

Ingeniería del Software:

ERS SIGREM v.2

SIGREM: Sistema Integral de Gestión de Recursos de Multas

Autores:

- Félix Martos Trenado
- Javier Rodríguez Horcajo
- Miguel Ángel Ruiz Ortega
- Francisco José Sánchez Hernando
- Sergio Tarancón Faus

Índice

1 Introducción

- 1.1 Propósito
- 1.2 Alcance
- 1.3 Definiciones, abreviaturas y acrónimos
- 1.4 Referencias
- 1.5 Resumen

2 Descripción general

- 2.1 Perspectiva del producto
- 2.2 Funciones del producto
- 2.3 Características del usuario
- 2.4 Restricciones
- 2.5 Supuestos y dependencias
- 2.6 Requisitos futuros

3 Requisitos específicos

- 3.1 Interfaces externas
- 3.2 Requisitos funcionales
- 3.3 Requisitos de rendimiento
- 3.4 Requisitos de base de datos lógica
- 3.5 Restricciones de diseño
- 3.6 Atributos del sistema software

1 Introducción

La presente especificación de requisitos pertenece al desarrollo del proyecto de la asignatura Ingeniería del Software, de la Facultad de Informática de la UCM. y está desarrollada siguiendo las directrices de la *IEEE Guide to Software Requirements Specifications* especificada en el IEEE Std. 830-1998.

1.1 Propósito

1.2 Alcance

1.3 Definiciones, abreviaturas y acrónimos

1.4 Referencias

1.5 Resumen

1.1 Propósito

El objeto de la especificación es definir de manera clara y precisa todos los requerimientos, funcionalidades y restricciones del sistema a construir. El documento tiene como audiencia al equipo de desarrollo, al cliente y a los usuarios finales del sistema. Será el canal de comunicación entre las partes implicadas. Está sujeta a revisiones por los desarrolladores y el cliente, que se recogerán por medio de sucesivas versiones del documento, hasta alcanzar su aprobación. Cuando eso ocurra, servirá de base al equipo de desarrollo para la construcción del nuevo sistema.

1.2 Alcance

El propósito del diseño y desarrollo de SIGREM (*Sistema Integral de **G**estión de **RE**curso**s** de **M**ultas*) es crear un producto que satisfaga las necesidades de una empresa de gestión de recursos de multas de tráfico, cuyo principal servicio será prestar asesoramiento jurídico a sus clientes. Éstos podrán crear contratos con la empresa que harán referencia a un vehículo sobre el que pesa alguna multa, y a su vez podrán solicitar un recurso para intentar anularla. La empresa asignará entonces un abogado de su plantilla a dicho recurso para intentar resolverlo favorablemente.

La aplicación desarrollada permitirá:

- Gestionar los contratos con los clientes, ya sea para crearlos, modificarlos o cancelarlos
- Gestionar la asignación de tareas a los empleados
- Gestionar la plantilla
- Gestionar la facturación a los clientes
- Controlar la gestión de nóminas de los empleados
- Gestionar los datos de empleados y clientes
- Gestionar los contratos asociados a cada cliente
- Crear, consultar y modificar las multas asociadas a un contrato
- Crear, consultar y modificar los recursos asociados a las multas.

Beneficios

La introducción de SIGREM facilitará la gestión de la empresa, tanto en lo referente a la asignación de tareas a los empleados, como en la relación con los clientes, y en el control de la facturación.

En la versión web de SIGREM también se permitirá a los clientes y a los abogados de la empresa acceder, de manera inmediata y por medio de una conexión remota a través de una página web, a los datos relacionados con su perfil de usuario.

