abreviaturas que sean pertinentes para el adecuado entendimiento del ERS
- Esta información puede ofrecerse mediante referencias a uno o más apéndices del ERS o referencias hacia otros documentos (manuales de la organización, procedimientos de aseguramiento de calidad, hojas de registro, etc.)

# Sección I: Introducción

---

## Algunas guías...

### 1.3 Definiciones, acrónimos y abreviaturas

Generar una lista de términos que pudieran tener una interpretación distinta a la real.

Se sugiere describir por separado cada elemento y a modo de tabla.

- \* 1.3.1 Definiciones
- \* 1.3.2 Acrónimos
- \* 1.3.3 Abreviaturas

DEFINICIÓN ABREVIATURA ACRÓNIMO	DESCRIPCIÓN
DIF	Desarrollo integral de la familia

# Sección I: Introducción

---

## 1.4 Referencias

- Ofrecer una lista completa de todos los documentos a los que se haga referencia en el ERS
- Identificar cada documento mediante su título, número de reporte (si es aplicable), fecha, y organización que lo publicó
- Especificar las fuentes de las cuales pueden obtenerse los documentos referenciados
- Esta información puede ofrecerse haciendo alusión a un apéndice o hacia otro documento

# Sección I: Introducción

---

## Algunas guías...

### 1.4 Referencias

**Listado de las fuentes de información utilizadas para comprender en su totalidad el proceso a automatizar.**

- **Libros**
- **Páginas de internet**
- **Entrevistados**
- **Manuales**
- **Software**
- **Folletos**
- **Listados/reportes**



# Sección I: Introducción

---

## 1.5 Descripción General del Documento

- Describir lo que contienen las secciones restantes del documento ERS
- Explicar cómo está organizado

## Sección II. Descripción general

---

Aquí ***no*** se establecen los requerimientos específicos, sino que se describe el fondo general que da lugar a los requerimientos, los cuales se definen detalladamente hasta la sección 3 del SRS; esto los hace más fáciles de entender.

No se describen los requerimientos sino su contexto.

# Sección II. Descripción general

---

- Usualmente se incluyen estas 6 subsecciones:
  - 2.1 Perspectiva del producto
  - 2.2 Funciones del software
  - 2.3 Características del usuario
  - 2.4 Restricciones
  - 2.5 Suposiciones y dependencias
  - 2.6 Requerimientos futuros

# Sección II. Descripción general

---

## 2.1 Perspectiva del producto

- Punto específico para detallar los factores (requerimientos) técnicos que intervienen en el buen desarrollo del producto software
- Describir el software en perspectiva con otros software relacionados
- Si el software es independiente y totalmente auto-contenido, eso tiene que decirse aquí.
- Si se está requiriendo un software que formará parte de un sistema más grande, se tiene que describir la relación de los requerimientos del sistema grande con las funcionalidades del software requerido y debe identificarse cómo se comunicarán ambos

# Sección II. Descripción general

---

## 2.1 Perspectiva del producto

- Esta subsección debe relacionar el futuro sistema con otros productos. Se puede dividir en pequeñas subsecciones indicando cada uno de los puntos a tener en cuenta:
  - 2.1.1. Interfaz del sistema
  - 2.1.2. Interfaz de usuario
  - 2.1.3. Interfaz de hardware
  - 2.1.4. Interfaz de software
  - 2.1.5. Interfaz de comunicaciones
  - 2.1.6. Interfaz de memoria
  - 2.1.7. Operaciones
  - 2.1.8. Requerimientos de adaptación

