



3. Fases en el desarrollo y ejecución del software

Caso práctico En la reunión de GoyoProg acerca del nuevo proyecto Ada, la supervisora, dejó bien claro que lo primero y más importante es tener claro qué queremos que haga el software y con qué herramientas contamos: lo demás vendría después, ya que si esto no está bien planteado, ese error se propagará a todas las fases del proyecto. —¿Por dónde empezamos? —pregunta Juan. —ANÁLISIS de REQUISITOS —contesta Ada.

Ya hemos visto en puntos anteriores que debemos elegir un modelo de ciclo de vida para el desarrollo de nuestro software.

Independientemente del modelo elegido, siempre hay una serie de etapas que debemos seguir para construir software fiable y de calidad.

Estas etapas son:

1. ANÁLISIS DE REQUISITOS.

Se especifican los requisitos funcionales y no funcionales del sistema.

2. DISEÑO.

Se divide el sistema en partes y se determina la función de cada una.

3. CODIFICACIÓN.

Se elige un Lenguajes de Programación y se codifican los programas.

4. PRUEBAS.

Se prueban los programas para detectar errores y se depuran.

5. DOCUMENTACIÓN.

De todas las etapas, se documenta y guarda toda la información.

6. EXPLOTACIÓN.

Instalamos, configuramos y probamos la aplicación en los equipos del cliente.

7. MANTENIMIENTO.

Se mantiene el contacto con el cliente para actualizar y modificar la aplicación el futuro.

3.1 Análisis

Cuando se habla de una fase del ciclo de vida, **el análisis consiste en producir un documento de especificación de requisitos que describa lo que el sistema debe hacer, pero no cómo hacerlo**. No se trata pues de una actividad sólo de análisis, sino también de síntesis.

Es **la fase de mayor importancia en el desarrollo del proyecto** y todo lo demás dependerá de lo bien detallada que esté. También es la más complicada, ya que no está automatizada y depende en gran medida del analista que la realice.

¿Qué se hace en esta fase?

Se especifican y analizan los requisitos funcionales y no funcionales del sistema. Pero previamente se especificará todo lo relativo a las **organizaciones, participantes en el proyecto, reuniones y objetivos** a partir de los cuales se obtendrán los requisitos para la realización del software.

Se define el análisis de requisitos como **“el proceso del estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema, de hardware o de software, así como el proceso de estudio y refinamiento de dichos requisitos”** (Estándar IEEE Std. 610 [IEEE 1990]).

Requisito es pues **“una condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado”**.

La definición de los requisitos en un proyecto debe ser fruto del trabajo conjunto de las partes involucradas en su desarrollo: jefe de proyecto, analistas, clientes y usuarios. Ningún colectivo antes citado puede redactar la Especificación de Requisitos Software (ERS) de forma individual ya que:

- ✓ El cliente no suele conocer el proceso de diseño y desarrollo del software.
- ✓ Los analistas no entienden completamente el problema del cliente dado que no dominan su área de trabajo.

La fase de análisis de requisitos, según el **estándar IEEE 1074 [IEEE, 1991]** se desglosa en tres grandes actividades:

- ✓ **Definir los requisitos de software**. Tarea iterativa para crear una definición o especificación preliminar de los requisitos que debe cumplir el software a partir de la información obtenida mediante técnicas de recogida de información.
- ✓ **Definir los requisitos de las interfaces del software con el resto del sistema y con el exterior**. Deben definirse las propiedades que se deben satisfacer para obtener una interacción eficaz con otros

elementos del sistema (el usuario, el hardware, otras aplicaciones software, ...). En particular la interfaz con el usuario es crítica para la **facilidad de uso**, y por tanto, el éxito del software. Los requisitos de interfaz con otras aplicaciones deben describir las características para que el software se relacione con ellas, las cuales pueden estar muy influenciadas por **restricciones de trabajo del sistema** (S.O. utilizado, SGBD empleado, compiladores, etc.). Así mismo deben definirse las características de **las interrelaciones con elementos hardware**.

- ✓ **Integrar los requisitos en un documento de especificación y asignarles prioridades.** La asignación de prioridades debe hacerse en función de su importancia o los beneficios que puede aportar su cumplimiento.

