

# IEEE-830

## Contents

<b>1</b>	<b>Introducción</b>	<b>2</b>
1.1	Propósito . . . . .	2
1.2	Ámbito del sistema . . . . .	2
1.3	Definiciones, Acrónimos y Abreviaturas . . . . .	2
1.4	Referencias . . . . .	2
1.5	Visión general del documento . . . . .	2
<b>2</b>	<b>Descripción general del documento</b>	<b>3</b>
2.1	Perspectiva del producto . . . . .	3
2.2	Funciones del producto . . . . .	3
2.3	Características de los usuarios . . . . .	3
2.4	Restricciones . . . . .	3
2.5	Suposiciones y dependencias . . . . .	4
2.6	Requisitos futuros . . . . .	4
<b>3</b>	<b>Requisitos específicos</b>	<b>4</b>
3.1	Interfaces externas . . . . .	5
3.2	Funciones . . . . .	5
3.3	Requisitos de rendimiento . . . . .	6
3.4	Restricciones de diseño . . . . .	6
3.5	Atributos del sistema . . . . .	6
3.6	Otros requisitos . . . . .	6
<b>4</b>	<b>Apéndices</b>	<b>7</b>

# 1 Introducción

En esta sección se proporcionará una introducción a todo el documento de Especificación de Requisitos Software(ERS). Consta de varias subsecciones: propósito, ámbito del sistema, definiciones, referencias y visión general del documento.

## 1.1 Propósito

El propósito del presente documento de ERS es contar con los requisitos necesarios para desarrollar el software del equipo de medición de material particulado fino (MP2,5) presente en atmósferas contaminadas. Dicha solución debe permitir una mayor precisión y exactitud que los sensores ópticos de bajo costo, mediante técnicas estadísticas de muestreo y manejo de algoritmos numéricos. Se pretende elaborar una solución económica y fiable que pueda integrarse en las redes de control de calidad del aire, comúnmente gestionadas por autoridades ambientales o gobiernos locales. Con esto se espera contribuir a la mejora de la salud pública en entornos urbanos.

## 1.2 Ámbito del sistema

- Este software se denominara comercialmente como **SECA** (Sistema para Equipos de Calidad de Aire)
- El software del proyecto facilitará cálculos en tiempo real y monitoreo del estado de sensores, emitiendo alertas ante fallos. No obstante, post-pruebas no habrá mantenimiento, actualizaciones, ni formación para usuarios. Tampoco se planifican desarrollos futuros del software. En conclusión, el software está diseñado para funcionalidad inmediata y fiabilidad, pero sin soporte continuo o expansión post-proyecto.
- 
- Se referenciarán todos aquellos documentos de nivel superior (p.e en Industria de sistemas, que incluyen hardware y software, deberían mantenerse la consistencia con el documento de especificaciones de requisitos globales del sistema, si existe)

## 1.3 Definiciones, Acrónimos y Abreviaturas

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

## 1.4 Referencias

En esta subsección se mostrará una lista completa de todos los documentos referenciados en la ERS

## 1.5 Visión general del documento

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

## 2 Descripción general del documento

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

Normalmente, esta sección consta de las siguientes subsecciones: Perspectiva del producto, funciones del producto, características de los usuarios, restricciones, factores que se asumen y futuros requisitos.

### 2.1 Perspectiva del producto

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

### 2.2 Funciones del producto

En esta subsección de la ERS se mostrará un resumen, a grandes rasgos, de las funciones del futuro sistema, por ejemplo, en una ERS para un programa de contabilidad, esta subsección mostrará que el sistema soportará el mantenimiento de cuentas, mostrará el estado de las cuentas y facilitará la facturación, sin mencionar el enorme detalle que cada una de estas funciones requiere.

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

### 2.3 Características de los usuarios

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

### 2.4 Restricciones

Esta subsección describirá aquellas limitaciones que se imponen sobre los desarrolladores del producto.

- Políticas de la empresa.
- Limitaciones del hardware.
- Interfaces con otras aplicaciones.
- Operaciones paralelas.
- Funciones de auditoría
- Funciones de control.
- Lenguaje(s) de programación
- Protocolos de comunicación.

- Requisitos de habilidad.
- Criticalidad de la aplicación.
- Consideraciones acerca de la seguridad.

## 2.5 Suposiciones y dependencias

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

## 2.6 Requisitos futuros

Esta subsección esbozará futuras mejoras al sistema, que podrán analizarse e implementarse en el futuro.