# Sección II. Descripción general

---

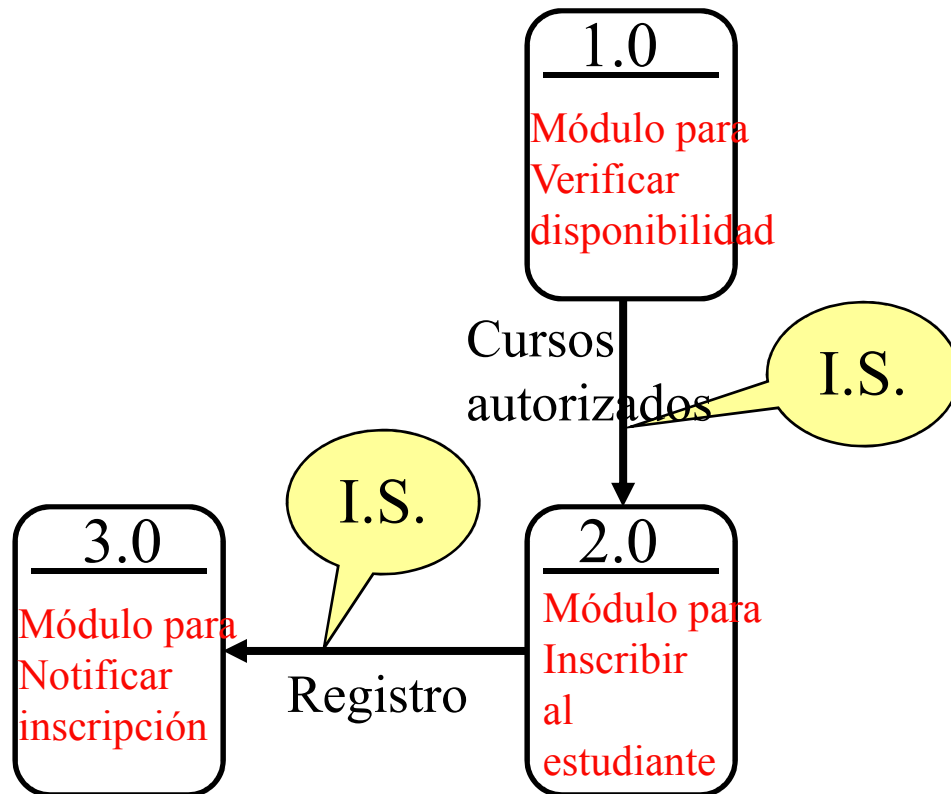
## 2.1 Perspectiva del Producto (cont.)

2.1.1 Interfaces de sistema: la comunicación entre los diversos módulos del sistema

- Enumerar cada interfaz de sistema
- Descripción de la interfaz del software para hacerlo compatible con el sistema

# Sección II. Descripción general

---



I.S. significa  
Interfaz de  
sistema

# Sección II. Descripción general

---

## 2.1 Perspectiva del software

### 2.1.2 Interfaces de usuario: la comunicación entre el usuario y el software

- Definir las características de :
  - Ventanas
  - Colores
  - Formatos y tamaños de letra
  - Menús
  - Íconos, botones
  - Contenido de los reportes impresos
  - Interfaz mediante A.S.R. y/o síntesis de voz...



# Sección II. Descripción general

---

## 2.1 Perspectiva del software

2.1.3 Interfaces de hardware: qué hardware usará el software para entrada/salida de información, incluyendo:

- Características de configuración (número de puertos de comunicación, etc.).
- Qué dispositivos de hardware serán soportados por el software y en qué forma serán soportados
- Protocolos de transferencia de datos entre dispositivos

Ejemplo: un sistema para administración de inventarios puede requerir el uso de scanners especiales para leer códigos de barras

# Sección II. Descripción general

---

## 2.1 Perspectiva del software

**2.1.4 Interfaces de software:** otros SWs necesarios para que el software requerido pueda funcionar, por ejemplo:

- Sistemas operativos: Windows, Linux, AIX, Solaris, etc.
- Sistemas manejadores de bases de datos: Oracle, Sybase, MySQL, DB2, etc.
- Intérpretes o compiladores de lenguajes
- Shells para sistemas expertos (como Sictus Prolog)
- Paquetes comerciales para ejecutar programas que usan redes neuronales (como MatLab)

También los programas para que el software pueda interactuar con los demás módulos

# Sección II. Descripción general

---

## 2.1 Perspectiva del software

### 2.1.5 Interfaces de comunicación:

Qué tecnología de redes se usará para comunicar a las computadoras en las cuales se instalará el software:

- Nombre genérico de la tecnología: LAN, WAN, MAN, VPN, WWW, WiFi, etc.
- Topología de la red: anillo, estrella, bus
- Protocolos de comunicación: Ethernet, TCP/IP, ATM, FrameRelay, PPP
- Tipo de cableado: fibra óptica, par trenzado, coaxial

# Sección II. Descripción general

---

## 2.1 Perspectiva del software

### 2.1.6 Restricciones de memoria

Se debe especificar si existe o no algún límite en cuanto a la memoria (tanto principal como secundaria) que podrá tener disponible el software para ser ejecutado

# Sección II. Descripción general

---

## 2.1 Perspectiva del software

### 2.1.7 Operaciones

Especificar los diferentes modos de operación del software por parte del usuario, por ejemplo:

2.1.7.1. Modos de operación de los distintos grupos de usuarios

2.1.7.2. Periodos de operaciones interactivas y automáticas

2.1.7.3. Funciones respaldo del procesamiento de datos

2.1.7.4. Operaciones de backup y recuperación

# Sección II. Descripción general

---

## 2.1 Perspectiva del software

### 2.1.8 Adaptación a un lugar específico

- Definir los requerimientos respecto a datos o secuencias de inicialización del software que sean específicas para un lugar, una misión o un modo operacional específico
- Especificar las características relacionadas con el lugar o la misión que deban ser modificadas para adaptar el software a una instalación específica

# Sección II. Descripción general

---

## 2.2 Funciones del producto

Sumario de las funciones principales; por ejemplo, para el caso de un software de contabilidad se mencionaría el mantenimiento de cuentas de cliente, el registro de sus transacciones, la preparación de facturas, pero sin mencionar los detalles requeridos de esas funciones.

# Sección II. Descripción general

---

## 2.2 Funciones del producto

- Se pueden utilizar métodos textuales o gráficos, siempre que dichos gráficos reflejen las relaciones entre funciones y no el diseño del sistema.
- Las funciones deben estar organizadas de manera que el cliente o cualquier otra persona lo entienda perfectamente.
- En AOO conviene utilizar Casos de Uso.



# Sección II. Descripción general

---

## 2.3 Características de los usuarios

Describir sus características respecto a:

- Nivel educativo
- Experiencia profesional
- Capacidades técnicas.

Esta descripción ayuda a comprender los motivos que dan origen a los requerimientos.

# Sección II. Descripción general

---

## 2.3 Características de los usuarios. Ejemplos

Tipo de usuario	Cajera
Formación	Conocimientos básicos de contabilidad
Habilidades	Manejo de computadores y sistemas de contabilidad
Actividades	Realizar operaciones de depósitos y retiros

Tipo de usuario	Asesor de crédito
Formación	Conocimientos sólidos de contabilidad (tercer nivel)
Habilidades	Manejo de computadores y sistemas de contabilidad
Actividades	Realizar los análisis de los clientes para los créditos, liquida créditos

Tipo de usuario	Gerente
Formación	Conocimientos sólidos de contabilidad (tercer nivel)
Habilidades	Manejo de computadores y sistemas de contabilidad
Actividades	Generar, analizar y tomar decisiones en función de los reportes que genere el sistema

# **Sección II. Descripción general**

---

## **2.4 Restricciones**

Información sobre posibles limitantes que deberán respetar los diseñadores, como:

- a) Políticas regulatorias de la Empresa
- b) Limitaciones en el hardware
- c) Interfaces hacia otras aplicaciones
- d) Operaciones paralelas
- e) Funciones de auditoría de software
- f) Consideraciones acerca de la seguridad...

Cualquier otra limitación que se imponga sobre los desarrolladores del producto.

# Sección II. Descripción general

---

## 2.5 Suposiciones y dependencias

- En este apartado aparecerá cualquier factor, que si cambia puede afectar a los requerimientos.
- No son restricciones de diseño, por ejemplo, asumir que un determinado sistema operativo estará disponible, presuponer una cierta organización de las unidades de la empresa.
- Si cambian ciertos detalles puede ser necesario revisar los requisitos.

