PL1 Ingeniería del Software

Álvaro Camacho Martín : 48083824R

**´Indice**

1. [Introducci´on 3](#_TOC_250020)
   1. [Prop´osito 3](#_TOC_250019)
   2. [A´mbito del Sistema 3](#_TOC_250018)
   3. [Definiciones, Acr´onimos y Abreviaturas 3](#_TOC_250017)
   4. [Referencias 3](#_TOC_250016)
   5. [Visi´on General del Documento 4](#_TOC_250015)
2. [Descripci´on General 4](#_TOC_250014)
   1. [Perspectiva del Producto 4](#_TOC_250013)
   2. [Funciones del Producto 4](#_TOC_250012)
   3. [Caracter´ısticas de los Usuarios 5](#_TOC_250011)
   4. [Restricciones 5](#_TOC_250010)
   5. [Suposiciones y Dependencias 5](#_TOC_250009)
   6. [Requisitos Futuros 6](#_TOC_250008)
3. [Requisitos Espec´ıficos 6](#_TOC_250007)
   1. [Interfaces Externas 7](#_TOC_250006)
   2. [Funciones 7](#_TOC_250005)
   3. [Requisitos de Rendimiento 9](#_TOC_250004)
   4. [Restricciones de Disen˜o 9](#_TOC_250003)
   5. [Atributos del Sistema 9](#_TOC_250002)
   6. [Otros Requisitos 9](#_TOC_250001)
4. [Ap´endices 9](#_TOC_250000)

# Introducci´on

En este documento se redactará el funcionamiento y requisitos del sistema de la empresa ‘DIVIERTE’.

## Prop´osito

El objetivo de este documento es explicar el funcionamiento del sistema, especificar los requisitos, y servir de guía a cualquier usuario de la aplicación, tanto a clientes como administradores o gestores. Además ofrece instrucciones para creadores y programadores en el desarrollo del sistema.

## A´mbito del Sistema

En esta subsecci´on:

Se podr´a dar un nombre al futuro sistema (p.ej. MiSistema)

Se explicar´a lo que el sistema har´a y lo que no har´a.

Se describir´an los beneficios, objetivos y metas que se espera alcanzar con el futuro sistema.

Se referenciar´an todos aquellos documentos de nivel superior (p.e. en In- genier´ıa de Sistemas, que incluyen Hardware y Software, deber´ıa man- tenerse la consistencia con el documento de especificaci´on de requisitos globales del sistema, si existe).

## Definiciones, Acr´onimos y Abreviaturas

En esta subsecci´on se definir´an todos los t´erminos, acronimos y abrevia- turas utilizadas en la ERS.

## Referencias

En esta subsecci´on se mostrar´a una lista completa de todos los documen- tos referenciados en la ERS.

## Visi´on General del Documento

Esta subsecci´on describe brevemente los contenidos y la organizaci´on del resto de la ERS.

# Descripci´on General

En esta secci´on se describen todos aquellos factores que afectan al pro- ducto y a sus requisitos. No se describen los requisitos, sino su contexto. Esto permitir´a definir con detalle los requisitos en la secci´on 3, haciendo que sean m´as f´aciles de entender.

Normalmente, esta secci´on consta de las siguientes subsecciones: Pers- pectiva del producto, funciones del producto, caracter´ısticas de los usuarios, restricciones, factores que se asumen y futuros requisitos.

## Perspectiva del Producto

Esta subsecci´on debe relacionar el futuro sistema (producto software) con otros productos. Si el producto es totalemente independiente de otros pro- ductos, tambi´en debe especificarse aqu´ı. Si la ERS define un producto que es parte de un sistema mayor, esta subsecci´on relacionar´a los requisitos del sistema mayor con la funcionalidad del producto descrito en la ERS, y se identificar´an las interfaces entre el producto mayor y el producto aqu´ı des- crito. Se recomienda utilizar diagramas de bloques.

## Funciones del Producto

En esta subsecci´on de la ERS se mostrar´a un resumen, a grandes rasgos, de las funciones del futuro sistema. Por ejemplo, en una ERS para un pro- grama de contabilidad, esta subsecci´on mostrar´a que el sistema soportar´a el mantenimiento de cuentas, mostrar´a el estado de las cuentas y facilitar´a la facturaci´on, sin mencionar el enorme detalle que cada una de estas funciones requiere.