Lo fundamental es la **buena comunicación entre el analista y el cliente** para que la aplicación que se va a desarrollar cumpla con sus expectativas. La culminación de esta fase es el **documento ERS (Especificación de Requisitos Software)**. En este documento quedan especificados:

- ✓ La planificación de las **reuniones** que van a tener lugar.
- ✓ Relación de los **objetivos** del usuario cliente y del sistema.
- ✓ Relación de los **requisitos** funcionales y no funcionales del sistema.
- ✓ Relación de **objetivos prioritarios y temporización**.
- ✓ Reconocimiento de **requisitos mal planteados o que conllevan contradicciones**, etc.

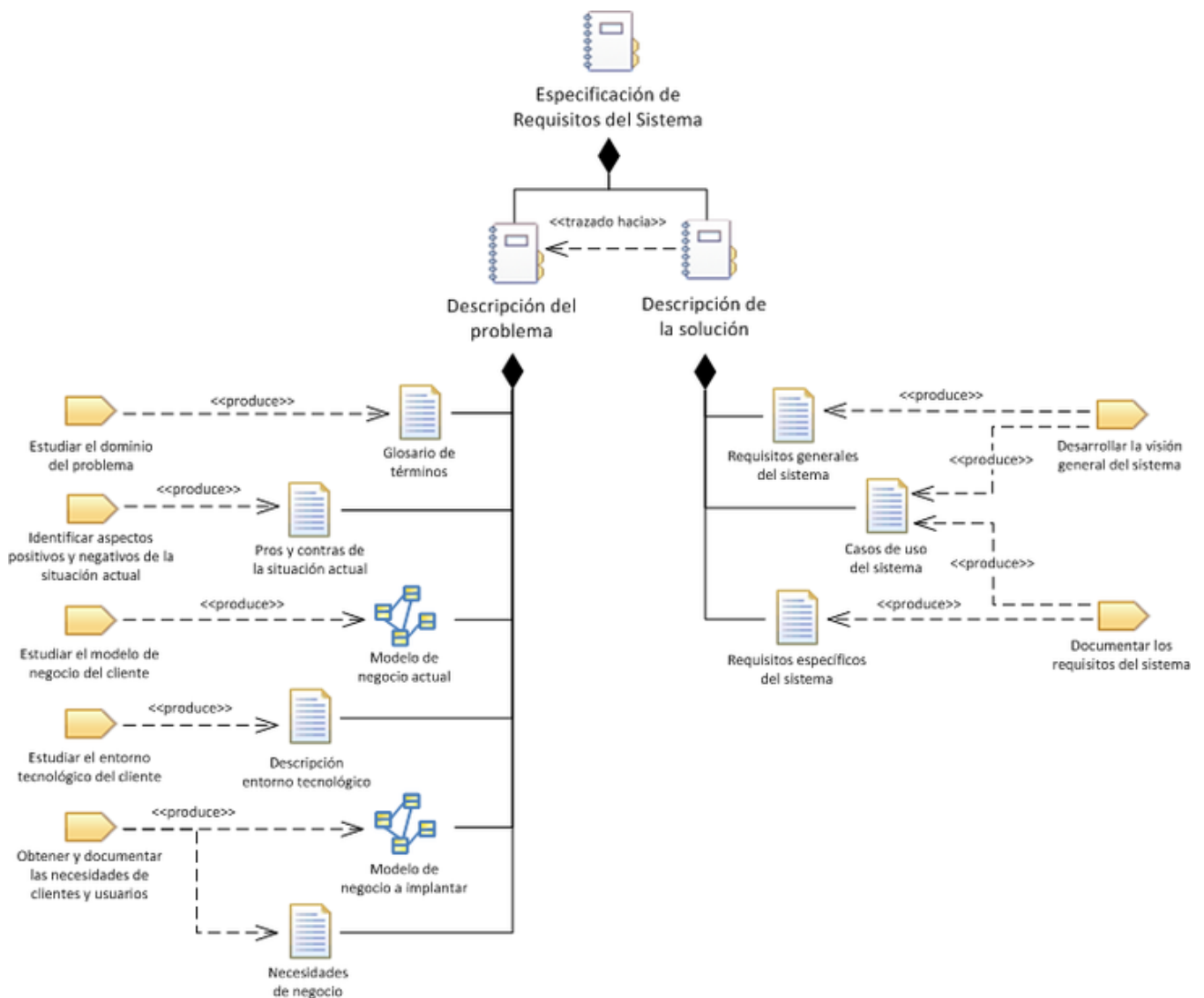
Las características deseables para un buen documento ERS son las siguientes [IEEE 1984B]:

- ✓ No ambigua
- ✓ Completa
- ✓ Correcta
- ✓ Verificable
- ✓ Consistente
- ✓ Modificable
- ✓ Facilidad para identificar el origen y las consecuencias de cada requisito
- ✓ Facilidad de utilización durante la fase de explotación y de mantenimiento

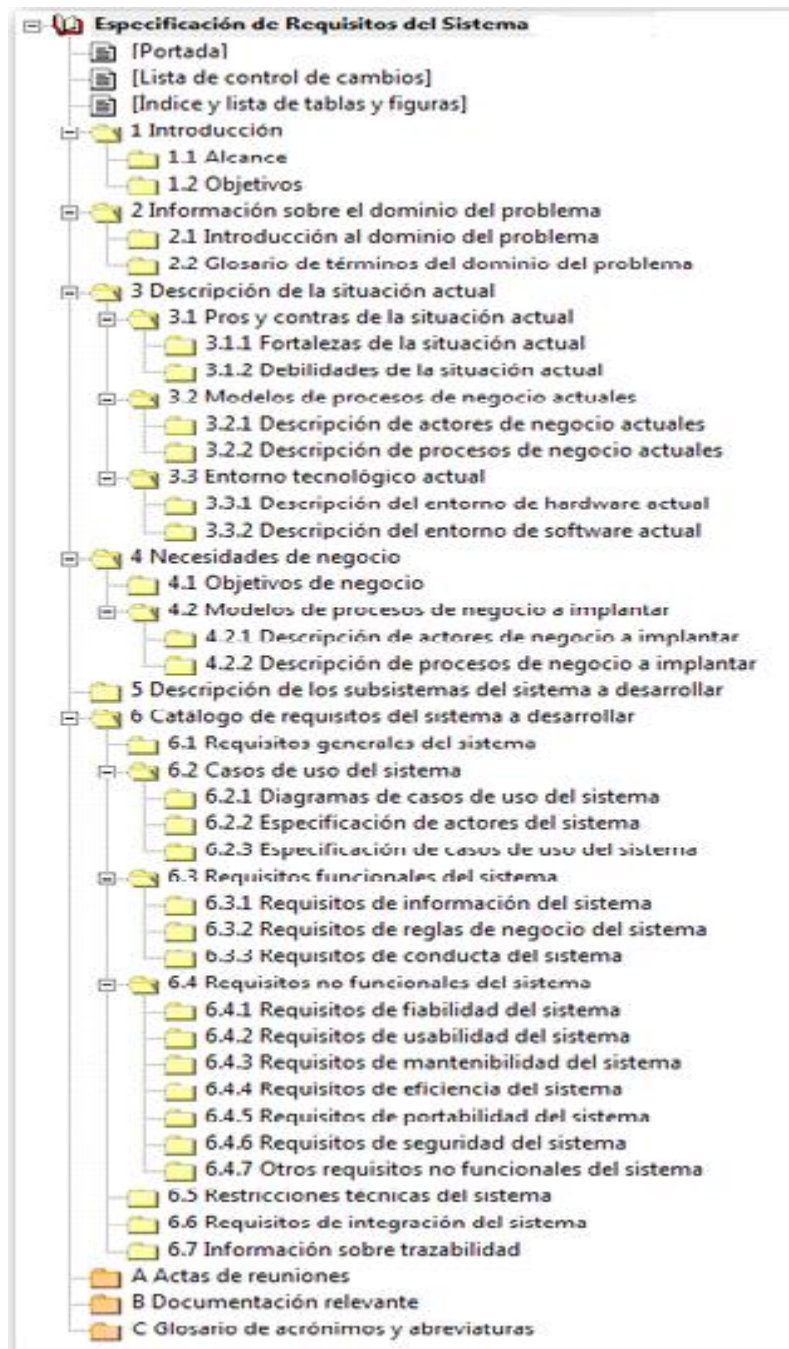
Todo aquello que no se detecte, o resulte mal entendido en la etapa inicial provocará un fuerte impacto negativo en los requisitos, propagando esta corriente degradante a lo largo de todo el proceso de desarrollo e incrementando su perjuicio cuanto más tardía sea su detección.

