

# **ERS (Especificaciones de Requerimientos de Software). Estándar 830-1993 IEEE (revisión del std 830-1984 IEEE)**

## **Descripción de los contenidos**

### **1. Introducción**

La introducción debe proveer un panorama global de toda la ERS.

#### **1.1. Objetivo**

- Determina el objetivo de la ERS.
- Determina los lectores de la ERS.

#### **1.2. Alcance**

- Determina el nombre del software a construir.
- Explica brevemente lo que el software va a hacer y, de ser necesario lo que no hará.
- Describe la aplicación del software especificado, incluyendo los beneficios y objetivos más relevantes.
- Consistente con las especificaciones similares de mayor nivel (por Ej.: especificaciones de requerimiento de equipamiento), si existen.

#### **1.3. Definiciones, acrónimos y abreviaturas**

Esta subsección explica todas las definiciones, abreviaturas, siglas y sinónimos utilizados en la especificación de requerimientos para su correcta interpretación. Esta información debe ser provista por referencia a uno o más apéndices en la ERS o por referencia a otros documentos.

#### **1.4. Referencias**

- Provee la lista de todos los documentos referenciados en la ERS.
- Identifica cada documento referenciado por título, fecha y editorial.
- Especifica las fuentes donde se pueden encontrar las referencias.

#### **1.5. Descripción global de la ERS**

- Describe los contenidos de la ERS.
- Explica la organización de la ERS.

### **2. Descripción General**

Esta sección de la ERS debe describir en general los factores que afectan el producto y sus requerimientos. Esta sección no expresa requerimientos específicos. En cambio, provee una base para esos requerimientos, los cuales se definen en detalle en la sección 3 haciéndolos más comprensibles.

#### **2.1. Perspectiva del Software**

Esta subsección debe poner al producto en perspectiva con otros productos relacionados. Establece si el software es totalmente independiente y autocontenido. En caso de formar parte de un sistema mayor, establece las interfaces de manera global. Puede usarse un diagrama en bloque mostrando los principales componentes del sistema mayor, sus interconexiones e interfaces externas.

#### 2.1.1. Interfaces de sistema.

Debe enumerarse cada interface del sistema e identificar la funcionalidad del software para cumplir con el requerimiento del sistema y la descripción de las interfaces que coincidan.

#### 2.1.2. Interfaces de usuario.

- *Características lógicas de cada interface entre el producto de software y sus usuarios.* Esto incluye las características de configuración (por ej.: formatos de pantalla requeridos, diseños de página o ventana, contenido de reportes o menús, o disponibilidad de programación de teclas de función) necesarias para cumplir los requerimientos del sistema.
- *Todos los aspectos para optimizar la interface con la persona que debe usar el sistema.* Puede contener una simple lista de cómo el sistema aparecerá al usuario. Un ejemplo puede ser el requerimiento para la opción de mensajes de error cortos o más largos.

#### 2.1.3. Interfaces de hardware.

Acá se deben especificar las características lógicas de cada interface entre el producto de software y los componentes del hardware del sistema. Esto incluye las características de configuración (cantidad de puertos, conjunto de instrucciones, etc.). También debe cubrir asuntos como qué dispositivos debe soportar, y los protocolos. Por ejemplo, el soporte de terminal puede especificar soporte de pantalla completa en vez de línea a línea.

#### 2.1.4. Interfaces de software.

Aquí se debe especificar si hay otros productos de software requeridos (por ejemplo, un sistema de administración de datos, un sistema operativo o un paquete matemático), e interfaces con otros sistemas de aplicación (por ejemplo, la conexión entre un sistema de cuentas a cobrar y un sistema mayor general). Para cada producto de software requerido, se debe proveer la siguiente información:

- Nombre
- Mnemónico
- Número de especificación
  
- Número de versión
- Fuente

Para cada interface, se debería proveer lo siguiente:

- Establecer el propósito de la conexión del software relacionado con este producto de software.
- Definición de la interface en cuanto a contenido y formato del mensaje. No es necesario detallar una interface bien documentada, pero se requiere una referencia al documento que define la interface.