1.3 Definiciones, acrónimos y abreviaturas

- SIGREM: Sistema Integral de Gestión de Recursos de Multas
- DTD: *Definición del Tipo de Documento*. Declaración en un documento XML que especifica restricciones en la estructura del mismo. Define los tipos de elementos, atributos y entidades permitidas, y puede expresar algunas limitaciones para combinarlos
- ERS: Especificación de Requisitos Software
- Java: Lenguaje de programación de Sun Microsystems. Se entiende que nos referimos a la versión 1.4 o superior
- JSP: *Java Server Pages*. Páginas de Servidor Java
- JVM: *Java Virtual Machine*. Máquina Virtual Java de Sun Microsystems
- MySQL: Servidor de bases de datos relacional compatible, en gran parte, con el estándar ANSI SQL 99. Suponemos que nos referimos a la versión 4 o superior
- Servlet: Aplicación web en Java que corre del lado del servidor
- SGBD: Sistema Gestor de Bases de Datos. En este caso concreto se trata de MySQL
- Tomcat: Servidor de aplicaciones Java. Es un contenedor de Servlets con un entorno JSP
- XHTML: *eXtensible Hypertext Markup Language*. Lenguaje extensible de marcas e hipertexto Reformulación del estándar HTML 4 en forma de lenguaje XML
- XML: *eXtensible Markup Language*. Lenguaje extensible de marcas. Es un metalenguaje, que se utiliza para definir formatos semánticos de otros lenguajes
- Cliente: Persona física o jurídica que realiza un contrato con la empresa de gestión de recursos de multas para recurrir alguna multa
- Empleado: Abogado de la empresa de gestión de recursos de multas que representa a los clientes en los recursos
- Contrato: Acuerdo por el cual la empresa de gestión de recursos de multas se obliga a prestar un servicio a un cliente, bajo dependencia o subordinación y a cambio de una retribución económica

- Gestor: Empleado de la empresa que se encarga de la gestión global de la aplicación y del control de la persistencia de los datos
- Recurso: Trámites realizados por los abogados empleados de la empresa en nombre de los clientes para intentar eliminar una sanción
- Tasa de éxito: Porcentaje de los recursos presentados en tiempo y forma adecuados que terminan con la retirada de la sanción
- Nómina: Descripción detallada de los conceptos y cantidades retribuidos a los empleados de la empresa
- Facturación: Ganancias obtenidas por la empresa del cobro de sus servicios a los clientes
- Minutaje: Tiempo que los abogados han dedicado a los recursos asignados, y que servirá para calcular su nómina

1.4 Referencias

Para la redacción de este documento se han tenido en cuenta los siguientes documentos:

1. IEEE Std 830-1994 *IEEE Guide to Software Requirements Specifications*. IEEE Standards Board.
2. Apuntes de la asignatura *Ingeniería del Software* de Ingeniería Informática. Antonio Navarro, 2004. Web de la Facultad de Informática de la UCM
3. Roger S. Pressman. *“Ingeniería del Software. Un enfoque práctico”*. 5ª ed. 2002. McGraw-Hill

1.5 Resumen

Este documento consta de tres secciones.

- Esta Sección 1 es la *Introducción* y proporciona una visión general de la ERS.
- En la Sección 2, llamada *Descripción General*, se da una amplia visión general del sistema, con el fin de conocer las principales funciones que debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles.
- La Sección 3, *Requisitos Específicos*, define detalladamente los requisitos que debe satisfacer el sistema, así como sus restricciones, y sus características más detalladas.

2 Descripción general

En esta sección se presenta una descripción a alto nivel del sistema. Se presentarán las principales áreas de negocio a las cuales el sistema debe dar soporte, las funciones que el sistema debe realizar, la información utilizada, las restricciones y otros factores que afecten al desarrollo del mismo.