3.1.1 Documento de Especificación de Requisitos

- ✓ El objetivo principal de la *Especificación de Requisitos del Sistema (ERS)* es servir como **medio de comunicación** entre clientes, usuarios, ingenieros de requisitos y desarrolladores. En la ERS **deben recogerse tanto las necesidades de clientes y usuarios** (*necesidades del negocio*, también conocidas como *requisitos de usuario*, *requisitos de cliente*, *necesidades de usuario*, etc.) **como los requisitos que debe cumplir el sistema software a desarrollar para satisfacer dichas necesidades** (*requisitos del producto*, también conocidos como *requisitos de sistema* o *requisitos software*).
- ✓ La estructura detallada de la ERS puede verse a continuación, en la que también se muestran las tareas que producen cada uno de sus contenidos. **El impacto en la ERS consiste en mejorar la calidad de su contenido y no en producirlo directamente**, la calidad de los requisitos se consigue analizando los requisitos, verificando su calidad y validándolos uno a uno.



✓ Índice de la Especificación de Requisitos Formal



✓ Buenas prácticas y recomendaciones de uso

- La **portada** de la ERS, independientemente de la maqueta concreta, debe incluir los siguientes elementos:

- **Nombre del documento:** Especificación de Requisitos del Sistema.
- **Nombre del proyecto:** el nombre del proyecto en el que se desarrolla la ERS.
- **Versión:** versión de la ERS según la numeración de versiones que indique la Gestión de la Configuración del proyecto. Lo habitual es que **la versión se componga de dos números**. El primero indica el número de la línea base, que debe ser cero mientras no se haya entregado la

primera línea base y que se debe incrementar con cada nueva línea base que se vaya entregando formalmente. El segundo número, que debe ponerse a cero cada vez que se incremente el primero, indica cambios dentro una línea base aún no entregada. En otras palabras, la primera línea base será la versión 1.0 del documento, la segunda línea base la 2.0 y así sucesivamente. Una versión como la 2.5 será la quinta versión intermedia a partir de la segunda línea base.

- **Fecha:** fecha de la publicación de la versión.
- **Desarrollado por:** nombre de la empresa o equipo de desarrollo de la ERS.
- **Desarrollado para:** nombre del cliente.

- **El índice:** debe indicar la página en la que comienza cada sección, subsección o apartado del documento. En la medida de lo posible, se sangrarán las entradas del índice para ayudar a comprender la estructura del documento. En el caso de documentos electrónicos, cada entrada podrá ser un hiperenlace al apartado correspondiente.

- **Ámbito:** Esta sección obligatoria debe contener una **descripción breve de la situación que genera la necesidad del nuevo desarrollo** y cualquier otra consideración que sitúe al posible lector en el contexto oportuno para comprender el resto del documento.

- **Alcance:** Esta sección debe describir a qué elementos afecta el desarrollo del nuevo sistema.

- **Objetivos:** Esta sección debe describir **los principales objetivos que se esperan alcanzar** cuando el sistema a desarrollar esté en producción.

- **Glosario de términos:** Esta sección debe contener una lista ordenada alfabéticamente de los principales términos, acrónimos y abreviaturas específicos del dominio del problema, especialmente de los que se considere que su significado deba ser aclarado. Cada término, acrónimo o abreviatura deberá acompañarse de su definición y se podrá adjuntar material multimedia que facilite su comprensión como fotografías, documentos escaneados, diagramas o incluso vídeo o sonido en el caso de que el formato de la ERS lo permita.

Ejemplo término: Merchandising: Ventas accesorias de un producto. Por ejemplo, el merchandising del Fútbol Club son las gorras, las camisetas, etc. que venden en locales comerciales o en los eventos del club.