Las funciones deber´an mostrarse de forma organizada, y pueden utili- zarse gr´aficos, siempre y cuando dichos gr´aficos reflejen las relaciones entre funciones y no el disen˜o del sistema.

## Caracter´ısticas de los Usuarios

Esta subsecci´on describir´a las caracter´ısticas generales de los usuarios del producto, incluyendo nivel educacional, experiencia y experiencia t´ecnica.

## Restricciones

Esta subsecci´on describir´a aquellas limitaciones que se imponen sobre los desarrolladores del producto

Pol´ıticas de la empresa

Limitaciones del hardware

Interfaces con otras aplicaciones

Operaciones paralelas

Funciones de auditor´ıa

Funciones de control

Lenguaje(s) de programacion

Protocolos de comunicaci´on

Requisitos de habilidad

Criticalidad de la aplicaci´on

Consideraciones acerca de la seguridad

## Suposiciones y Dependencias

Esta subsecci´on de la ERS describir´a aquellos factores que, si cambian, pueden afectar a los requisitos. Por ejemplo, los requisitos pueden presu- poner una cierta organizaci´on de ciertas unidades de la empresa, o pueden presuponer que el sistema correr´a sobre cierto sistema operativo. Si cambian dichos detalles en la organizaci´on de la empresa, o si cambian ciertos detalles t´ecnicos, como el sistema operativo, puede ser necesario revisar y cambiar los requisitos.

## Requisitos Futuros

Esta subsecci´on esbozar´a futuras mejoras al sistema, que podr´an anali- zarse e implementarse en un futuro.

# Requisitos Espec´ıficos

Esta secci´on contiene los requisitos a un nivel de detalle suficiente como para permitir a los disen˜adores disen˜ar un sistema que satisfaga estos requi- sitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aqu´ı es- pecificado describir´a comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas. Esta es la secci´on m´as larga e importante de la ERS. Deber´an aplicarse los siguientes principios:

El documento deber´ıa ser perfectamente legible por personas de muy distintas formaciones e intereses.

Deber´an referenciarse aquellos documentos relevantes que poseen algu- na influencia sobre los requisitos.

Todo requisito deber´a ser un´ıvocamente identificable mediante algu´n c´odigo o sistema de numeraci´on adecuado.

Lo ideal, aunque en la pr´actica no siempre realizable, es que los requi- sitos posean las siguientes caracter´ısticas:

Correccion: La ERS es correcta si y s´olo si todo requisito que figura aqu´ı (y que ser´a implementado en el sistema) refleja alguna necesidad real. La correcci´on de la ERS implica que el sistema implementado ser´a el sistema deseado.

*•*

No ambiguos: Cada requisito tiene una sola interpretaci´on. Para eliminar la ambigu¨edad inherente a los requisitos expresados en lenguaje natural, se deber´an utilizar gr´aficos o notaciones forma- les. En el caso de utilizar t´erminos que, habitualmente, poseen m´as de una interpretaci´on, se definir´an con precisi´on en el glosario.

*•*

Completos: Todos los requisitos relevantes han sido incluidos en la ERS. Conviene incluir todas las posibles respuestas del sistema a los datos de entrada, tanto v´alidos como no v´alidos.

*•*

Consistentes: Los requisitos no pueden ser contradictorios. Un con- junto de requisitos contradictorio 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 (cam- bios 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 s´olo si todos sus requisitos son verificables. Un requisito es verificable (testeable) si existe un proceso finito y no costoso para demostrar que el sistema cumple con el requisito. Un requisito ambiguo no es, en general, verifi- cable. Expresiones como a veces, bien, adecuado, etc. introducen ambigu¨edad en los requisitos. Requisitos como “en caso de acci- dente la nube t´oxica no se extender´a m´as all´a de 25Km” no es verificable por el alto costo que conlleva.

*•*

Modificables: La ERS es modificable si y s´olo si se encuentra es- tructurada de forma que los cambios a los requisitos pueden rea- lizarse de forma f´acil, completa y consistente. La utilizaci´on de herramientas autom´aticas de gesti´on de requisitos (por ejemplo RequisitePro o Doors) facilitan enormemente esta tarea.