2.1 Perspectiva del producto

2.2 Funciones del producto

2.3 Características del usuario

2.4 Restricciones

2.5 Supuestos y dependencias

2.6 Requisitos futuros

2.1 Perspectiva del producto

El presente producto está pensado para ser implementado en dos fases:

1. En una primera versión funcionará como una aplicación Java *standalone* monousuario y sin soporte de base de datos externa. La persistencia de los datos se llevará a cabo mediante ficheros de texto en formato XML. Podrá funcionar en cualquier equipo con una JVM 1.4 o superior.
2. En una segunda versión se transformará en una aplicación web, utilizando JSP y servlets, y el servidor Tomcat 4. Será multiusuario y con soporte de base de datos MySQL.

El producto en principio no interactuará con ningún otro sistema, aunque podrán exportarse sus datos a diferentes formatos electrónicos. Para ello se utilizará XML estándar. Tendrá una interfaz visual cómoda y fácil de utilizar por los usuarios

2.2 Funciones del producto

Las funcionalidades prestadas por el producto desarrollado son las siguientes:

- Permitir la creación, modificación consulta y cancelación de los contratos de los clientes
- Añadir, modificar, consultar y eliminar multas a los contratos
- Facilitar la asignación de los abogados contratados por la empresa a los recursos interpuestos por los clientes sobre sus multas
- Añadir, consultar, modificar y eliminar recursos a las multas asociadas a los contratos de los clientes
- Agilizar el proceso de facturación de la empresa en relación a los contratos establecidos con los clientes
- Controlar la gestión de nóminas de los empleados
- Consultar y modificar los datos de los empleados de la empresa y de los clientes
- Gestionar la plantilla de la empresa

2.3 Características del usuario

Los usuarios del producto no necesitan tener ningún tipo de experiencia previa para poder utilizarlo correctamente. Sólo deberán saber interactuar adecuadamente con la interfaz gráfica, que a su vez será amigable y de fácil manejo.

Dado que la primera versión del producto será monousuario, éste deberá tener un amplio conocimiento del funcionamiento de la empresa para poder hacer uso de todas las funcionalidades de la aplicación.

En la versión web del producto habrá tres tipos de usuarios, que podemos agrupar en dos clases:

- Administrador: tendrá acceso a todas las funcionalidades y a todos los datos de la aplicación. Deberá tener un amplio conocimiento del funcionamiento de la empresa.
- Clientes y Abogados: podrán acceder y modificar únicamente la parte de los datos relacionados con su perfil de usuario. Las operaciones que podrán realizar en la aplicación estarán restringidas.

La comunicación con la aplicación para cada uno de ellos será por medio de una interfaz gráfica de fácil manejo, y que restringirá las operaciones permitidas a cada tipo de usuario.

2.4 Restricciones

La aplicación está programada en lenguajes independientes de la plataforma hardware. Por tanto sus restricciones en este sentido son mínimas.

La persistencia de datos en formato XML permitirá que el producto desarrollado pueda comunicarse con otras aplicaciones, pero éstas en ningún caso impondrán restricciones adicionales a la ejecución de la propia aplicación.

Se controlarán los intentos de acceso no autorizados a información sensible.

2.5 Supuestos y dependencias

Supondremos que los requisitos descritos en la ERS serán estables una vez que hayan sido aprobados por la empresa a la que va dirigida el producto. Cualquier petición de cambio en la ERS debe ser aprobada por todas las partes implicadas y gestionada por el grupo de desarrollo.

La aplicación monousuario podrá funcionar en cualquier equipo con una JVM 1.4 o superior. Será, por tanto, casi totalmente independiente del sistema operativo en el que se ejecute. La versión web dependerá de un servidor Tomcat 4 y MySQL 4 para tener acceso a los datos. La parte cliente sólo necesitará un navegador compatible con el estándar XHTML 1.0. para poder acceder correctamente a la interfaz de la aplicación.

2.6 Requisitos futuros

En la segunda versión del producto, al convertirse en una aplicación multiusuario, tendrá que garantizarse la correcta gestión de la concurrencia de usuarios. Además, existirán tres perfiles de usuario que podrán acceder a la aplicación:

- los *abogados* contratados por la empresa
- los *clientes* que soliciten los servicios de la empresa
- los *administradores* pertenecientes a la empresa

Cada uno de ellos interactuará con el producto de una forma distinta, y tendrá únicamente acceso a la parte de la aplicación asociada con su perfil. Por lo tanto, sus interfaces de comunicación con el producto serán diferentes.