- 2.1.5. Interfaces de comunicación.  
Aquí se deben especificar las diferentes interfaces de comunicación, tales como los protocolos de redes locales, etc.
- 2.1.6. Restricciones de memoria.  
Aquí se debe especificar cualquier característica y límite en las memorias principal y secundaria.
- 2.1.7. Operación.  
Se deberán especificar las operaciones normales y especiales requeridas por el usuario tales como:
  - Los distintos modos de operación en la organización usuaria; por ejemplo operaciones iniciadas por el usuario.
  - Períodos de operaciones interactivas y períodos de operaciones solas.
  - Funciones de soporte de procesamiento de datos.
  - Operaciones de resguardo y recuperación de datos.
- 2.1.8. Requerimientos de adaptación del lugar.
  - Define los requerimientos para cualquier dato o secuencia de inicialización que son específicos para un determinado lugar, misión o modo operacional, por ejemplo, tabla de valores, límites de seguridad, etc.
  - Especificar el lugar o las características relacionadas con la misión que se deberían modificar para adaptar el software a una instalación en particular.

## **2.2. Funciones del Software**

Esta subsección de la ERS debe proveer un resumen de las principales funciones que el software debe ejecutar. Por ejemplo, una ERS para un programa de contabilidad puede utilizar esta parte para asentar el mantenimiento de la contabilidad del cliente, declaración del cliente, y preparación de factura sin mencionar la gran cantidad de detalles que cada una de estas funciones necesita.

A veces el resumen de funciones que se requiere para esta parte, se puede tomar directamente de la sección de un nivel superior de especificación (si existe) que asigna funciones en particular al producto de software.

- Las funciones deben estar organizadas claramente de forma tal que resulte de fácil entendimiento para el que lo lee por primera vez.
- Textos o gráficos pueden usarse para mostrar las distintas funciones y sus relaciones.

## **2.3. Características del usuario**

Características generales sobre las personas que van a interactuar sobre el software: educación, experiencia, especialización técnica.

## **2.4. Restricciones generales**

Describe características generales que podrían limitar las opciones a los diseñadores del software. Puede incluir:

- Regulaciones

- Limitaciones de hardware
- Interfaces con otras aplicaciones
- Operación en paralelo
- Funciones de auditoria
- Funciones de control
- Requerimientos de lenguaje de alto nivel
- Protocolos (por ejemplo: XON-XOFF, ACK-NACK)
- Requerimientos de confiabilidad
- Gravedad de la aplicación
- Consideraciones de seguridad

## **2.5. Suposiciones y Dependencias**

Lista cada uno de los factores que afectan los requerimientos expresados en la ERS. Estos factores no son restricciones de diseño del software pero son más bien cambios que pueden afectar los requerimientos. Por ejemplo, la suposición de que un determinado sistema operativo va a estar disponible en el hardware designado para el producto. Si, de hecho, el sistema operativo no está disponible, la ERS deberá modificarse de acuerdo a esto.

## **2.6. Distribución de requerimientos**

Identifica requerimientos que pueden ser demorados hasta futuras versiones del sistema

### **3. Especificación de Requerimientos**

- Requerimientos específicos: correctos, completos, no ambiguos, consistentes, verificables, modificables, rastreables, ranqueados por importancia o estabilidad.
- Requerimientos específicos cruzados con documentos anteriores con los que están relacionados.
- Todos los requerimientos deben ser identificados unívocamente.
- Organizar los requerimientos para maximizar su lectura.

#### **3.1. Requerimientos de Interfaces Externas**

Debe contener una descripción detallada de todas las entradas y las salidas que posee el software. Debe complementar la descripción de interface en la sección 2 sin repetir la información. Debe incluir la siguiente información:

- Nombre del ítem
- Objetivo
- Fuentes de entrada o destino de la salida
- Rangos válidos, exactos y/o tolerancia
- Unidades de medida
- Oportunidad
- Relaciones con otras entradas/salidas
- Formatos de ventanas/organización
- Formatos datos
- Comandos de formatos
- Mensajes de finalización

##### **3.1.1. Interfaces de usuario**

- Describe las características de la faz hombre-máquina.
- Incluye: formato de la pantalla, *lay-out* y contenido de las salidas y los menús, secuencia de uso.

##### **3.1.2. Interfaces de hardware**

- Describe las fases entre el software y el hardware, incluye los dispositivos y periféricos a utilizar.

##### **3.1.3. Interfaces de software**

- Describe el uso de otros productos de software y/o las fases con otras aplicaciones existentes

##### **3.1.4. Interfaces de comunicaciones**

- Describe las interfaces de comunicaciones tales como protocolos de redes, etc.

