

[Nombre del proyecto]

[Nombre de la empresa]

Comentario [IW1]: Es la forma de identificar el proyecto en el que se está trabajando, este debería ser creativo.

Comentario [IW2]: Es la forma de identificar el grupo de trabajo .

ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

[Fecha]

Comentario [IW3]: Indica la fecha de entrega del documento al cliente.

[Versión del documento]

Comentario [IW4]: Versión del documento que se establece según la forma escogida de administración de configuración.

[Logos]

Comentario [IW5]: Imágenes que identifican el proyecto y la empresa.

[Autores]

Comentario [IW6]: Personas que realizaron el documento.



HISTORIAL DE CAMBIOS

En esta sección se presenta una tabla que describe la evolución y los cambios que se le realizan al documento desde que se inicia hasta que se haya llegado a la versión base que es entregada al cliente.

Versión	Fecha	Sección del documento modificada	Descripción de cambios (corta)	Responsable (S)
Indica la versión del documento, que depende según la forma de administración de configuraciones seleccionada.	Se incluye la fecha en la que fue realizado el cambio del documento.	Permite especificar las secciones del documento que fueron modificadas.	Es un pequeño resumen de los cambios más relevantes que fueron realizados en la versión	Indica las personas del grupo de trabajo que son responsables del o los cambios realizados en el documento.

Tabla 1: Historial de cambios

**Contenido**

HISTORIAL DE CAMBIOS	1
CONTENIDO.....	2
LISTA DE TABLAS	4
LISTA DE ILUSTRACIONES	5
1. INTRODUCCIÓN	6
1.1 PROPÓSITO	6
1.2 ALCANCE	6
1.3 DEFINICIONES, ACRÓNIMOS, Y ABREVIACIONES	7
1.4 REFERENCIAS	8
1.5 APRECIACIÓN GLOBAL.....	8
2. DESCRIPCIÓN GLOBAL	10
2.1 PERSPECTIVA DEL PRODUCTO	10
2.1.1 Interfaces con el sistema	10
2.1.2 Interfaces con el usuario.....	11
2.1.3 Interfaces con el Hardware.....	13
2.1.4 Interfaces con el Software.....	13
2.1.5 Interfaces de Comunicación.....	16
2.1.6 Restricciones de Memoria.....	16
2.1.7 Operaciones	16
2.1.8 Requerimientos de Adaptación del Sitio	17
2.2 FUNCIONES DEL PRODUCTO	17
2.3 CARACTERÍSTICAS DEL USUARIO	18
2.4 RESTRICCIONES.....	19
2.5 MODELO DEL DOMINIO	21
2.6 SUPOSICIONES Y DEPENDENCIAS	23
2.7 DISTRIBUCIÓN DE REQUERIMIENTOS	24
3. REQUERIMIENTOS ESPECÍFICOS	27
3.1 REQUERIMIENTOS DE INTERFACES EXTERNAS	31
3.1.1 Interfaces con el Usuario	31
3.1.2 Interfaces con el Hardware.....	32
3.1.3 Interfaces con el Software.....	33
3.1.4 Interfaces de Comunicaciones	34
3.2 CARACTERÍSTICAS DEL PRODUCTO DE SOFTWARE.....	34
3.3 REQUERIMIENTOS DE DESEMPEÑO.....	36
3.4 RESTRICCIONES DE DISEÑO	38
3.5 ATRIBUTOS DEL SISTEMA DE SOFTWARE (NO FUNCIONALES).....	39
3.5.1 Confiabilidad	39
3.5.2 Disponibilidad.....	39
3.5.3 Seguridad	40
3.5.4 Mantenibilidad.....	40
3.5.5 Portabilidad	41
3.6 REQUERIMIENTOS DE LA BASE DE DATOS	41
4. PROCESO INGENIERÍA DE REQUERIMIENTOS	43



5. PROCESO VERIFICACIÓN.....	44
6. ANEXOS.....	45



Lista de Tablas

Tabla 1: Historial de cambios	1
Tabla 2: Acrónimos.....	8
Tabla 3: Interfaces con el Software.....	15
Tabla 4: Descripción de las Características del Usuario	19
Tabla 5: Definiciones del Modelo de Dominio.....	22
Tabla 6: Formato de documentación del Modelo del Dominio.....	22
Tabla 7: Documentación de Requerimientos	30



Lista de Ilustraciones

Ilustración 1: Propósito	6
Ilustración 2: Alcance	7
Ilustración 3: Apreciación Global	9
Ilustración 4: Tipos de productos de software	10
Ilustración 5: Interfaces con el usuario	12
Ilustración 6: Operaciones.....	17
Ilustración 7: Tips para definir funciones del producto	18
Ilustración 8: Características del Usuario	19
Ilustración 9: Restricciones	20
Ilustración 10: Limitaciones.....	21
Ilustración 11: Descripción documentación del modelo del dominio	23
Ilustración 12: Suposiciones	24
Ilustración 13: Dependencias [1]	24
Ilustración 14: Distribución de requerimientos.....	26
Ilustración 15: Características de los Requerimientos	29
Ilustración 16: Documentación de Requerimientos	31
Ilustración 17: Interfaces con el Usuario	32
Ilustración 18: Interfaces de Hardware	33
Ilustración 19: Interfaces con el Software.....	34
Ilustración 20: División por Funcionalidades	35
Ilustración 21: Ejemplo Enunciado Requerimientos.....	37
Ilustración 22: Atributos de Calidad a tener en cuenta	38
Ilustración 23: Características de Confiabilidad	39
Ilustración 24: Características de Disponibilidad.....	40
Ilustración 25: Características de Seguridad	40
Ilustración 26: Características de Mantenibilidad.....	41
Ilustración 27: Portabilidad del Sistema	41
Ilustración 28: Características Bases de Datos.....	42



1. Introducción

Comentario [MT7]: Todos los elementos de ejemplo presentados en este documento están resumidos y están aquí para dar una idea de lo que se espera, pero debe Ud. Complementar e investigar.

1.1 Propósito

En esta sección se describe la intención con la que se realiza el documento, es decir, se deben exponer las razones por las que es importante, identificando el producto de software al cual se le van a especificar los requerimientos, la audiencia que está interesada en el contenido y el alcance del documento, es decir, especificar el alcance del producto que será descrito [1]. En la ilustración 1 se resume el contenido de esta sección.



Ilustración 1: Propósito

1.2 Alcance

Se describe el alcance del producto, es decir, la sección contiene una breve descripción del producto de software sobre el cual se realiza el SRS, indicando su nombre, las funcionalidades que incluirá y su utilidad (objetivos, beneficios). También puede ser incluida la relación entre el producto y las metas corporativas o estrategia de negocio resaltando la importancia que tiene para la organización [1].

Comentario [DP8]: Referencia al template de <http://www.processimpact.com/goodies.shtml>



Ilustración 2: Alcance

1.3 Definiciones, Acrónimos, y Abreviaciones

Se deben especificar los términos clave que serán utilizados en el documento con el fin de aclarar el contenido y asegurar su óptimo entendimiento. Se recomienda reutilizar los especificados en el documento SPMP y formar un glosario general de términos. En la tabla 2 se presentan algunos ejemplos de acrónimos para el documento SRS.

API	Aplication Programming Interface
CRUD	Create, Retrieve, Update, Delete
DBMS	Data Base Management System
GNU	Es un acrónimo recursivo que significa "GNU No es Unix"
GNU GPL	General Public License o licencia pública general
JDBC	Java DataBase Connectivity
JVM	Java Virtual Machine



LAN	Local Area Network
PHP	Es un acrónimo recursivo “Hypertext Preprocessor”
RFC	Request For Comments
SDD	Software Design Description
SQL	Structured Query Language
SRS	Software Requirement Specification
WLAN	Wireless Local Area Network
WPA	World Poker Association

Tabla 2: Acrónimos

1.4 Referencias

Indique aquí todas las referencias bibliográficas utilizadas en el documento. Utilice formato IEEE o APA para definirlas. Para administrar automáticamente las referencias, se recomienda el uso de la herramienta Zotero (www.zotero.org).

1.5 Apreciación Global

Esta sección debe tener una descripción general del contenido del documento, especificando su organización o distribución. Es una guía para el lector que le permitirá estructurar la información que encontrará en del documento.



Ilustración 3: Apreciación Global



2. Descripción Global

En general en esta sección se describe los factores generales que afectan al producto y sus requerimientos, es importante aclarar que en esta sección NO se especifican formalmente los requerimientos, es solo información de fondo que brinda a los lectores una descripción de todo el sistema. Esta en lenguaje de Usuario. Los elementos presentados en esta sección se asociaran en la sección 3 con Requerimientos Específicos.

2.1 Perspectiva del Producto

Esta sección describe el contexto y origen del producto que va a ser especificado en este SRS. La ilustración 1 muestra los diferentes tipos de productos de software que pueden ser desarrollados y, dependiendo de dicho tipo de producto se le debe dar un enfoque diferente a esta sección.