# Sección II. Descripción general

---

## 2.6 Requerimientos futuros

- Se indican aquí posibles mejoras del sistema en el futuro. Estas mejoras deben estudiarse y analizarse una vez concluido y puesto en marcha el sistema.
- Son modificaciones a realizar en un futuro incierto.

## Sección III. Requerimientos Específicos

---

- Esta sección de la especificación de requisitos software contiene todos los requerimientos hasta un nivel de detalle suficiente para permitir a los diseñadores diseñar un sistema que satisfaga dichos requisitos, y que permita diseñar las pruebas que ratifiquen que el sistema cumple con las necesidades requeridas.
- Los requisitos que se aquí se indiquen deben describir comportamientos externos del sistema, observables por el usuario así como por parte de los operadores y otros sistemas.
- Puesto que deben indicarse todos los requisitos, esta sección es la más larga de la ERS y debe cumplir los principios descriptos.

# Sección III. Requerimientos específicos

---

- En esta sección se describe la funcionalidad del sistema desde dos perspectivas:

## REQ. NO FUNCIONALES



*Es la descripción de la forma cómo operaran las entradas y las salidas del sistema. Se complementa, mas no se repite, la información señalada en el apartado II (Descr. General).*

## REQ. FUNCIONALES



*Describen las acciones fundamentales sobre procesamiento para las entradas y la generación de salidas. Es una lista acerca de lo que el sistema “debe hacer”.*

# Sección III. Requerimientos Específicos

---

## 3.1. Interfaces externas

- En esta subsección se definirán los requisitos que afecten a:
  - la interfaz de usuario
  - la interfaz con otros sistemas (hardware y software),
  - así como a interfaces de comunicaciones.
- Ésta debe ser una descripción detallada de todas las entradas y salidas del sistema del software.
- Debe complementar las descripciones de la interfaz en 3.2 y no debe repetirse la información allí.



# Sección III. Requerimientos Específicos

---

## 3.1 Interfaces externas (requerimientos no funcionales)

- La redacción debe incluir la siguiente información:
  - Nombre del item
  - Descripción
  - Fuente de entradas y destino de salidas
  - Rangos, tolerancias, tiempos
  - Organización de ventanas
  - Formato de datos
- Se describe información de cómo operará cada interface en el sistema:
  - a) Interfaz de usuario
  - b) Interfaz de hardware
  - c) Interfaz de software
  - d) Interfaz de comunicaciones

# Sección III. Requerimientos Específicos

---

## 3.2. Funciones

- En esta subsección se deben especificar todas aquellas acciones o funciones que deberá llevar a cabo el sistema a desarrollar.
- El estándar IEEE 830, en sus últimas versiones, permite la organización de esta subsección de múltiples formas y simplemente sugiere alguna manera para hacerlo.
- Para organizar esta subsección de la ERS se elegirá alguna de las siguientes alternativas, o incluso alguna otra que se considere más conveniente.

# Sección III. Requerimientos Específicos

---

## 3.2. Funciones

Para lograr un mejor entendimiento de los requerimientos, conviene organizarlos con base en alguno de los siguientes criterios:

- Por modo de operación del sistema
- Por clase de usuario
- Por objetos
- Por características
- Por estímulos
- Por respuestas
- Por jerarquía funcional

# Sección III. Requerimientos Específicos

---

## 3.2. Funciones

- Se definen acciones fundamentales del sistema
  - Validación de entradas
  - Secuencia exacta de operaciones
  - Efectos de parámetros
  - Acciones a ejecutar
  - Recuperación de errores
  - Secuencia de entradas y salidas
  - Fórmulas y conversiones

# Sección III. Requerimientos Específicos

---

## 3.2. Funciones.

### Ejemplo Plantilla

Número de requisito	RF2
Nombre de requisito	Permitir la gestión (crear, modificar, eliminar) de usuarios, clientes y socios.
Tipo	<input type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	BD Tabla: Usuario, Clientes, Socios
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

- Se definen a continuación, entradas, procesos y salidas del requerimiento

# Sección III. Requerimientos Específicos

---

## 3.3. De rendimiento.

- Se detallan 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, se especifican los requisitos que afecten a la información que se guardará 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).