3 Requisitos específicos

En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por el sistema. Todos los requisitos aquí expuestos son esenciales, es decir, no sería aceptable un sistema que no satisfaga alguno de los requisitos aquí presentados. Estos requisitos se han especificado teniendo en cuenta, entre otros, el criterio de “testabilidad”: dado un requisito, debería ser fácilmente demostrable si es satisfecho o no por el sistema.

3.1 Interfaces externas

3.2 Requisitos funcionales

3.3 Requisitos de rendimiento

3.4 Requisitos de base de datos lógica

3.5 Restricciones de diseño

3.6 Atributos del sistema software

3.1 Interfaces Externas

3.1.1 Interfaz de Usuario

En una primera versión nuestra aplicación funcionará bajo un entorno de ventanas, y dentro de éste, la interfaz de comunicación con el usuario se mostrará en forma de formularios con ventanas para introducir datos en los respectivos campos. A todo ello se tendrá acceso mediante el uso de teclado y ratón.

En una segunda versión se convertirá en una aplicación cliente/servidor con salida web. De modo que los usuarios podrán acceder a ella mediante un navegador compatible con el estándar XHTML 1.0

3.1.2 Interfaz del Hardware

No se ha definido.

3.1.3 Interfaz del Software

En esta versión no se ha definido.

3.1.4 Interfaces de Comunicaciones

En esta versión no se ha definido.

3.2 Requisitos funcionales

Describiremos los requisitos siguiendo una organización en función de *Estímulo/Respuesta funcional*:

3.2.1 Gestión de clientes

Requisito 1.1

función: añadir cliente

descripción: añade un nuevo cliente a la base de datos

entrada: nombre, dni, direccion, cp, poblacion, provincia, telefono1, telefono2, movil, email, fax

salida: código del cliente

origen: operador consola

destino: sistema

necesita: base de datos de clientes

precondición: no exista el cliente en la base de datos
postcondición: cliente añadido a la base de datos
efectos laterales: ninguno

Requisito 1.2

función: eliminar cliente
descripción: elimina un cliente de la base de datos
entrada: código del cliente
salida: ninguna
origen: operador consola
destino: sistema
necesita: base de datos de clientes
precondición: exista el cliente en la base de datos
postcondición: cliente eliminado de la base de datos
efectos laterales: se archivan los contratos asociados al cliente y todos los datos que dependan de ellos

Requisito 1.3

función: modificar cliente
descripción: modifica los datos de un cliente
entrada: código cliente
salida: ninguna
origen: operador consola
destino: sistema
necesita: base de datos de clientes
precondición: exista el cliente en la base de datos
postcondición: cliente modificado y almacenado en la base de datos
efectos laterales: ninguno

Requisito 1.4

función: consultar cliente por código
descripción: consulta los datos de un cliente
entrada: código del cliente
salida: datos del cliente elegido
origen: operador consola

destino: sistema
necesita: base de datos de clientes
precondición: exista el cliente en la base de datos
postcondición: mostrar datos del cliente
efectos laterales: ninguno

Requisito 1.5

función: consultar cliente por nombre
descripción: consulta los datos de un cliente
entrada: nombre del cliente
salida: datos del cliente elegido
origen: operador consola
destino: sistema
necesita: base de datos de clientes
precondición: exista el cliente en la base de datos
postcondición: mostrar datos del cliente
efectos laterales: ninguno

Requisito 1.6

función: consultar cliente por NIF/CIF
descripción: consulta los datos de un cliente
entrada: NIF/CIF del cliente
salida: datos del cliente elegido
origen: operador consola
destino: sistema
necesita: base de datos de clientes
precondición: exista el cliente en la base de datos
postcondición: mostrar datos del cliente
efectos laterales: ninguno