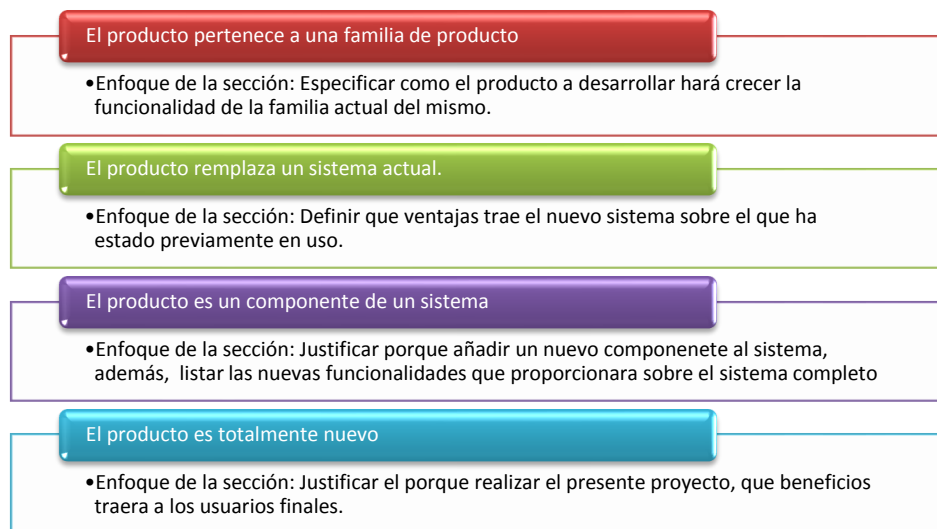


Ilustración 4: Tipos de productos de software

2.1.1 Interfaces con el sistema

En esta sección se deben definir y especificar todas las interfaces que usara el sistema para comunicarse con otros. Un problema común en esta sección es que se define el motor de base de datos como otro sistema y esto no es así, dado que dicho motor también hará parte de nuestro sistema. Para cada tipo de producto esta sección se especifica como sigue.



- ✓ **Producto que pertenece a una familia de productos:** Especificar como los demás productos de la familia pueden comunicarse con el producto que se está desarrollando.
- ✓ **Producto reemplaza a un sistema actual:** Por lo general, cuando un sistema es reemplazado, algunos componentes se deben reutilizar, así que los dos sistemas (nuevo y antiguo) deben definir su forma de comunicación. Esto no es una tarea sencilla ya que en ocasiones el sistema antiguo fue programado en otro lenguaje así que se deben implementar estrategias como CORBA (Common Object Request Broker Architecture) [<http://www.omg.org/gettingstarted/corbafaq.htm>], que es una arquitectura que permite que dos aplicaciones en diferentes lenguajes puedan interactuar.
- ✓ **Componente de un sistema:** En este tipo de proyectos donde un componente es un incremento más para llegar al sistema total, es totalmente necesario definir y especificar las interfaces con los demás componentes (módulos, paquetes, etc), así como también se debe definir mediante que interfaces los demás componentes se comunicaran con el componente a desarrollar.
- ✓ **Producto totalmente nuevo:** por lo general los productos nuevos no interactúan con otros sistemas, pero se debe especificar las interfaces del producto a desarrollar para el caso que nuevos componentes interactúen con el sistema. Vale aclarar, que por lo general cuando se realizan sistemas transaccionales nuevos, dichos productos si interactúan con otros sistemas como entidades financieras o administradoras estatales, por lo tanto, para interacción con entidades financieras se deben manejar estándares de seguridad como el ISO 17799[decreto 1465] que garantizan la seguridad necesaria en el proceso de remisión y recepción de la información.

2.1.2 Interfaces con el usuario

En esta sección se especifican las características lógicas de cada interfaz entre el producto de software y sus usuarios [3]. Por otra parte, en esta sección también se deben especificar los aspectos de optimización de interfaces, en resumen, esto significa que para hacer medibles los requerimientos de interfaz se debe definir métricas para calificar la usabilidad de una interfaz en particular. Por ejemplo, para hacer medible la interfaz gráfica de usuario asociada a la operación “Registrar Usuario” se puede medir



el tiempo necesario en entrenamiento para que el usuario pueda realizar dicha operación mediante la interfaz gráfica [4].

A continuación se presenta un ejemplo de las interfaces de usuario genéricas usadas en la mayoría de proyectos de software actuales:

Teclado:	<ul style="list-style-type: none">• Interfaz usada para el ingreso de datos en campos de texto. Ejemplo de uso de teclado: Registro de usuario
Ratón:	<ul style="list-style-type: none">• Interfaz usada para la navegabilidad entre interfaces gráficas, la selección de las diferentes opciones que da el sistema, el envío de formularios hacia la aplicación.
Pantalla:	<ul style="list-style-type: none">• El monitor permite al usuario, mediante una interfaz gráfica, observar las diferentes GUI's que conforman la aplicación. Se debe especificar la resolución que debe ser soportada por la pantalla, por ejemplo: "La pantalla debe soportar una resolución de 1024 * 768"
Interfaz GUI:	<ul style="list-style-type: none">• Para las interfaces GUI se debe especificar la forma de hacerlas es decir, librerías, programas y extensiones utilizadas para dicha tarea, además, al igual que en la interfaz "Pantalla" se debe definir la resolución que tendrán las diferentes GUI's de la aplicación. Ejemplo: "Las interfaces gráficas de usuario tendrán una resolución de 1024 * 768, y serán implementadas en Java Swing y Java Awt."
Tarjeta Gráfica:	<ul style="list-style-type: none">• En ocasiones las aplicaciones aconsejan que para una experiencia mejor entre el usuario y el software es recomendable el uso de una tarjeta gráfica integrada de 64 MB de RAM.
Tarjeta de Red:	<ul style="list-style-type: none">• Si la aplicación está destinada a funcionar bajo un entorno de red es necesario que cuente con una tarjeta de red Ethernet o inalámbrica.

Ilustración 5: Ejemplo Interfaces con el usuario

Cómo se ve en la ilustración 5, además de definir las interfaces que usará el sistema, también se deben dar especificaciones técnicas necesarias para que cada interfaz permita estar en la capacidad de permitir la interacción entre sistema y usuario. Estas especificaciones técnicas no solo hacen referencia a capacidades de cómputo sino también, como estas interfaces interactúan con el usuario en caso de algún tipo de incapacidad.



2.1.3 Interfaces con el Hardware

En esta sección se especifica como el software va a interactuar con componentes hardware específicos. En general, se definen y especifican puertos de comunicación, conjuntos de instrucciones, dispositivos soportados y protocolos utilizados [3] y [4]. A continuación se muestra un ejemplo de protocolos y puertos utilizados para una aplicación distribuida:

- ✓ **Protocolo de comunicación TCP/IP:** (Justificar porque el uso de este protocolo) Ejemplo: Se usará el protocolo de red TCP/IP ya que es un mecanismo de transporte confiable y además es orientado a conexión, evitando recepción de paquetes incompletos o dañados. No se usará protocolo UDP ya que este no es confiable en flujos de paquetes en tiempo real y el porcentaje de pérdida de datagramas es muy alto [5].
- ✓ **Puertos de red TCP:** (Para la comunicación de una aplicación distribuida se deben especificar los puertos de comunicación para configurar los firewalls en el lugar de trabajo) Ejemplo: La aplicación utilizara el puerto TCP 7777, mayor a 1024 para evitar conflictos con otras aplicaciones.
- ✓ **Cables y conexiones:** (Definir como los diferentes ordenadores de la aplicación realizarán su interacción) Ejemplo: para establecer la conexión entre los equipos se utilizara cable UTP conectados a un dispositivo de interconexión tales como hubs, switches y routers.

2.1.4 Interfaces con el Software

En esta sección se especifican como se comunicara el sistema a desarrollar con otros productos de software como por ejemplo sistemas gestores de bases de datos (DBMS), sistema operativo, maquinas virtuales, etc [3]. Para cada uno de estos productos de software se debe especificar nombre, número de versión y fuente; dado que con base a este documento se realizara la implementación del producto, es necesario especificar la documentación de las interfaces que se utilizarán. En esta sección también se debe justificar porque utilizar en el sistema a desarrollar otros productos de software [4]. A continuación se presenta una tabla ejemplo para un desarrollo con recursos gratuitos exceptuando el sistema operativo (Microsoft Windows XP Professional).