-Descripción de la situación actual [opcional]

Esta sección opcional debe contener información sobre la situación actual de la organización para la que se va a desarrollar el sistema software. En concreto, **debe contener información sobre los pros y contras de la situación actual, necesidades del modelo de negocio, sobre los modelos de proceso de negocio actuales y sobre el entorno tecnológico actual de la organización.**

Ejemplo fortaleza:

Los usuarios se encuentran muy cómodos al usar los menús de la aplicación actual.

Ejemplo debilidad:

La entrega de un pedido se suele retrasar en torno a dos días.

Ejemplo de Descripción textual de un proceso de negocio actual (puede incluir un esquema)

Actualmente, cuando un cliente quiere realizar una reserva para sus vacaciones lo primero que estudiamos es el destino o los posibles destinos. A continuación, es necesario conocer cuántas personas van a viajar y las edades de las mismas. Si el número de personas es suficiente para constituir un grupo se pasará la tramitación al Departamento de Viajes en grupo. Una vez que el cliente ha decidido el destino, se consultará la disponibilidad de hotel en las fechas indicadas y de medios de transporte. A veces, incluso habiendo reservado el hotel tenemos que cambiar el destino por falta de billete. Dentro de este proceso, la información

se toma de fuentes distintas. Hay información disponible vía web en páginas privadas para consulta de agencias de viajes y otras mediante acceso remoto (a las operadoras de vuelo, por ejemplo) también hay casos en los que es necesario recurrir a una llamada telefónica.

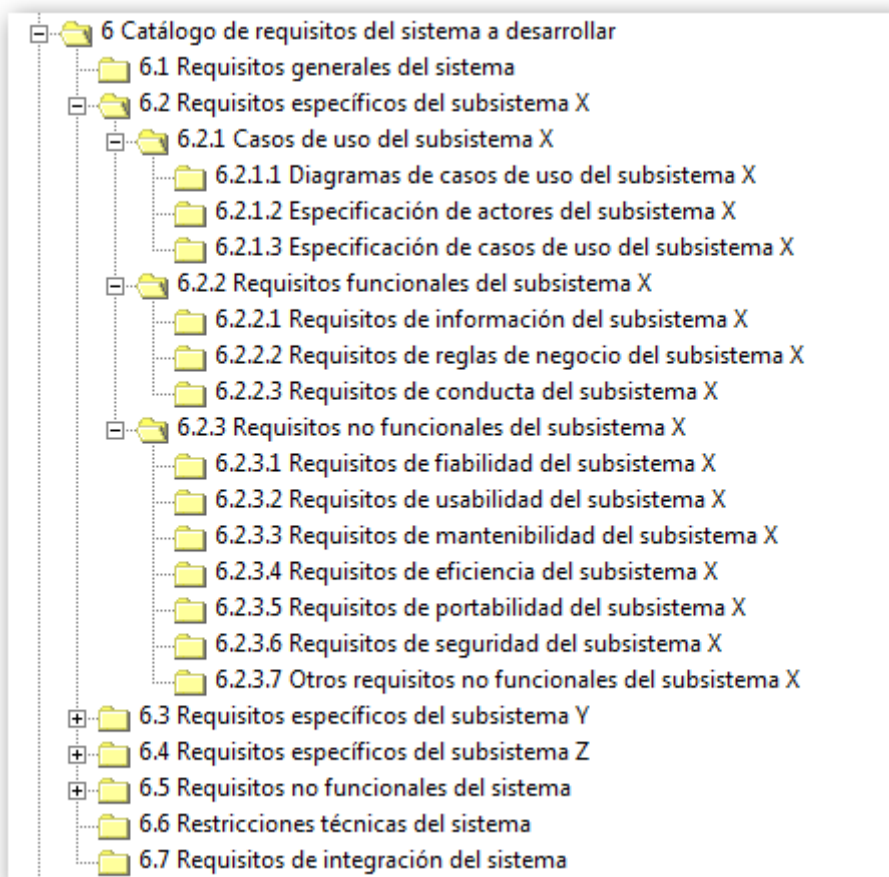
- **Entorno tecnológico actual:** Esta sección debe contener información general sobre el entorno tecnológico en la organización del cliente antes del comienzo del desarrollo del sistema software, incluyendo hardware, redes, software, etc. Se prestará especial atención a la arquitectura de servicios.