### 3.2. Requerimientos Funcionales

Los requerimientos funcionales definen las acciones fundamentales que se deben realizar en el software para aceptar y procesar las entradas y procesando y generando las salidas.

#### 3.2.1. Requerimiento Funcional #1

##### 3.2.1.1. Especificación

###### 3.2.1.1.1. Introducción

- Describe el objetivo de la función

###### 3.2.1.1.2. Entradas

- Describe en forma detallada todos los datos de entrada.
- Incluye: fuente de los datos, cantidad, secuencia, rangos.

###### 3.2.1.1.3. Procesos

- Describe todas las acciones a realizarse sobre los datos de entrada para obtener la salida.
- Incluye: validación de datos de entrada, secuencia, respuestas a situaciones anormales, etc.

###### 3.2.1.1.4. Salidas

- Describe en forma detallada todos los datos de salida.
- Incluye: destino de los datos, cantidad, secuencia, rangos válidos, relaciones entre E/S, fórmulas de conversión, etc.

#### 3.2.2. Requerimiento Funcional #2 ...

### 3.3. Requerimientos de Performance

Describe los requerimientos numéricos estáticos y dinámicos establecidos para el software o la interacción de las personas con el mismo:

Estáticos:

- número de terminales
- cantidad de usuarios concurrentes
- cantidad y tipo de información a manejar

Dinámicos:

- tareas y cantidad de transacciones a procesar durante las horas pico y normales de uso de la aplicación.

### **3.4. Requerimientos lógicos de la base de datos**

Aquí se deben especificar los requerimientos lógicos para cualquier información que se ingrese en la base de datos.

- Tipos de información usados por varias funciones
- Frecuencia de uso
- Capacidades de acceso
- Entidades de datos y sus relaciones
- Restricciones de integridad
- Requerimientos de retención de datos

### **3.5. Restricciones de Diseño**

#### **3.5.1. Cumplimiento de Estándares**

- Requerimientos de Estándares tales como: formato de informes, convenciones de nombres, procedimientos contables, pistas de auditoría.

#### **3.5.2. Limitaciones de Hardware**

- Describe requerimientos para que el software pueda operar en determinados tipos de hardware.

### **3.6. Atributos del Software**

#### **3.6.1. Confiabilidad**

Especifica los factores necesarios para determinar el nivel de confiabilidad del software en el momento de entrega

#### **3.6.2. Disponibilidad**

Especifica los factores necesarios para garantizar el nivel de disponibilidad del software tales como puntos de control, recuperación ante caídas, arranque, etc.

#### **3.6.3. Seguridad**

Especifica los elementos de protección contra accesos accidentales o maliciosos para usar, modificar, destruir o revelar. Requerimientos específicos en esta área pueden incluir:

- Uso de técnicas criptográficas
- Mantenimiento de un historial de transacciones
- Asignación de funciones a diferentes módulos
- Restricción de comunicaciones entre algunas áreas del programa
- Verificación de la integridad de los datos para variables críticas

#### 3.6.4. Facilidad de Mantenimiento

Especifica qué tipos de métricas y qué valores de los mismos aseguran la facilidad de mantenimiento. Puede ser cierta modularidad, interfaces, complejidad, etc.

#### 3.6.5. Portabilidad

Especifica los atributos del software relacionados con la facilidad para cambiarlo a otro servidor o sistema operativo. Puede incluir:

- Porcentaje de componentes que dependen del servidor
- Porcentaje de código que depende del servidor
- Uso de lenguaje portable probado
- Uso de un compilador particular
- Uso de un sistema operativo determinado

### 4. ***Información de apoyo***

Esta información hace a la ERS más fácil de usar. Incluye:

- Tabla de contenidos
- Índice
- Apéndices

#### 4.1. **Tabla de contenidos e Índice**

#### 4.2. **Apéndices**

Los Apéndices no siempre son considerados parte de la especificación de requerimientos actuales y no siempre son necesarios. Pueden incluir:

- Ejemplos de formatos de E/S, estudios de análisis de costos, o resultados de encuestas a usuarios.
- Información de referencia o apoyo que puede ayudar a los lectores de la ERS.
- Una descripción de los a ser resueltos por el software.
- Instrucciones especiales de empaquetamiento para el código y el medio en cuanto a seguridad, carga inicial, exportación u otro requerimiento.

Cuando se incluyen apéndices, la ERS debe explicitar si deben ser considerados como parte de los requerimientos.