# 3 Requisitos específicos

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que satisfaga estos requisitos, y demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas. Esta es la sección más larga e importante de la ERS. Deberán aplicarse los siguientes principios:

- El documento debería ser perfectamente legible por personas de muy distintas formaciones e intereses.
- Deberán referenciarse aquellos documentos relevantes que poseen alguna influencia sobre los requisitos.
- Todo requisito deberá ser unívocamente identificable mediante algún código o sistema de numeración adecuado.
- Lo ideal, aunque en la práctica no siempre realizable, es que los requisitos posean las siguientes características:
  - **Corrección:** La ERS es correcta si y sólo si todo requisito que figura aquí(y que será implementado en el sistema) refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el deseado.
  - **No ambiguos:** Cada requisito tiene una sola interpretación. Para eliminar la ambigüedad inherente a los requisitos expresados en lenguaje natural, se deberán utilizar gráficos o notaciones formales. En el caso de utilizar términos que, habitualmente, poseen más de una interpretación, se definirán con precisión en glosario.
  - **Completo:** Todos los requisitos relevantes han sido incluidos en la ERS. Conviene incluir todas las posibles respuestas del sistema a los datos de entrada, tanto válidos como no válidos.

- **Consistentes:** Los requisitos no pueden ser contradictorios. Un conjunto de requisitos contradictorios no es implementable.
- **Clasificados:** Normalmente, no todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por importancia (esenciales, condicionales u opcionales) o por estabilidad (cambios que se espera que afecten al requisito). Esto sirve, ante todo, para no emplear excesivos recursos en implementar requisitos no esenciales.
- **Verificables:** La ERS es verificable si y sólo si todos sus requisitos son verificables. Un requisito es verificable (testable) si existe un proceso finito y no costoso para demostrar que el sistema cumple con el requisito. Un requisito ambiguo no es, en general, verificable. Expresiones como a veces, bien, adecuado, etc introducen ambigüedad en los requisitos. Requisitos como "en caso de accidente la nube tóxica no se extenderá más allá de 25km" no es verificable por el alto costo que conlleva.
- **Modificables:** La ERS es modificable si y sólo si se encuentra estructurada de forma que los cambios a los requisitos puedan realizarse de forma fácil, completa y consistente. La utilización de herramientas automáticas de gestión de requisito (por ejemplo RequisitePro o Doors) facilitan enormemente esta tarea.
- **Trazables:** La ERS es trazable si se conoce el origen de cada requisito y facilita la referencia de cada requisito a los componentes y de la implementación. La trazabilidad hacia atrás indica el origen (documento, persona, etc) de cada requisito. La trazabilidad hacia delante de un requisito R indica qué componentes del sistema son los que realizan el registro R.

### 3.1 Interfaces externas

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

### 3.2 Funciones

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

Es importante tener en cuenta que, en 1983, el estándar de IEEE 830 establecía que las funciones deberían expresarse como una jerarquía funcional (en paralelo con los DFDs propuestas por el análisis estructurado). Pero el estándar de IEEE 830, en sus últimas versiones, ya permite organizar esta subsección de múltiples formas, y sugiere, entre otras, las siguientes:

- Por tipos de usuarios: Distintos usuarios poseen distintos requisitos. Para cada clase de usuario que exista en la organización, se especificarán los requisitos funcionales que le afecten o tengan mayor relación con sus tareas.
- Por objetos: Los objetos son identidades del mundo real que serán reflejadas en el sistema. Para cada objeto, se detallarán sus atributos y sus funciones. Los objetos pueden agruparse en clases. Esta organización de la ERS no quiere decir que el diseño del sistema siga el paradigma de Orientación a Objetos.

- Por estímulos: Se especificarán los posibles estímulos que recibe el sistema y las funciones relacionadas con dicho estímulo.
- Por jerarquía funcional: Si ninguna de las anteriores alternativas resulta de ayuda, la funcionalidad del sistema se especificará como una jerarquía de funciones que comparten entradas, salidas o datos internos. Se detallarán las funciones (entrada, proceso, salida) y las subfunciones del sistema. Esto no implica que el diseño del sistema deba realizarse según el paradigma de diseño estructurado.

Para organizar esta subsección de la ERS se elegirá alguna de las anteriores alternativas, o incluso alguna otra que se considere más conveniente. Deberá, eso sí, justificarse el porqué de tal elección.

### 3.3 Requisitos de rendimiento

Se detallarán los requisitos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema, etc. También, si es necesario, se especificarán los requisitos de datos, es decir, aquellos 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).

### 3.4 Restricciones de diseño

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

### 3.5 Atributos del sistema

Se detallarán los atributos de calidad (las "ilities") del sistema. Fiabilidad, mantenibilidad, portabilidad, y muy importante, la seguridad. Deberá especificarse qué tipos de usuarios están autorizados, o no, a realizar ciertas tareas, y cómo se implementarán los mecanismos de seguridad (por ejemplo, por medio de un *login* y una *password*).

### 3.6 Otros requisitos

Cualquier otro requisito que no encaje en otra sección.

## 4 Apéndices

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

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