3.2.2 Gestión de contratos**Requisito 2.1**

función: añadir contrato
descripción: añade un nuevo contrato asociado a un cliente a la base de datos
entrada: código del cliente, datos del vehículo
salida: código del contrato
origen: operador consola

destino: sistema
necesita: nada
precondición: exista el cliente y no exista contrato asociado al vehículo
postcondición: contrato añadido a la base de datos
efectos laterales: ninguno

Requisito 2.2

función: eliminar contrato
descripción: elimina un contrato de la base de datos
entrada: código del contrato
salida: ninguna
origen: operador consola
destino: sistema
necesita: base de datos de contratos
precondición: exista el contrato en la base de datos
postcondición: contrato eliminado de la base de datos
efectos laterales: se archivan las multas asociadas al contrato.

Requisito 2.3

función: modificar contrato
descripción: modifica los datos de un contrato existente
entrada: código del contrato
salida: ninguna
origen: operador consola
destino: sistema
necesita: base de datos de contratos
precondición: exista el contrato en la base de datos
postcondición: contrato modificado y almacenado en la base de datos
efectos laterales: ninguno

Requisito 2.4

función: consultar contrato por código
descripción: consulta un contrato de la base de datos
entrada: código del contrato
origen: operador consola
destino: sistema
necesita: base de datos de contratos

precondición: exista el contrato en la base de datos
postcondición: mostrar contrato
efectos laterales: ninguno

Requisito 2.5

función: consultar contrato por matrícula
descripción: consulta un contrato de la base de datos
entrada: matrícula
origen: operador consola
destino: sistema
necesita: base de datos de contratos
precondición: exista el contrato en la base de datos
postcondición: mostrar contrato
efectos laterales: ninguno

3.2.3 Gestión de multas**Requisito 3.1**

función: añadir multa
descripción: añade una multa a la lista de multas
entrada: código del contrato, expediente, boletin, fechaDenuncia, infraccion, descripcion
salida: código de la multa
origen: operador consola
destino: sistema
necesita: base de datos de multas
precondición: no exista la multa en la base de datos
postcondición: multa añadida a la base de datos
efectos laterales: ninguno

Requisito 3.2

función: eliminar multa
descripción: elimina una multa
entrada: código de la multa
salida: ninguna
origen: operador consola
destino: sistema
necesita: base de datos de multas
precondición: exista la multa en la base de datos

postcondición: multa eliminada de la base de datos
efectos laterales: si hay recursos sobre esta multa se archivarán también

Requisito 3.3

función: modificar multa
descripción: modifica los datos de una multa existente
entrada: código de la multa
salida: ninguna
origen: operador consola
destino: sistema
necesita: base de datos de multas
precondición: exista la multa en la base de datos
postcondición: multa modificada y almacenada en la base de datos
efectos laterales: ninguno

Requisito 3.4

función: consultar multa por código
descripción: consulta una multa de la base de datos
entrada: código de la multa
salida: datos de la multa
origen: operador consola
destino: sistema
necesita: base de datos de multas
precondición: exista la multa en la base de datos
postcondición: mostrar multa
efectos laterales: ninguno

Requisito 3.5

función: consultar multa por expediente
descripción: consulta una multa de la base de datos
entrada: número de expediente
salida: datos de la multa
origen: operador consola
destino: sistema
necesita: base de datos de multas
precondición: exista la multa en la base de datos
postcondición: mostrar multa

efectos laterales: ninguno

Requisito 3.6

función: consultar multa por boletín
descripción: consulta una multa de la base de datos
entrada: número de boletín
salida: datos de la multa
origen: operador consola
destino: sistema
necesita: base de datos de multas
precondición: exista la multa en la base de datos
postcondición: mostrar multa
efectos laterales: ninguno