Producto de Software	Windows	phpMyAdmin	MySQL	JDBC	JVM
Descripción	La aplicación será desarrollada para la plataforma Windows que es una familia de sistemas operativos basados en una interfaz gráfica de usuario que aplican el paradigma de ventanas.[8]	Este DBMS es una herramienta escrita en PHP que permite manejar la base de datos seleccionada (MySQL). Además, es una herramienta robusta que brinda diferentes funcionalidades como lo son: CRUD de tablas, ejecución de sentencias en SQL, administrar privilegios, entre otras.[9]	Este sistema de gestión de bases de datos soporta la realización de transacciones en simultánea, multiusuario y multihilo; además, cuenta con la GNU GPL. [10]	Es un API que permite llevar a cabo operaciones sobre las bases de datos utilizando Java como lenguaje de programación; es independiente del sistema operativo y de la base de datos en la cual se utilice, emplea el lenguaje de query SQL.[11]	La máquina virtual de Java es un programa instalable en el sistema operativo que permite la interpretación y la ejecución de código de java.
Propósito de Uso	Asegurar que la mayoría de los usuarios finales puedan utilizar el producto sin inconvenientes, pues este sistema operativo es el más difundido y usado	Realizar la administración de la base de datos con facilidad, utilizando páginas Web, Internet, sentencias SQL, lo cual es proveído por este DBMS.	Utilizar una base de datos segura que soporte múltiples consultas, proporcione una capacidad de respuesta y almacenamiento eficiente y además,	Acceder a la base de datos por medio de un manejador de conexiones que implemente las interfaces en lenguaje de Java, debido a que la aplicación requiere realizar operaciones de	Puesto que el lenguaje de programación que se usará para el desarrollo de la aplicación, que es Java, no es un lenguaje de bajo nivel o de máquina es necesario usar un intérprete que permita la correcta



	en la actualidad. Además, el equipo de desarrollo tiene un mayor y fácil acceso a este.		que asegure la integridad de la información.	consulta y actualización constantemente a la base de datos	ejecución de estas instrucciones.
Versión	Versiones basadas en la tecnología NT	Versión 2.8.0.1	Versión 4.1.18	Versión 4.0	Versión 1.6
Fuente	<i>Microsoft Corporation</i> Ayuda: http://windowshelp.microsoft.com/Windows/es-ES/default.msp	<i>The pdhpMyAdmin project</i> Documentación: http://www.phpmyadmin.net/documentation/	<i>MySQL AB</i> Documentación: http://dev.mysql.com/doc/	<i>Sun Microsystems</i> Documentación: http://java.sun.com/j2se/1.4.2/docs/guide/jdbc/index.html	<i>Sun Microsystems</i> Documentación: http://java.sun.com/products/archive/jdk/1.1.8_009/jre/index.html
Comentarios Adicionales	La aplicación podría funcionar en otros sistemas operativos que soporten el JRE versión 1.6.	Se utilizará lenguaje SQL para realizar el CRUD de las tablas en la base de datos.	El equipo cuenta con más de un servidor que tiene este gestor de base de datos, lo que brinda facilidad de acceso y permite al equipo de desarrollo afrontar contratiempos de disponibilidad de la base de datos.	Puesto que JDBC es independiente del S.O, permite que la aplicación pueda funcionar en otras plataformas diferentes a Windows	Una gran ventaja de la JVM es la portabilidad que le brinda a Java, permitiendo la ejecución de programas desarrollados en este lenguaje en S.O que soporten la máquina virtual.

Tabla 3: Interfaces con el Software



2.1.5 Interfaces de Comunicación

En esta sección se especifica todas las interfaces de comunicación utilizadas tales como protocolos de red local etc [4]. Por lo general, cuando las aplicaciones hacen uso de diferentes computadores, se utiliza TCP/IP mediante una LAN o una WLAN, para esto se aconseja hacer referencia al RFC de TCP, pero cuando el producto de software utiliza comunicación por serial y los mensajes son únicos para el sistema es necesario especificar que significa cada bit de la trama.

2.1.6 Restricciones de Memoria

En resumen esta sección especifica características especiales de tarjeta de video, memoria volátil, dispositivos de almacenamiento secundario, etc [4]. Para esta sección se recomienda hacer una sumatoria de los requerimientos de memoria de cada uno de los productos de software usados para el desarrollo del sistema, el siguiente ejemplo está relacionado con los productos utilizados en la Tabla 3:

- **Máquina Virtual de Java:** Para la instalación del JRE v. 1.6 se requieren 80 MB de disco duro libre y 64 MB de RAM.
- **PhpMyAdmin:** Para la instalación del controlador de la base de datos, la cual debe encontrarse en la misma máquina de la base de datos, es necesario contar con al menos 128 MB de RAM, esto para permitir la utilización de todas las aplicaciones necesarias durante la ejecución de la aplicación.

Existen otros tipos de restricciones memoria para aplicaciones web como las del explorador web entre otras. Para las aplicaciones tipo stand-alone las restricciones de memoria están sujetas al diseño del sistema ya que es la única forma de conocer su uso de recursos

2.1.7 Operaciones

Esta sección debe especificar las operaciones especiales requeridas por el usuario tales como:

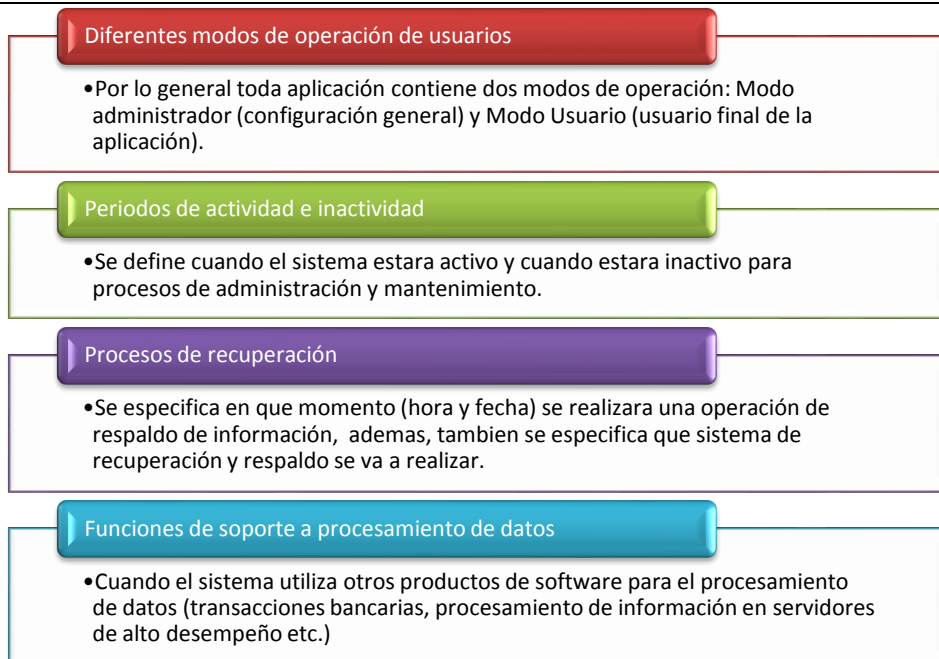


Ilustración 6: Operaciones

2.1.8 Requerimientos de Adaptación del Sitio

Debe definir los requerimientos para adaptar el software a un sitio dado o a un modo operacional. Además debe especificar las características que se deben modificar para adaptar el software a una instalación particular.

2.2 Funciones del Producto

Esta sección es una descripción general de las funciones que debe cumplir el producto de software [4]. El listado de funciones debe proveer la especificación clara de cada una, con el fin de asegurar que cualquier persona que lo lea, ya sea el cliente o desarrolladores puedan comprenderlo. Es aconsejable utilizar diagramas, gráficas, entre otros medios visuales que permitan comprender la relación lógica entre las funciones. Por ejemplo, un diagrama de casos de uso con su respectiva documentación. Un ejemplo claro son los Casos de Uso o User Stories que seguramente deberán ser referenciados.



Ilustración 7: Tips para definir funciones del producto

En la ilustración 7 se presentan las ideas claves que se deben tener en cuenta para definir y describir las funciones del producto.

2.3 Características del Usuario

Debe Identificar las diferentes clases de usuarios que usted prevea utilizaran la aplicación. Las diferentes clases de usuarios pueden ser diferenciadas basados en la frecuencia del uso, los diferentes roles, la experiencia técnica, los niveles de seguridad o de privilegios, el nivel de estudios, o de experiencia.

Además en general debe describir las características pertinentes de cada clase de usuario, las cuales se ilustran en la ilustración 8 y luego se describen en la Tabla 4 [1].

Usualmente este elemento también se tienen en cuenta en los documentos de Casos de uso, por tanto basta con referenciar al documento que lo contiene.

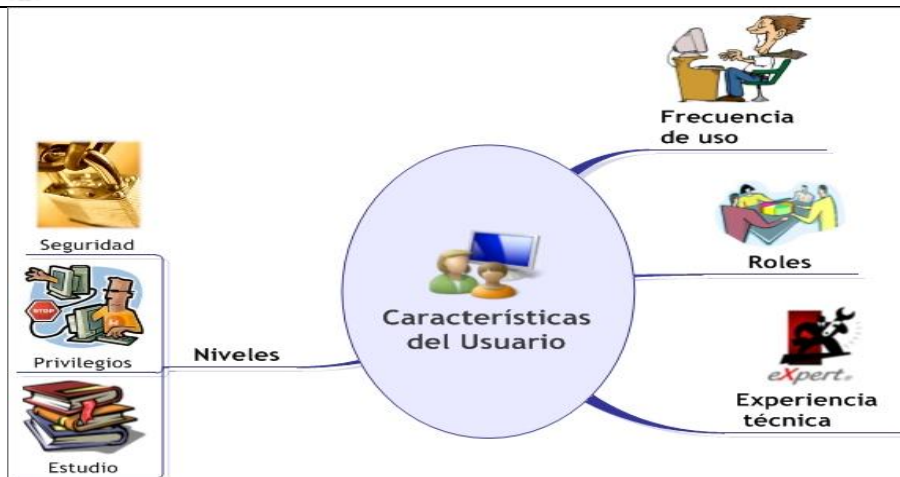


Ilustración 8: Características del Usuario

Características del Usuario	Descripción
Nivel de Seguridad o de Privilegios	Listar todos los privilegios o funcionalidades a las que tiene acceso el usuario.
Roles	Los diferentes roles que puede desempeñar el usuario, como por ejemplo: jugador, visitante, administrador, etc. Especificando las características específicas de cada uno de los roles.
Nivel de Estudios o Experiencia Técnica	El nivel de estudio o de experiencia técnica relacionada con el uso de la aplicación.
Frecuencia de Uso	Periodicidad con la que el usuario utiliza la aplicación.