Ejemplo de Descripción del entorno de hardware actual

El entorno de hardware actual se compone de un servidor Sun Fire X4600 M2 y de una Intranet de 25 PCs. El entorno de software actual se compone de Solaris 10 y Oracle19c en el servidor y software de aplicación NominaPlus v5 en los 25 PCs de la Intranet. La aplicación actual de gestión de nóminas es una aplicación web funcionando sobre OC4J en el servidor.

- Catálogo de requisitos del sistema a desarrollar

Esta sección obligatoria debe contener la descripción de la solución que el jefe de proyecto propone al cliente para satisfacer sus necesidades de negocio.



- Requisitos generales del sistema

Esta sección debe contener la especificación de los *requisitos generales del sistema*.

Ejemplo de Requisito general del sistema

El sistema deberá registrar los accesos al edificio mediante el uso de lectores de tarjetas magnéticas.

Ejemplo de Requisitos generales del sistema (jerarquizados)

RG-0001 Gestionar los recursos del centro deportivo (padre)

RG-0001.01 Gestionar las reservas de las pistas del centro

RG-0001.02 Gestionar los horarios semanales de las actividades deportivas que se realizan en las salas del centro

RG-0001.03 Gestionar el torneo de pádel de verano y la liguilla de fútbol de otoño

- Requisitos funcionales del sistema

Esta sección debe contener los requisitos funcionales del sistema que se hayan identificado a partir de los requisitos generales. Se dividen en los siguientes tipos:

- **Requisitos de información del sistema:** Estos requisitos deben especificar qué información debe almacenar el sistema para poder ofrecer la funcionalidad descrita en los casos de uso del sistema o en otros requisitos.

Ejemplo de Requisito de información del sistema

El sistema deberá almacenar información sobre los perros de caza que componen una rehala. En concreto, el código d*el chip de identificación, el nombre al que responde, la raza, la fecha de nacimiento y la ubicación actual (finca o similar)

- **Requisitos de reglas de negocio del sistema:** Estos requisitos deben especificar qué reglas de negocio debe respetar el sistema, evitando que se incumplan durante su funcionamiento.

Ejemplo de Requisito de regla de negocio del sistema

El sistema deberá respetar la siguiente regla de negocio: un socio del club deportivo podrá reservar un máximo de 5 pistas a la semana y una única pista el mismo día a la misma hora.

- **Requisitos de conducta del sistema:** Estos requisitos deben especificar cualquier otro comportamiento deseado del sistema que no se haya especificado mediante los casos de uso del sistema, como generación de informes, funcionalidades transversales a varios casos de uso del sistema, etc.

Ejemplo de Requisito de conducta del sistema

El sistema deberá generar, a petición del usuario, un informe mensual de préstamos y devoluciones de fondos de la biblioteca.

- Requisitos no funcionales del sistema

Esta sección debe contener los *requisitos no funcionales* que se hayan identificado. Se dividen en los siguientes tipos:

- **Requisitos de fiabilidad:** Estos requisitos deberán establecer, de la manera más objetiva y medible posible, los niveles que debe cumplir el sistema a desarrollar en aspectos como recuperabilidad y tolerancia a fallos.

Ejemplo de Requisito de fiabilidad

El sistema deberá tardar un máximo de 10 minutos para la recuperación de un fallo de caída total, en el 95% de las ocasiones.

- **Requisitos de usabilidad:** Estos requisitos deberán establecer, de la manera más objetiva y medible posible, los niveles que debe cumplir el sistema a desarrollar en aspectos como facilidad de aprendizaje, comprensión, operatividad y atractividad.

Ejemplo de Requisito de usabilidad

El sistema deberá permitir en el 80% de las veces que con un máximo de 5 clicks sea suficiente para llegar a la información deseada.

- **Requisitos de eficiencia o rendimiento:** Estos requisitos deberán establecer, de la manera más objetiva y medible posible, los niveles que debe cumplir el sistema a desarrollar en aspectos como tiempo de respuesta.