3.2.4 Gestión de recursos

Requisito 4.1

función: añadir recurso
descripción: añade un recurso a una multa
entrada: código de la multa, fechaEmision, escritoRecibido, escritoPresentado, estado, abogado, descripcion
salida: código del recurso
origen: operador consola
destino: sistema
necesita: base de datos de multas
precondición: exista la multa
postcondición: recurso añadido a la multa
efectos laterales: ninguno

Requisito 4.2

función: eliminar recurso
descripción: elimina un recurso asociado a una multa
entrada: código del recurso
salida: ninguna
origen: operador consola
destino: sistema
necesita: base de datos de contratos
precondición: exista el contrato

postcondición: recurso eliminado de la multa

efectos laterales: elimina la asignación del recurso al abogado que lo tramita

Requisito 4.3

función: modificar recurso

descripción: actualiza la situación de un recurso a una multa

entrada: código del recurso

salida: ninguna

origen: operador consola

destino: sistema

necesita: base de datos de recursos

precondición: exista el recurso

postcondición: actualización de datos del recurso

efectos laterales: ninguno

Requisito 4.4

función: consultar recurso

descripción: consulta la situación de un recurso

entrada: código del recurso

salida: datos del recurso elegido

origen: operador consola

destino: sistema

necesita: base de datos de recursos

precondición: exista el recurso

postcondición: mostrar situación del recurso

efectos laterales: ninguno

3.2.5 Gestión económica

Requisito 5.1

función: facturar contratos

descripción: calcula la facturación de la empresa

entrada: lista de contratos

salida: facturación de la empresa

origen: operador consola

destino: sistema

necesita: base de datos de contratos

precondición: ninguna

postcondición: mostrar facturación
efectos laterales: ninguno

Requisito 5.2

función: calcular nóminas
descripción: calcula los pagos a realizar a los empleados
entrada: lista de empleados, datos de pago de los empleados
salida: nóminas de la empresa
origen: operador consola
destino: sistema
necesita: base de datos de empleados
precondición: ninguna
postcondición: mostrar nóminas
efectos laterales: ninguno

Requisito 5.3

función: gestión de balance
descripción: calcula el balance de la empresa
entrada: datos de gastos, datos de cobros
salida: balance de la empresa
origen: operador consola
destino: sistema
necesita: base de datos de gastos, base de datos de cobros
precondición: ninguna
postcondición: mostrar balance
efectos laterales: ninguno

3.2.6 Gestión de empleados**Requisito 6.1**

función: contratar empleado
descripción: contrata un nuevo empleado
entrada: perfil, dni, nombre, direccion, cp, poblacion, provincia, telefono1, telefono2, movil, email, fax, nomina
salida: código del empleado
origen: operador consola
destino: sistema
necesita: base de datos de empleados

precondición: no exista el empleado en la base de datos
postcondición: empleado añadido a la base de datos
efectos laterales: ninguno

Requisito 6.2

función: despedir empleado
descripción: despide a un empleado contratado
entrada: código del empleado
salida: ninguna
origen: operador consola
destino: sistema
necesita: base de datos de empleados
precondición: exista el empleado en la base de datos
postcondición: empleado eliminado de la base de datos
efectos laterales: si el empleado es un abogado y estaba asignado a alguna multa, se elimina la asignación y se requiere una nueva asignación de abogado

Requisito 6.3

función: modificar empleado
descripción: modifica los datos de un empleado
entrada: código empleado
salida: ninguna
origen: operador consola
destino: sistema
necesita: base de datos de empleados
precondición: exista el empleado en la base de datos
postcondición: empleado modificado y almacenado en la base de datos
efectos laterales: ninguno