Tabla 4: Descripción de las Características del Usuario

2.4 Restricciones

Debe especificar las características o restricciones del sistema entre las cuales se encuentran las descritas en la ilustración9:

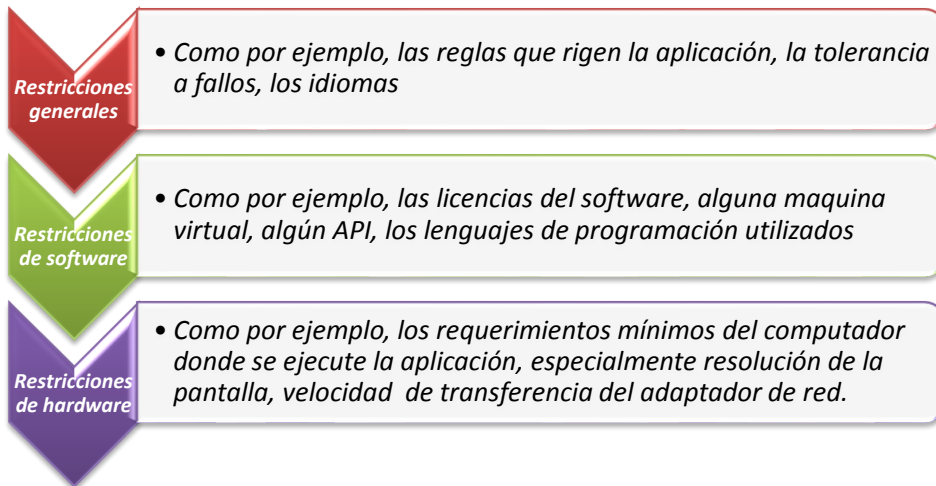


Ilustración 9: Restricciones

Además se debe proporcionar una descripción general de cualquier aspecto que limitara las opciones de los diseñadores. Algunas de estas se muestran en la ilustración 10 [1][3], todo lo expresado aquí debe estar justificado bien sea con referencias a documentnos legales o de contrato.



Ilustración 10: Limitaciones

2.5 Modelo del Dominio

Esta sección del documento debe reflejar el análisis inicial realizado al sistema a desarrollar, mediante la presentación del Diagrama del Modelo del Dominio y la documentación asociada.

Antes de especificar como documentar un diagrama del modelo del domino es importante dar las definiciones otorgadas por las organizaciones más importantes para la ingeniería de software, la siguiente tabla muestra dichas definiciones:

Organización / Autor	Definición
Construx	El modelo del dominio define un grupo de información que debe ser almacenado en el sistema, y las relaciones entre estos grupos de información. El modelo contiene al menos una lista de objetos del negocio, entidades de datos o clases (dichos objetos deben tener sus relaciones). El modelo del domino detallado debe tener también atributos para cada entidad del modelo sea una clase, una entidad de datos o un objeto del negocio.



<i>Flower 96</i>	<i>Es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés [10]. También se les denomina modelos conceptuales.</i>
<i>Larman</i>	<i>El modelo del dominio podría considerarse como un diccionario visual de las abstracciones relevantes, vocabulario del dominio e información del dominio.</i>

Tabla 5: Definiciones del Modelo de Dominio

Finalmente Larman termina con una conclusión muy importante para entender el porqué de un modelo del dominio. “LOS MODELOS DEL DOMINIO NO SON COMPONENTES DE SOFTWARE, el modelo del dominio es una representación de las cosas reales del mundo del dominio de interés” [11].

A continuación se presenta una forma para documentar los elementos en el diagrama del modelo del dominio.

ID		Elemento del Dominio	
Descripción			
Atributos			
Nombre	Descripción		Tipo de Dato
Objetivo			

Tabla 6: Formato de documentación del Modelo del Dominio



ID	•Identificador único del elemento del dominio del problema.
Elemento del Dominio	•Indica el nombre del elemento del dominio del problema que será documentado.
Descripción	•Contiene una breve descripción del elemento, se debe indicar el por qué del creación del mismo dentro del elemento del dominio.
Atributos	•De acuerdo a los atributos del elemento escogido se debe tomar cada uno para documentarlos.
Nombre	•Nombre del atributo que se documentará.
Descripción	•Descripción del objetivo del atributo dentro del elemento.
Tipo de Dato	•De acuerdo al lenguaje planeado para la implementación del sistema, establecer el tipo de dato que se le asignará al elemento.
Objetivo	•Descripción global acerca del elemento documentado con el fin de exponer su funcionalidad en el sistema.

Ilustración 11: Descripción documentación del modelo del dominio

2.6 Suposiciones y Dependencias

Lista de factores que afectan los requerimientos:

Suposiciones:

Se deben listar todas aquellas suposiciones que puedan llegar a afectar los requerimientos indicados en la sección 3. Estos pueden incluir componentes comerciales o de terceras personas que usted planea utilizar. Tenga en cuenta que el proyecto podría afectarse si estas suposiciones son incorrectas, no se comparten, o se cambian [1].



Ilustración 12: Suposiciones

Dependencias:



Ilustración 13: Dependencias [1]

2.7 Distribución de Requerimientos

Esta sección permite realizar un mapeo de los requerimientos con respecto a las funciones del producto y a los componentes, módulos o subsistemas que se definan inicialmente. Esta distribución es muy importante porque permite localizar los requerimientos, conociendo en que módulo o parte del sistema se puede encontrar ya



sea para validar, verificar o actualizar. Esto debe ser consistente con la estructura propuesta en la sección 3 de este documento.

En la ilustración 14 se propone un formato sencillo para el mapeo y la distribución de los requerimientos, es un ejemplo de la realización de un proyecto de Póker en donde se manejaron diferentes módulos y se utilizaron casos de uso para identificar los requerimientos.

Es importante tener en cuenta que cuando se realizan diagramas con imágenes, manejo de colores y demás herramientas gráficas se debe dejar explícito el significado de cada uno para que el lector se familiarice y comprenda con facilidad todo lo que se quiere expresar.

Por último, en esta sección se deben listar los requerimientos planeados para futuras versiones del sistema con una breve descripción de cada uno.