Ejemplo de Requisito de rendimiento

El sistema deberá tener un tiempo máximo de respuesta de 5 segundos para cualquier operación de consulta.

- **Requisitos de mantenibilidad:** Estos requisitos deberán establecer, de la manera más objetiva y medible posible, los niveles que debe cumplir el sistema a desarrollar en aspectos como estabilidad, facilidad de análisis, facilidad de cambio, facilidad de pruebas.

Ejemplo de Requisito de mantenibilidad

El código fuente que se implemente en JAVA deberá cumplir las recomendaciones de Code Conventions for the Java Programming Language (pautas para la calidad estructural del código)

- **Requisitos de portabilidad:** Estos requisitos deberán establecer, de la manera más objetiva y medible posible, los niveles que debe cumplir el sistema a desarrollar en aspectos relacionados con la escalabilidad: capacidad de instalación, capacidad de sustitución, adaptabilidad, coexistencia, compatibilidad con hardware o software, etc.

Ejemplos de Requisito de portabilidad

El sistema deberá evitar el uso de extensiones propietarias al estándar SQL-92 en el sistema de gestión de bases de datos que utilice.

- **Requisitos de seguridad:** Estos requisitos deberán establecer, de la manera más objetiva y medible posible, los niveles que debe cumplir el sistema a desarrollar en aspectos como accesos al sistema, identificación y autenticación, protección de datos y privacidad.

Ejemplo de Requisito de seguridad

El sistema deberá ser capaz de evitar ataques de inyección de SQL sistemáticos.

- **Otros requisitos no funcionales:** Esta sección debe contener los requisitos no funcionales que se hayan identificado y que no pertenezcan a ninguna de las categorías anteriores.

- **Restricciones técnicas del sistema:** Esta sección debe contener las restricciones técnicas que se imponen al sistema software a desarrollar (tecnología a usar, protocolos de comunicaciones, compatibilidad con navegadores, etc.).

Ejemplo de Requisitos de restricción técnica

El sistema deberá ser compatible con los navegadores Internet Edge y Mozilla Firefox en su última versión.

El sistema deberá utilizar como sistema de gestión de bases de datos Oracle 19c.

- **Requisitos de integración del sistema:** Estos requisitos deben identificar aquellos servicios disponibles en el entorno tecnológico de producción o componentes software (por ejemplo, bibliotecas enlazables) cuya funcionalidad sea relevante para el sistema a desarrollar y deban ser utilizados por el mismo.

Ejemplo de Requisito de integración del sistema

El sistema deberá utilizar el servicio @firma para todos los aspectos relacionados con validación y firma electrónica.

- **Información sobre trazabilidad[NO IMPLEMENTAR]:** Esta sección obligatoria debe contener el conjunto de *matrices de trazabilidad* que se considere oportuno para identificar las dependencias entre los diferentes elementos que aparecen en la ERS. Una RTM (matriz de trazabilidad de requisitos) contiene el seguimiento de todos los requisitos, dependencias, riesgos y pruebas, también ayuda a garantizar que los requisitos recopilados definan realmente el sistema requerido por los usuarios.

Anexos [opcional]

Los anexos se usarán para proporcionar información adicional a la documentación obligatoria del documento. Sólo deben aparecer si se consideran oportunos y se identificarán con letras ordenadas alfabéticamente: A, B, C, etc.

A continuación, se describen algunos anexos habituales.

Anexo A Actas de reuniones: Este anexo debe contener el catálogo de actas de reuniones que se hayan mantenido, registradas mediante el documento para acta de reuniones.

Anexo B Documentación relevante: Este anexo debe contener cualquier documentación que se considere relevante para el sistema a desarrollar. Por ejemplo, documentos que deriven de la actividad normal del negocio, leyes o referencias a leyes de aplicación en la organización, fotografías que ilustren la forma de trabajar, informes que genera el software actual, etc.

Anexo C Glosario de acrónimos y abreviaturas: Este anexo debe contener una lista ordenada alfabéticamente de los acrónimos y abreviaturas que aparezcan en el documento. Para facilitar la reutilización entre proyectos, los acrónimos y abreviaturas comunes a la mayoría de los proyectos aparecerán en este glosario separados de los términos específicos del dominio del proyecto software a desarrollar.