*•*

Trazables: La ERS es trazable si se conoce el origen de cada requi- sito y se facilita la referencia de cada requisito a los componentes del disen˜o y de la implementaci´on. La trazabilidad hacia atr´as indica el origen (documento, persona, etc.) de cada requisito. La trazabilidad hacia delante de un requisito R indica qu´e compo- nentes del sistema son los que realizan el requisito R.

*•*

## Interfaces Externas

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

## Funciones

Esta subsecci´on (quiz´a la m´as larga del documento) deber´a especificar todas aquellas acciones (funciones) que deber´a llevar a cabo el software. Nor-

malmente (aunque no siempre), son aquellas acciones expresables como “el sistema deber´a . . . ”. Si se considera necesario, podr´an utilizarse notaciones gr´aficas y tablas, pero siempre supeditadas al lenguaje natural, y no al rev´es. Es importante tener en cuenta que, en 1983, el Est´andar de IEEE 830 establec´ıa que las funciones deber´ıan expresarse como una jerarqu´ıa funcional (en paralelo con los DFDs propuestos por el an´alisis estructurado). Pero el Est´andar de IEEE 830, en sus u´ltimas versiones, ya permite organizar esta

subsecci´on de mu´ltiples formas, y sugiere, entre otras, las siguientes:

Por tipos de usuario: Distintos usuarios poseen distintos requisitos. Pa- ra cada clase de usuario que exista en la organizaci´on, se especificar´an los requisitos funcionales que le afecten o tengan mayor relaci´on con sus tareas.

Por objetos: Los objetos son entidades del mundo real que ser´an refle- jadas en el sistema. Para cada objeto, se detallar´an sus atributos y sus funciones. Los objetos pueden agruparse en clases. Esta organizacion de la ERS no quiere decir que el disen˜o del sistema siga el paradigma de Orientaci´on a Objetos.

Por objetivos: Un objetivo es un servicio que se desea que ofrezca el sistema y que requiere una determinada entrada para obtener su resul- tado. Para cada objetivo o subobjetivo que se persiga con el sistema, se detallar´an las funciones que permitan llevarlo a cabo.

Por est´ımulos: Se especificar´an los posibles est´ımulos que recibe el sis- tema y las funciones relacionadas con dicho est´ımulo.

Por jerarqu´ıa funcional: Si ninguna de las anteriores alternativas resulta de ayuda, la funcionalidad del sistema se especificar´a como una jerar- qu´ıa de funciones que comparten entradas, salidas o datos internos. Se detallar´an las funciones (entrada, proceso, salida) y las subfunciones del sistema. Esto no implica que el disen˜o del sistema deba realizarse segu´n el paradigma de Disen˜o Estructurado.

Para organizar esta subsecci´on de la ERS se elegir´a alguna de las ante- riores alternativas, o incluso alguna otra que se considere m´as conveniente. Deber´a, eso s´ı, justificarse el porqu´e de tal elecci´on.

## Requisitos de Rendimiento

Se detallar´an los requisitos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el nu´mero de terminales, el nu´mero esperado de usuarios simultaneamente conectados, nu´mero de transacciones por segundo que deber´a soportar el sistema, etc.

Tambi´en, si es necesario, se especificar´an lo requisitos de datos, es decir, aquellos requisitos que afecten a la informaci´on que se guardar´a 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).

## Restricciones de Disen˜o

Todo aquello que restrinja las decisiones relativas al disen˜o de la aplica- ci´on: Restricciones de otros est´andares, limitaciones del hardware, etc.

## Atributos del Sistema

Se detallar´an los atributos de calidad (las “ilities”) del sistema: Fiabilidad, mantenibilidad, portabilidad, y, muy importante, la seguridad. Deber´a espe- cificarse qu´e tipos de usuario est´an autorizados, o no, a realizar ciertas tareas, y c´omo se implementar´an los mecanismos de seguridad (por ejemplo, por me- dio de un *login* y una *password* ).

## Otros Requisitos

Cualquier otro requisito que no encaje en otra secci´on.

# Ap´endices

Pueden contener todo tipo de informaci´on 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 an´alisis de costes.
3. Restricciones acerca del lenguaje de programaci´on.