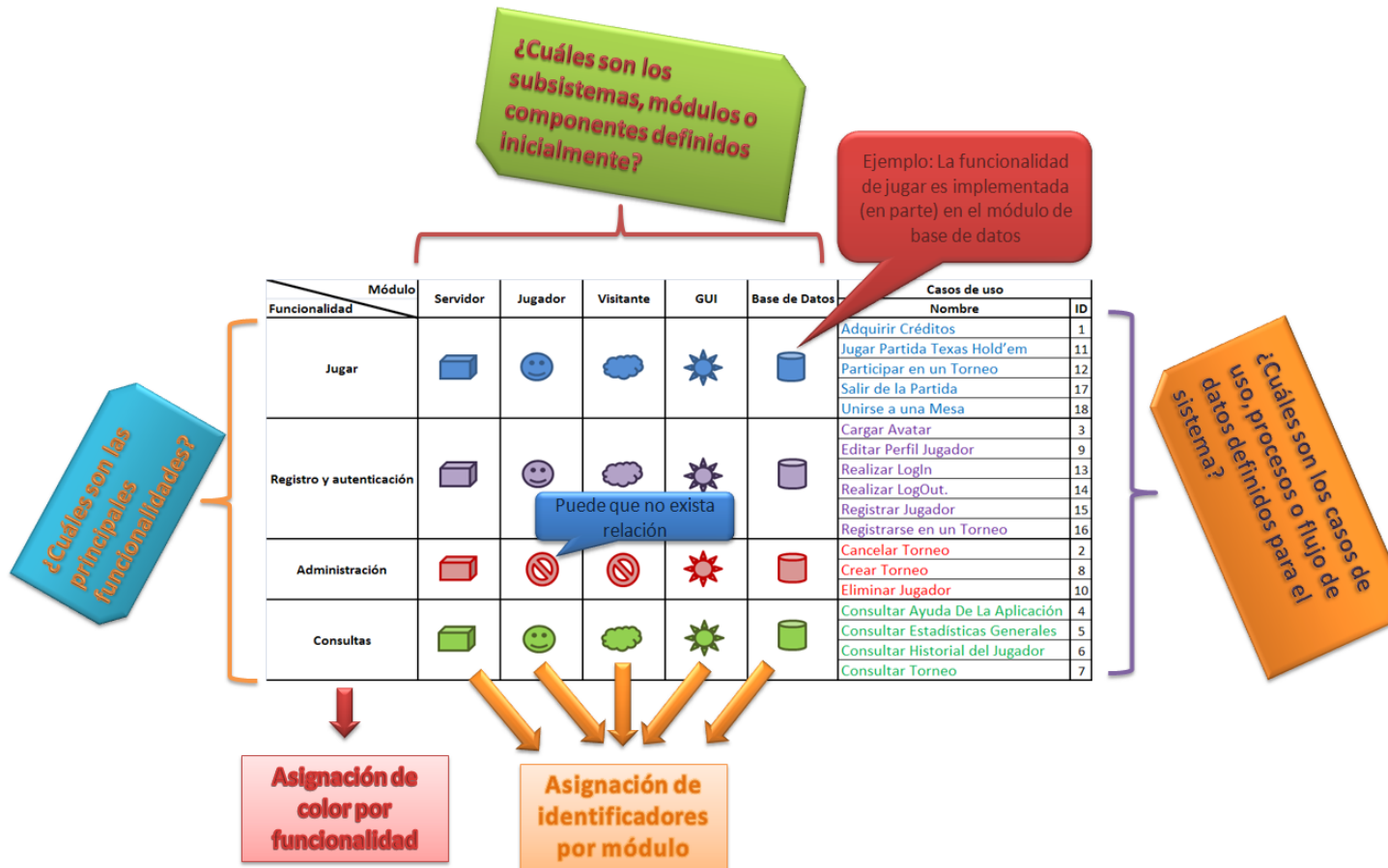


Ilustración 14: Distribución de requerimientos



3. Requerimientos Específicos

Esta sección es en cierto sentido la más importante del proyecto en cuanto a diseño e implementación se refiere. La idea principal de todo el documento, es la de realizar una conexión, un vínculo entre el cliente final y los desarrolladores de la aplicación o el sistema y esta sección es la encargada de esa comunicación.

La especificación de requerimientos busca trasladar las necesidades reales del usuario a un lenguaje más técnico que facilite su construcción, sin embargo es importante mantener un cierto nivel de descripción y lenguaje, para que el cliente no se pierda en tecnicismos y de esta forma hacerlo parte activa de todo el proceso (diseño y construcción).

Para hacer de este documento una parte valiosa en el proceso de construcción del sistema, todos los requerimientos que se ubiquen en esta sección deben cumplir con las siguientes características son [12]:

- **Atómico:** *TODOS los requerimientos debe ser atómicos, no compuestos, por lo tanto deben tener solo un propósito y su funcionalidad no debe poder ser dividida en más de una parte*
- **Correctos:** *los requerimientos que se incluyan en el documento deben ser solo los que tienen que ser incluidos en el sistema según el análisis previo y las entrevistas con los clientes y demás stakeholders. Requerimientos innecesarios o con enfoques a tecnologías o gustos de un solo usuario, hacen más extenso el diseño. atrasan la implementación y al final prueban ser inútiles para el sistema en general.*
- **No Ambiguos:** *los requerimientos finales deben tener solo una interpretación, no importa quién los lea, para facilitar el cumplimiento de esta característica es el diseño de términos que se refieran a una función en particular, por ejemplo palabras que se usen día a día en el negocio o en la empresa y que puedan entenderse con facilidad. Sin embargo el uso de estas palabras requieren la inclusión en el glosario de los documentos (SPMP, SRS, SDD).*
- **Completo:** *para que un requerimiento sea completo es preciso que incluya toda la información necesario para su perfecto entendimiento, esto es descripción, razón y criterio de medición, entre otros (para más información remitirse a la Tabla 7: Documentación de Requerimientos. La idea principal de que un requerimiento sea completo, es mantener al lector enfocado en un solo documento, y no que tenga que mirar en otro lado para poder entender lo que está escrito.*
- **Consistente:** *cada uno de los requerimientos debe estar pensado para que no contradiga otro requerimiento u otra sección de alguno de los documentos, de esta forma a la hora de implementar el sistema no se presentarán errores de*

Comentario [MT9]: Esta sección debe ser propuesta de acuerdo a la estructura presentada en la sección 2.7.

Por tanto la estructura propuesta por el Estándar no es la única.

Se espera en esta sección presentar requerimientos o relación a documentos que contengan los requerimientos.



redundancia o conflictos de intereses. Además es importante utilizar el mismo tipo de lenguaje entre requerimientos (ver No Ambiguos).

- **Importancia:** los requerimientos incluidos en el documento deben estar clasificados según el orden de importancia que ocupan dentro del sistema, de esta forma es más fácil para los programadores y desarrolladores enfocarse en las funcionalidades más críticas de la aplicación. Por ejemplo "Agregar Jugador" no es tan importante si "Crear un Juego" no funciona.
- **Verificable:** todos los requerimientos deben tener una forma con la cual se pueda saber si se están cumpliendo con los objetivos, o mejor si la funcionalidad de dicho requerimiento está presente en la implementación. Algunos de los métodos para verificar si un requerimiento es verificable son: Inspección, Análisis, Demostración y Prueba [13].
- **Modificable:** cualquier cambio a los requerimientos debe poder hacerse de manera sencilla, manteniendo la estabilidad del sistema y al mismo tiempo su estructura y estilo. La redundancia de requerimientos no permite su fácil ubicación y mucho menos su cambio.
- **Trazable:** un requerimiento es trazable si su origen es claro y además si su relación con otro tipo de requerimientos es fácilmente mapeable. Cuando un sistema entra en su fase de mantenimiento, la trazabilidad de los requerimientos es fundamental para llevar a cabo una fácil transición.
- **Asociados a Versión:** los requerimientos deben estar ordenados de acuerdo a la evolución y deben estar relacionados con el producto que satisfacen
- **No Redundante:** los requerimientos no deben estar en más de una parte del documento ni deben repetirse.
- **Precisos:** no dejan lugar a malas interpretaciones ni necesitan ser explicados, son concisos. Si es necesario (números) se debe hacer inclusión de cifras y posiciones decimales.

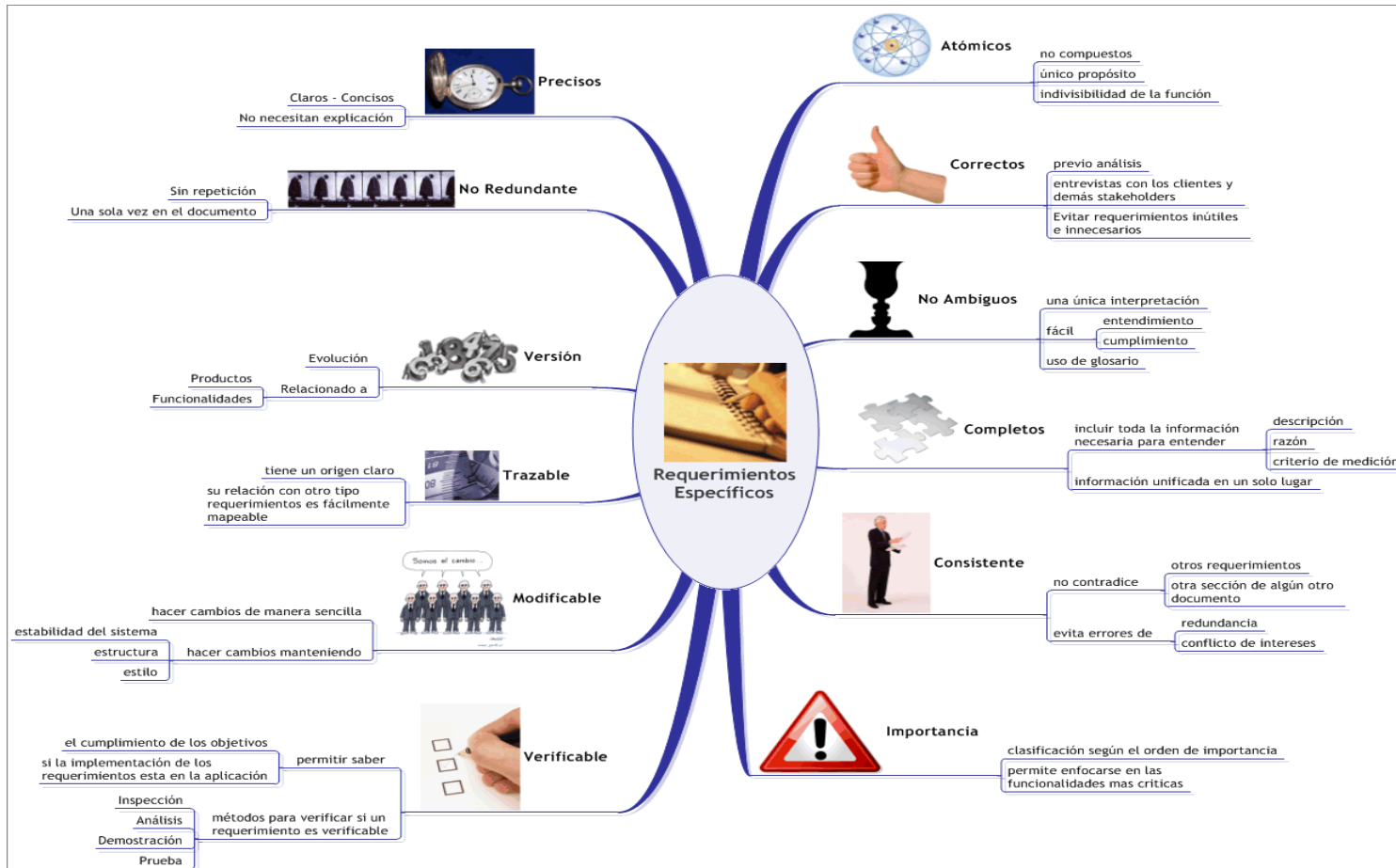


Ilustración 15: Características de los Requerimientos



La Tabla 7: Documentación de Requerimientos basada en la plantilla de Volere [14] presenta un posible formato que permite especificar los requerimientos funcionales de manera que se cumpla con las características anteriores.