# Sección III. Requerimientos Específicos

---

## 3.4. Restricciones de diseño.

- Se incluyen aquí todas las restricciones que afecten al diseño de la aplicación, como pueden ser estándares internos de la organización, limitaciones hardware, etc.

# Sección III. Requerimientos Específicos

---

## 3.3. Requerimientos lógicos para base de datos

- Tipos de información para varias funciones
- Frecuencia de uso
- Capacidad de acceso
- Entidades y relaciones
- Limitaciones de integridad
- Requerimientos de persistencia

## 3.4 Restricciones de diseño

- Señalar si el hardware limita al diseño de interfaz y de datos
- Señalar qué otros estándares se van a aplicar que limiten funciones de diseño.



# Sección III. Requerimientos Específicos

---

## 3.5. Atributos del sistema

- Se detallan atributos como la fiabilidad, mantenibilidad, seguridad, mecanismos de acceso restringido (password), usuarios autorizados a realizar ciertas tareas críticas
- Narra la forma como se dará cumplimiento a los siguientes atributos:
  - a) Confiabilidad
  - b) Disponibilidad
  - c) Seguridad
  - d) Facilidad de mantenimiento
  - e) Portabilidad

# Sección III. Requerimientos Específicos

---

## 3.6. Otros requerimientos

- Podrían incluirse requerimientos adicionales a los especificados en el apartado 3.2.
- Se consideran requerimientos adicionales:
  - Diagramas de flujo de datos
  - Diccionario de datos

## Sección IV. Apéndices

---

- Los apéndices no siempre son considerados parte del SRS real y no siempre son necesarios. Ellos pueden incluir:
  - a) Ejemplos de formatos de las entradas/salidas, las descripciones del análisis del costo que se estudiaron o resultados de estudios del usuario;
  - b) Apoyando o dando información a fondo que puede ayudar a los lectores del SRS;
  - c) Una descripción de los problemas a ser resueltos por el software;
  - d) las instrucciones del empaquetamiento especiales para el código y los medios de comunicación para reunir la seguridad, exportar la carga inicial u otros requisitos.
- Cuando los apéndices son incluidos, el SRS debe declarar explícitamente si o no los apéndices serán considerados parte de los requisitos.

# Trabajo Práctico: ERS

---

## Sistema Servicio Postventas Concesionaria

- Redactar en forma completa la Definición de Requerimientos según esquema propuesto en clase teórica.
- Desarrollar la Especificación de Requerimientos de acuerdo a la estructura propuesta por IEEE 830 para el documento (Sección 1 y 2).
- Desarrollar la sección 3 del Documento ERS organizada por clase de usuario, utilizando una plantilla básica para especificar los requerimientos.

## ***La Plantilla de A.3 de ERS Sección 3 organizada por la clase del usuario***

### **3. Los requisitos específicos**

#### **3.1 Requisitos de la interface externa**

3.1.1 interfaz del usuario

3.1.2 interfaz del hardware

3.1.3 interfaz del software

3.1.4 interfaces de comunicaciones

#### **3.2 Requisitos funcionales**

##### **3.2.1 usuario clase 1**

3.2.1.1 requisito 1.1 funcional

.

.

.

3.2.1.n requisito 1.n Funcional

##### **3.2.2 usuario clase 2**

.

##### **3.2.m usuario clase m**

3.2.m.1 requisito Funcional m.1

.

3.2.m.n requisito Funcional m.n

## ***La Plantilla de A.3 de ERS Sección 3 organizada por la clase del usuario***

- 3.3 Requisitos de rendimiento
- 3.4 Restricciones de diseño
- 3.5 Atributos de sistema de software
- 3.6 Otros requisitos