Requisito 6.4

función: consultar empleado por código
descripción: consulta los datos de un empleado
entrada: código empleado
salida: datos del empleado
origen: operador consola
destino: sistema
necesita: base de datos de empleados
precondición: exista el empleado en la base de datos

postcondición: mostrar datos del empleado
efectos laterales: ninguno

Requisito 6.5

función: consultar empleado por nombre
descripción: consulta los datos de un empleado
entrada: nombre del empleado
salida: datos del empleado
origen: operador consola
destino: sistema
necesita: base de datos de empleados
precondición: exista el empleado en la base de datos
postcondición: mostrar datos del empleado
efectos laterales: ninguno

Requisito 6.6

función: consultar empleado por NIF/CIF
descripción: consulta los datos de un empleado
entrada: NIF/CIF
salida: datos del empleado
origen: operador consola
destino: sistema
necesita: base de datos de empleados
precondición: exista el empleado en la base de datos
postcondición: mostrar datos del empleado
efectos laterales: ninguno

3.3 Requisitos de rendimiento

En la versión monousuario del producto, habrá un único terminal de comunicación con la aplicación, y un único usuario interactuando con ella. Además, la persistencia de los datos será consistente y fiable, al tiempo que se garantizará su seguridad y confidencialidad. La respuesta de la aplicación a las peticiones del usuario será inmediata.

En la versión web:

- El sistema deberá soportar la carga simultánea generada por un porcentaje de usuarios estimado en su momento por la empresa gestora, que dependerá del número de clientes y de los casos que lleve cada día.
- El sistema completo deberá correr sobre un servidor de características medias.

3.4 Requisitos de base de datos lógica

- En la primera versión del producto se diseñarán e implementarán las estructuras de datos que darán soporte al almacenamiento de los datos. Además, se utilizará un conjunto de ficheros XML para la persistencia de datos. Se definirá un DTD adecuado para representar la información manejada por el software: datos de clientes, contratos, multas, recursos, empleados, etc. Los ficheros tendrán que ser por tanto XML válidos según nuestro DTD
- La segunda versión del proyecto utilizará una base de datos basada en SQL cuyo diseño seguirá el modelo entidad-relación

3.5 Restricciones de diseño

En la versión web el sistema basará sus comunicaciones en protocolos estándar de Internet y los servidores deben ser capaces de atender consultas concurrentemente.

Los distintos subsistemas deberán tener un diseño e implementación sencillos, independientes de la plataforma o el lenguaje de programación.

3.6 Atributos del sistema software

3.6.1 Fiabilidad

El sistema debe ser muy fiable en lo referente a la gestión de datos de la empresa, de tal modo que se garantice su correcto almacenamiento y su persistencia. Debe proporcionar mecanismos de copia de seguridad para facilitar la recuperación en caso de fallo, y un sistema de verificación de datos para evitar trabajar con valores erróneos.

3.6.2 Disponibilidad

El sistema debe estar disponible en un régimen de 24 horas al día, todos los días del año. Con tiempos mínimos de inicio y recuperación.

3.6.3 Seguridad

El sistema debe tener mecanismos de protección adecuados para evitar el acceso a los datos a usuarios ajenos al entorno de la aplicación. También se deben utilizar mecanismos para verificar la integridad de los datos. Todos los accesos a los datos se archivarán en un histórico de acceso, para verificar la legitimidad. En la versión web del producto se utilizarán mecanismos de cifrado de la información transmitida para proteger la confidencialidad. Además, se establecerán diferentes niveles de acceso a los datos para los distintos usuarios que podrán acceder a la aplicación, así como se restringirán las funcionalidades que cada uno de ellos podrán realizar.

3.6.4 Mantenibilidad

El sistema estará dividido en módulos de forma que se facilite el mantenimiento del software. Se adjuntará también documentación sobre el proceso de desarrollo de la aplicación y del modo en que estará construida.

3.6.5 Portabilidad

El sistema será una aplicación Java estándar, por lo que podrá ser portada a todas las plataformas en las que esté disponible una JVM 1.4 o 1.5.