# Requerimiento	Tipo de Requerimiento	Casos de Uso Asociados
Descripción		
Razón		
Autor		
Criterio de medición		
Prioridad	Módulo Asociado	
Versión	Fecha	

Tabla 7: Documentación de Requerimientos

# Requerimiento	<ul style="list-style-type: none">•Indica el número identificador del requerimiento.•Debe ser unico.
Tipo de Requerimiento	<ul style="list-style-type: none">•Este campo especifica la clase de requerimiento.•Pueden ser funcionales o no funcionales entre otras clases que se hallan definido en el proyecto.
Casos de Uso Asociados	<ul style="list-style-type: none">•Se especifican los identificadores de los Casos de Uso que se ven afectados por este requerimiento.
Descripción	<ul style="list-style-type: none">•Explicación del requerimien, exponiendo situaciones en las que debe manifestarse en el sistema.
Razón	<ul style="list-style-type: none">•Justificación de la existencia del requerimiento, ¿por qué se contempló?



Autor	<ul style="list-style-type: none">• Persona que realizó el levantamiento del requerimiento.• Usualmente el levantamiento lo realiza un integrante del grupo de trabajo.
Criterio de medición	<ul style="list-style-type: none">• Especifica la forma en que el requerimiento va a ser evaluado una vez haya sido implementado en el sistema.
Prioridad	<ul style="list-style-type: none">• Indica el grado de importancia de este requerimiento para el funcionamiento de la aplicación.
Módulo asociado	<ul style="list-style-type: none">• Indica la ubicación del requerimiento dentro de los módulos o subsistemas diseñados en el sistema (sección 2.7).
Versión	<ul style="list-style-type: none">• Identifica la versión del requerimiento para controlar el historial de cambios que se le ha realizado.
Fecha	<ul style="list-style-type: none">• Indica la fecha en la que fue realizada esta versión del requerimiento

Ilustración 16: Documentación de Requerimientos

3.1 Requerimientos de Interfaces Externas

Las siguientes secciones se encuentran estrechamente relacionadas con la sección 2.1, la explicación de estas, busca simplemente profundizar un poco más en el contenido que debe tenerse en cuenta en el momento de llenar esta plantilla y definir los requerimientos de Interfaces Externas.

3.1.1 Interfaces con el Usuario

En esta sección se debe explicar la forma en que el sistema o que la aplicación permitirá la comunicación con el usuario o el cliente final. Esta comunicación incluye el ingreso de datos al sistema, la navegación por ventanas, la forma en que se muestran las interfaces (si las hay) y los diferentes dispositivos del computador en el que se va a utilizar el sistema en construcción.

Las interfaces deben contener la información lógica, por ejemplo formatos de pantalla requeridos (1024*768, 800*600, 600*400), distribución de elementos en la pantalla o si tiene accesos rápidos con el teclado, por dar algunos ejemplos [12].

Algunas interfaces que se deben tener en cuenta son:

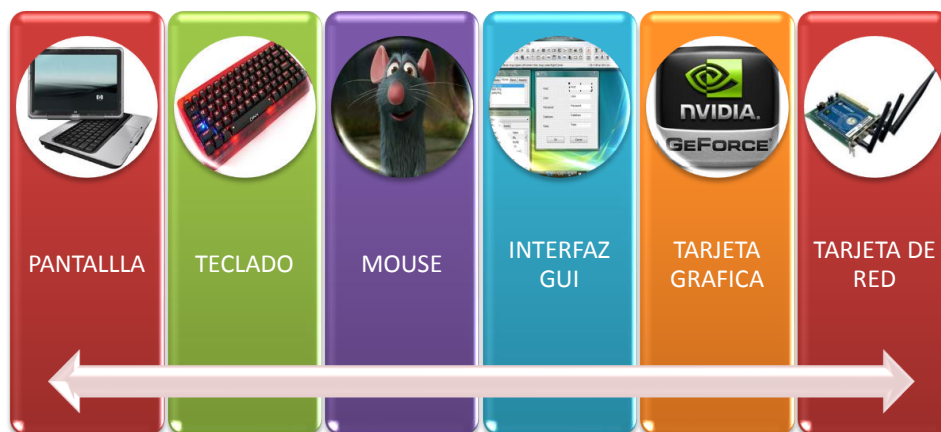


Ilustración 17: Interfaces con el Usuario

3.1.2 Interfaces con el Hardware

Puesto que el sistema que se planea desarrollar usualmente debe estar en capacidad de compartir datos e información con otros computadores y dispositivos (ejemplo dispositivos móviles), es necesario tener en cuenta la forma en que los componentes de software (aplicación o sistema) se comunicaran con los componentes de hardware de los otros dispositivos.

Las características del sistema a nivel de hardware que se deben tener en cuenta para el desarrollo, se enfocan en cómo se va a llevar a cabo la comunicación entre las máquinas de los usuarios participantes, algunas de estas interfaces son:

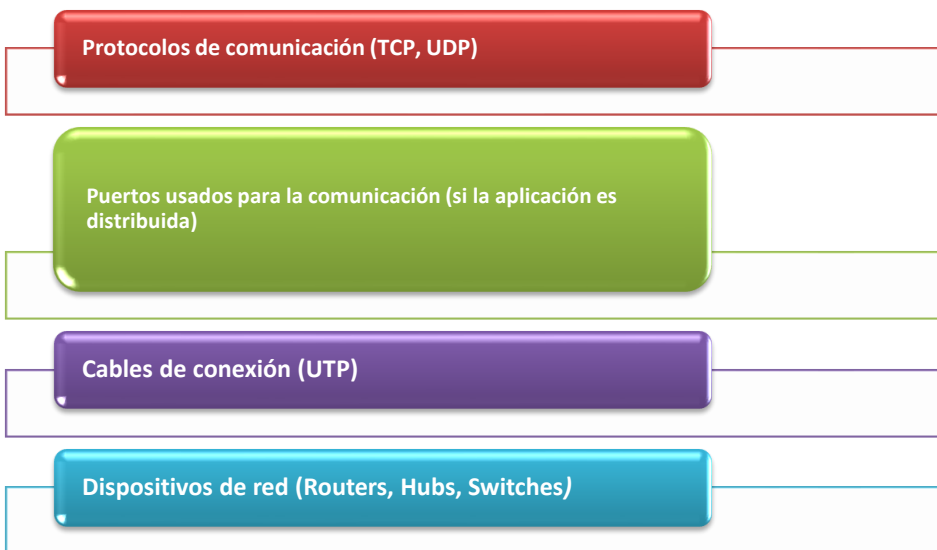
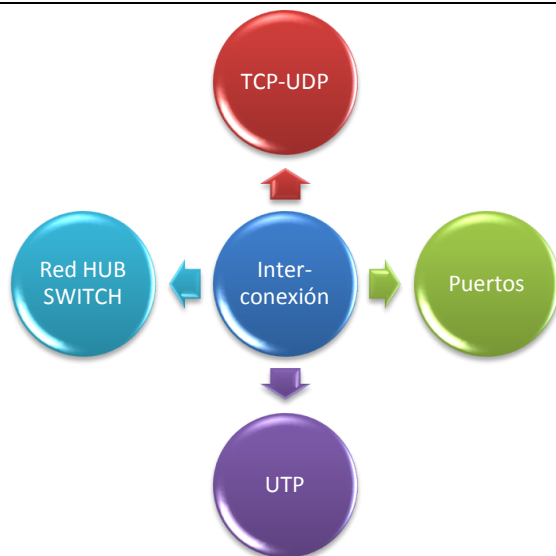


Ilustración 18: Interfaces de Hardware

3.1.3 Interfaces con el Software

Para la ejecución del sistema apropiadamente es necesario tener algunos requerimientos mínimos de Software, esto es versión del sistema operativo, servidores de bases de datos y en general todos los productos de software que permitan la



correcta utilización del sistema desarrollado. La Tabla 3: Interfaces con el Software muestra un formato que puede usarse para explicar estas interfaces.



Ilustración 19: Interfaces con el Software

3.1.4 Interfaces de Comunicaciones

Las interfaces de comunicación son básicamente los métodos como se interconectan las diferentes aplicaciones en las máquinas en donde se ejecutan (nuevamente si la aplicación es distribuida), estas interfaces incluyen el tipo de red que se debe montar para la conexión de computadores (LAN, WAN) y los protocolos de comunicación usados (TCP). Para apoyar la información escrita en esta sección, se aconseja explicar cierta parte en la sección 3.1.2

3.2 Características del Producto de Software

Los requerimientos funcionales definen las acciones fundamentales que deben encontrarse y ser cumplidas en el sistema, aceptando entradas, procesando y generando las salidas. Éstos generalmente se redactan de la siguiente forma: "El sistema debe...."

Para esta sección es aconsejable utilizar la Tabla 7: Documentación de Requerimientos con el fin de hacer de los requerimientos lo más entendibles y completos posibles. Además se sugiere realizar el acoplamiento con los casos de uso, permitiendo una



mayor trazabilidad y la capacidad de modificar requerimientos conociendo el impacto en todo el sistema.

Puesto que esta sección es la más importante del SRS, en ella se enumeran y se explican todas las características que debe poseer el sistema a implementar, y posiblemente la más extensa, es aconsejable tratar de dividirla por módulos, por funcionalidades o por cumplimiento de casos de uso.

A continuación se presenta una división basada en el proyecto 7 Texas Poker [15] en el cual se decidió fraccionar esta sección del por funcionalidades del sistema:

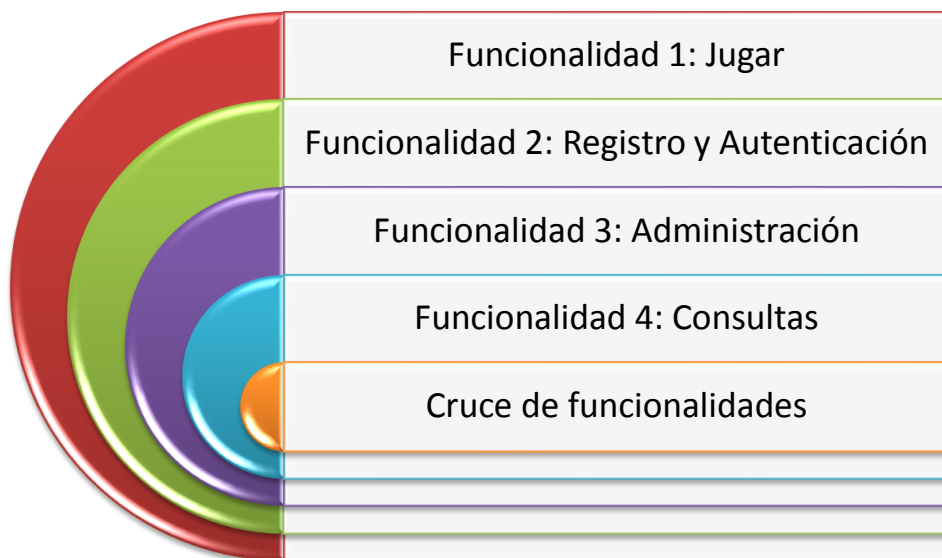


Ilustración 20: División por Funcionalidades

3.2.1 Funcionalidad 1: Jugar

Uso de la Tabla 6: Formato de documentación del Modelo del Dominio para la documentación de los requerimientos asociados a esta funcionalidad.

3.2.2 Funcionalidad 2: Registro y Autenticación

Uso de la Tabla 6: Formato de documentación del Modelo del Dominio para la documentación de los requerimientos asociados a esta funcionalidad.



3.2.3 Funcionalidad 3: Administración

Uso de la Tabla 6: Formato de documentación del Modelo del Dominio para la documentación de los requerimientos asociados a esta funcionalidad.

3.2.4 Funcionalidad 4: Consultas

Uso de la Tabla 6: Formato de documentación del Modelo del Dominio para la documentación de los requerimientos asociados a esta funcionalidad.

3.2.5 Cruce de funcionalidades

Uso de la Tabla 6: Formato de documentación del Modelo del Dominio para documentar los requerimientos que no pueden incluirse en alguna de las funcionalidades principales.

Tanto para esta sección como para la siguiente es muy importante tener en cuenta que todos los requerimientos deben tener su identificador y si se desea el caso de uso al que está asociado, esto con el fin de cumplir con la característica de trazabilidad y de esta forma en caso de modificación o de implementación facilitar la ubicación del requerimiento.

3.3 Requerimientos de Desempeño

Una vez se han establecido los requerimientos funcionales en la sección anterior, es hora de incluir los requerimientos de desempeño. En este punto se realiza la división en requerimientos estáticos que incluyen la cantidad de usuarios concurrentes o simultáneos que el sistema puede y debe soportar, contando las terminales o equipos utilizados y conectados al mismo tiempo, y en requerimientos dinámicos que pueden incluir el número de transacciones que debe soportar el sistema dependiendo de la funcionalidad y los tiempos de respuesta del sistema ante las peticiones hechas por los usuarios entre otros.

Es importante darle criterios de medición a estos requerimientos de desempeño, de forma que en las fases de verificación y validación se pueda comprobar que el desempeño del sistema, basado en los requerimientos de esta sección, cumpla con las necesidades del cliente o con las especificaciones finales acordadas en el SPMP. Un ejemplo de cómo escribir este tipo de requerimientos es:



La actualización de la interfaz gráfica debe hacerse por lo menos una vez cada minuto.



La actualización de la GUI debe realizarse frecuentemente con el fin de mantener al usuario informado de los cambios gráficos de la aplicación.

Ilustración 21: Ejemplo Enunciado Requerimientos

Otros atributos de calidad (Requerimientos no funcionales) con respecto al desempeño del sistema que deben tenerse en cuenta a la hora de documentar las restricciones de diseño son [16]:

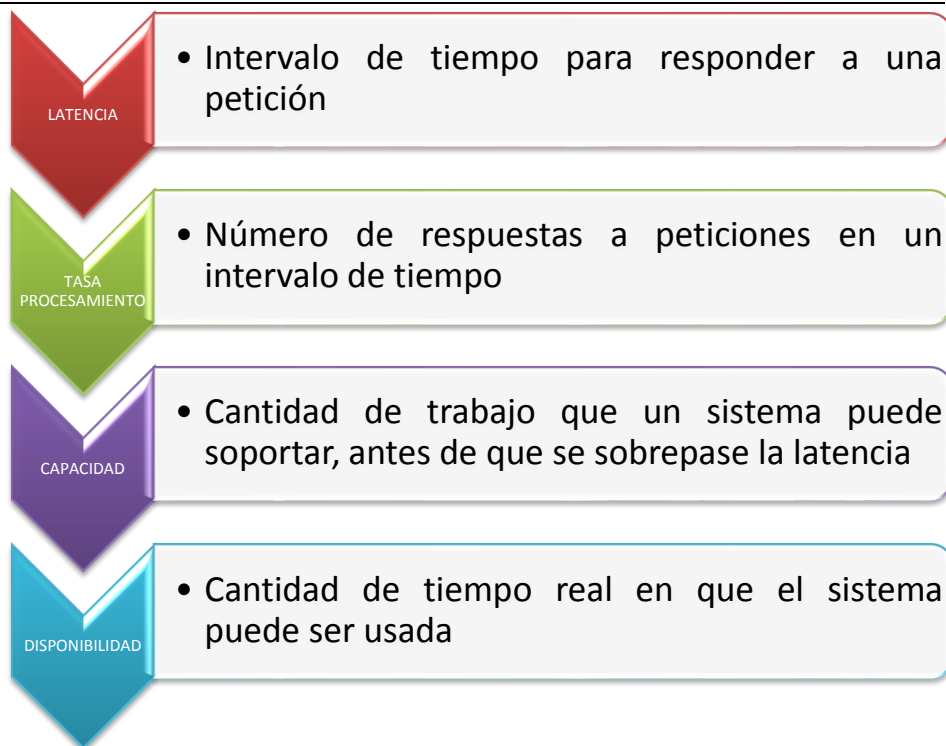


Ilustración 22: Atributos de Calidad a tener en cuenta

3.4 Restricciones De Diseño

En este numeral se tienen en cuenta las diferentes limitaciones del proyecto en cuestiones de diseño, los principales tópicos que deben tratarse:

- *Lenguajes de programación*
- *Herramientas CASE (Análisis y Diseño)*
- *Restricciones de Hardware y Software teniendo en cuenta el paradigma y el lenguaje de programación*
- *Diseño de la arquitectura final del sistema (Cliente-Servidor, Repositorio, N-Tier)*
- *Construcción del modelo de bases de datos*

Puesto que el documento de diseño que se usa para el proyecto es el SDD, es válido nombrar las restricciones sujetas al diseño e incluir referencia a ese documento para futuras o más completas explicaciones.



3.5 Atributos del Sistema de Software (No funcionales)

Para la sección de los requerimientos no funcionales también se puede usar el formato de la Tabla 7: Documentación de Requerimientos. Para la definición de los atributos, al igual que con los requerimientos de desempeño, es necesario incluir los criterios de evaluación medibles. Los principales atributos de calidad que deben ser incluidos en este documento son:

3.5.1 Confiabilidad

Se deben explicar los mecanismos que se van a tener en cuenta para el manejo de la información almacenada en el sistema. También incluye la forma en que la aplicación se va a mantener operativa a lo largo del tiempo especificado en la sección 3.5.2. Ejemplos de este tipo de requerimientos son:



Ilustración 23: Características de Confiabilidad

3.5.2 Disponibilidad

Porcentaje de tiempo al día o a la semana que el sistema debe funcionar sin necesidad de reiniciarlo. Si por ejemplo el sistema depende o necesita un módulo de comunicación con otras aplicaciones externas, el horario o mejor la disponibilidad de estas últimas también debe tenerse en cuenta para esta sección. Si es necesario que la aplicación o el sistema desarrollado cuenten con un administrador, el tiempo disponible que este posea para dedicarse a su rol debe ser incluido al mapear los requerimientos de disponibilidad.



Ilustración 24: Características de Disponibilidad

3.5.3 Seguridad

Este atributo depende en gran cantidad de la información que se maneje en el sistema, usualmente los métodos de seguridad más utilizados son los de permisos para usuarios o creación de cuentas, sin embargo si es necesario también podrían implementarse sistemas de encriptación o registros de acciones ejecutadas por los usuarios loggeados en el sistema. Si la aplicación es susceptible a ataques de Denegación de Servicio, mecanismos como replicación de servidores o balanceadores de cargas también pueden ser tenidos en cuenta.

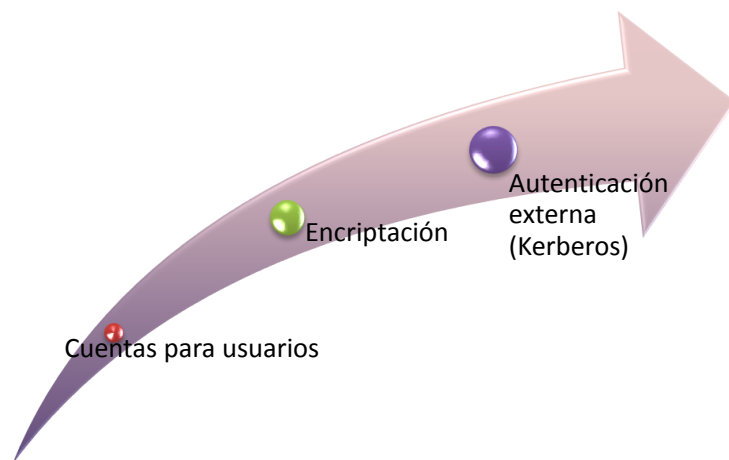


Ilustración 25: Características de Seguridad

3.5.4 Mantenibilidad

Una característica que hace del producto de software desarrollado fácil de mantener, es su división por funciones y por módulos, de esta forma si se necesita hacer una modificación a un requerimiento o en general a alguna función, no es necesario volver a implementar el sistema desde cero, ni afectar los módulos ya disponibles. Otra característica que se debe tener en cuenta es la de la documentación del código, en



caso que la modificación no sea hecha por los desarrolladores originales, con el fin de facilitar el entendimiento de la estructura interna del sistema.



Ilustración 26: Características de Mantenibilidad

3.5.5 Portabilidad

Se debe especificar si el sistema podrá ser migrado a otras plataformas de sistemas operativos o si podrá ser ejecutado en diferentes ambientes de cómputo estos ambientes pueden incluir hardware, software o una combinación de los dos. También se debe tener en cuenta el lenguaje y el compilador utilizados por el equipo desarrollador.



Ilustración 27: Portabilidad del Sistema

3.6 Requerimientos de la base de datos

Para la especificación de los requerimientos de la Base de Datos, es necesario tener en cuenta varios aspectos, entre estos se encuentran:

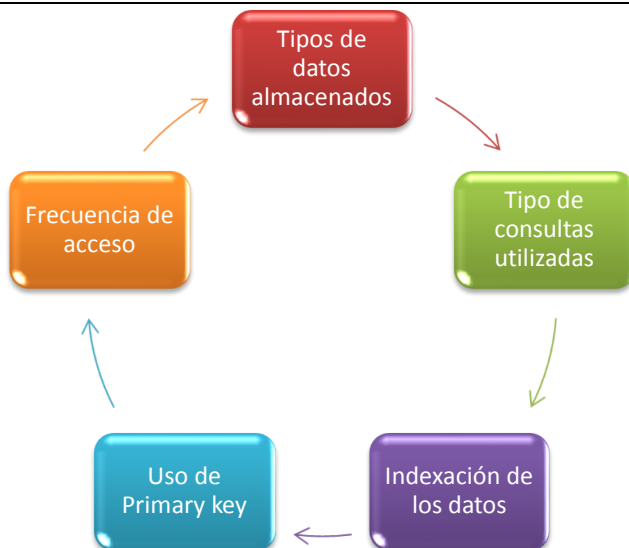


Ilustración 28: Características Bases de Datos

- **Tipos de datos almacenados:** dependiendo del motor de base de datos escogidos, es posible tener diferentes tipos de datos, sin embargo al hacer esta sección podrían incluirse los más usados: Char, Varchar, Numeric y Date por mencionar algunos.
- **Tipo de consultas utilizadas:** la forma en que los datos serán accedidos, consultados e ingresados desde los DAO's (Data Access Object) para evitar la introducción de sentencias malintencionadas. Los tipos más conocidos son Statement y Prepared Statment.
- **Indexación de los datos:** la eficiencia de las consultas complejas pueden reducirse dependiendo de la forma en que se haga el diseño de la base de datos, una buena forma de mostrar este aspecto es con el diagrama de Entidad Relación
- **Utilización de Primary Key:** al igual que con el aspecto anterior la utilización adecuada de una primary key puede evitar ciclos y además permite y facilita eficiencia en el tiempo de consultas hechas en tiempo real.
- **Frecuencia de acceso:** dependiendo del tipo de sistema que se desea implementar, especificar la frecuencia de acceso a la base de datos incluyendo el número de conexiones abiertas y tener en cuenta el tipo de consulta utilizada, la carga extra que puede producir el manejo de DAO's puede disminuir, aumentando de esta forma el desempeño en general de la aplicación.



4. Proceso Ingeniería de Requerimientos

En esta sección adicional, se espera que Ud. Presente el proceso, técnicas y métodos utilizados para la construcción de esta especificación.

Debe ser complementaria a la sección de Control de Requerimientos de su SPMP, la cual presenta precisamente como se gestionaran los cambios propuestos por un cliente o por Uds. Mismo a los requerimientos , basados en los mecanismos de gestión (localización, y trazabilidad) propuestos en el presente documento.



5. Proceso Verificación

En esta sección adicional, se espera que Ud. Presente el proceso de verificación y validación aplicado a esta especificación y el proceso aplicado para verificar y validar los requerimientos individuales.



6. Anexos





REFERENCIAS

- [1] Wiegers, Karl. , Software Requirements Specification. Process Goodies 2002, Disponible en <http://www.processimpact.com/goodies.shtml>
- [2] IronWorks, Plantilla SPMP, Segundo Semestre 2007, Pontificia Universidad Javeriana.
- [3] Construx Software, Software Requirements Specification CXOne Standard, Construx Software Builder, Inc, Noviembre 2002.
- [4] IEEE (Institute of Electrical and Electronics Engineers), IEEE Recommended Practice for Software Requirements Specifications, IEEE-SA Standards Board, Junio 1998.
- [5] Introduction to TCP/IP [homepage de Internet]. Copyright 1995 PCLT. Disponible en: <http://www.yale.edu/pclt/COMM/TCPIP.HTM>
- [6] Página principal de Windows [homepage de Internet]. ©2007 Microsoft Corporation. [citado 2007 Mar 25]. Disponible en: <http://www.microsoft.com/spain/windows/default.mspx>
- [7] phpMyAdmin | MySQL Database Administration Tool| [homepage de Internet]. © phpMyAdmin Devel Team. [citado 2007 Mar 25]. Disponible en: <http://www.phpmyadmin.net>
- [8] MySQL AB :: Developer Zone [homepage de Internet]. © 1995-2007 MySQL AB. [citado 2007 Mar 25]. Disponible en: <http://dev.mysql.com>
- [9] Java SE Technologies – Java Database Connectivity (JDBC) [homepage de Internet]. Copyright 1994-2007 Sun Microsystems, Inc. [citado 2007 Mar 25]. Disponible en: <http://java.sun.com/javase/technologies/database/index.jsp>
- [10] Fowler, M. 1996. Analysis patterns: Reusable Object Models, Reading, MA: Addison-Wesley
- [11] Larman C. UML Y PATRONES. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. 2nd ed. Aragón DF. Madrid: Pearson Educación. S.A.; 2003.



[12] IEEE (Institute of Electrical and Electronics Engineers), IEEE Guide for Developing System Requirements Specifications, IEEE-SA Standards Board, Abril 1996.

[13] Nuseibeh, B. et al, Requirements Engineering: A Roadmap, [citado 2007 Septiembre 07], Disponible en: <http://www.doc.ic.ac.uk/~ban/pubs/sotar.re.pdf>

[14] Pagina de Miguel Torres [homepage de Internet]. Bogotá. Ing. Miguel Eduardo Torres Moreno MSc. Copyright - Miguel Torres 2007. [actualizado el 26 Feb 2007; citado 2007 Septiembre 07]. Materias - Ingeniera de Software, Robertson, S. et. At. Mastering the Requirements Process

[15] IronWorks, Especificación de Requerimientos De Software 7 Texas Poker, Primer Semestre 2007, Pontificia Universidad Javeriana

[16] Barbacci, M. et al, Quality Attributes, Software Engineering Institute, Carnegie Mellon University, December 1995