**Documento de Requisitos del Sistema**

A continuación, encontrarán los elementos que deben incluir en el Documento de Requisitos del Sistema. Todos los párrafos del color de este texto son las instrucciones para completar cada sección. Estos deben ser removidos del documento a entregar.

Cada sección indica los elementos que debe contener. No altere el formato del documento (distribución de los títulos y secciones, tamaños, tipos de letra, espaciado, márgenes, etc.). Las figuras y tablas deben estar numeradas y etiquetadas con un tamaño de letra menor o igual a la del texto de este párrafo.

**Consistencia y uniformidad del documento**

Si bien el documento está divido en secciones que tratan diferentes elementos del proceso de gestión del proyecto, todos estos elementos de alguna manera se relacionan unos con otros y es por ello importante poder conectarlos (usando referencias cruzadas a secciones o artefactos generados en otros procesos). De igual manera es importante que el documento sea claro y consistente para que pueda ser útil en la gestión adecuada de su proyecto. En ese sentido, es importante ser claro y conciso en la redacción.

**Portada**

La primera página es la portada del documento. Incluye el nombre del proyecto, de la empresa, la fecha de entrega, la versión actual, un logo que identifique a la organización y el nombre de los autores del documento

**Tamaño**

1 página.

**Referencias**

[1], sección 5

# Historial de cambios

**Propósito**

Describir brevemente los cambios que ha sufrido el documento, con el fin de llevar una adecuada administración de configuración.

**Contenido**

Una tabla que indique los cambios que ha sufrido el documento. Debe incluir, al menos: fecha del cambio, descripción del cambio y persona o personas que realizaron el cambio. La estructura de esta tabla debe ser consistente con los lineamientos indicados en la sección .

**Referencias**

[1], sección 5.

# Prefacio

**Propósito**

Al igual que el exordio en un discurso oratorio, esta sección busca atraer la atención y preparar el ánimo de los oyentes.

Esta sección también debe proveer una descripción general del contenido del documento, especificando su organización o distribución de forma que el lector pueda estructurar la información que encontrará en del documento.

**Contenido**

Descripción breve del contenido, propósito y alcance del proyecto. Descripción de la audiencia esperada del documento (para quién es importante leer este documento).

**Tamaño sugerido**

1/2 página.

**Referencias**

[1], sección 5.

# Tabla de contenidos

**Propósito**

Encontrar rápidamente una sección específica del documento.

# Lista de figuras

**Propósito**

Encontrar rápidamente una figura específica del documento. Toda figura debe tener una fuente bibliográfica con su respectiva referencia o ser una elaboración original de los autores. También debe estar numerada y etiquetada y ser referenciada al menos una vez en el texto.

# Lista de tablas

**Propósito**

Encontrar rápidamente una tabla específica del documento. También debe estar numerada y etiquetada y ser referenciada al menos una vez en el texto.

# Introducción

**Propósito**

Describir brevemente el alcance del documento y dar una breve explicación o resumen del mismo.

**Contenido**

Ver subsecciones

**Referencias**

## Propósito del documento

**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 requisitos, la audiencia que está interesada en el contenido y el alcance del documento.

**Tamaño sugerido**

1/2 página.

**Referencias**

## Alcance del producto

**Propósito**

Describir el 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.

**Tamaño sugerido**

2 páginas.

**Referencias**

## Glosario

**Propósito**

Que el lector pueda entender el significado esencial de siglas, acrónimos y conceptos técnicos y del dominio del problema.

**Contenido**

Listado de siglas, acrónimos y conceptos, con su correspondiente explicación.

**Referencias**

[1], sección 5.3.

# Descripción global

**Propósito**

Esta sección describe los factores generales que afectan al producto y sus requisitos. Es importante aclarar que en esta sección no se especifican formalmente los requisitos, es solo información de fondo que brinda a los lectores una descripción de todo el sistema. Debe estar escrita en lenguaje de usuario. Los elementos presentados en esta sección se asociarán en la sección 8 con requisitos específicos.

**Contenido**

Ver cada subsección

## Perspectiva de producto

**Propósito**

Esta sección describe el contexto y origen del producto que va a ser especificado en este SRS.

**Contenido**

* Si el producto pertenece a una *familia de productos* especificar cómo el producto a desarrollar hará crecer la funcionalidad de la familia actual del mismo
* Si el producto reemplaza a un sistema actual definir qué ventajas trae el nuevo sistema sobre el que ha estado previamente en uso.
* Si el producto es un componente de un sistema se debe justificar por qué añadir un nuevo componente al sistema. Además, de deben listar las nuevas funcionalidades que proporcionará sobre el sistema completo.
* Si el producto es totalmente nuevo se debe justificar por qué realizar el producto, y describir qué beneficios traerá a los usuarios finales.

**Tamaño sugerido**

4 páginas.

**Referencias**

### Interfaces con el sistema

**Propósito**

En esta sección se deben definir y especificar todas las interfaces que usará 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 del sistema.

**Contenido**

* Si el producto que pertenece a una familia de productos, especificar cómo los demás productos de la familia pueden comunicarse con el producto que se está desarrollando.
* Si el producto remplaza a un sistema actual, por lo general, 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.
* Si el producto es un componente de un sistema, es totalmente necesario definir y especificar las interfaces con los demás componentes (módulos, paquetes, etc.), así como también se debe definir mediante qué interfaces los demás componentes se comunicarán con el componente a desarrollar.
* Si el producto es totalmente nuevo se deben especificar las interfaces del producto a desarrollar para el caso de que nuevos componentes interactúen con él. Vale aclarar, que por lo general cuando se realizan sistemas transaccionales nuevos, dichos productos 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 que garantizan la seguridad necesaria en el proceso de remisión y recepción de la información.

### Restricciones de memoria

**Propósito**

Esta sección específica características especiales de tarjeta de video, memoria volátil, dispositivos de almacenamiento secundario, etc.

**Contenido**

Hacer una sumatoria de los requisitos de memoria de cada uno de los productos de software requeridos por sistema.

### Operaciones

Esta sección debe especificar las operaciones especiales requeridas por el usuario.

Contenido

* Describir modos de operación del usuario. Por ejemplo, modo administrador (configuración general) y modo usuario (usuario final de la aplicación).
* Periodos de actividad e inactividad. Se define cuando el sistema estará activo o inactivo para procesos de administración y mantenimiento.
* Procesos de recuperación. Se especifica en que momento (hora y fecha) se realizará una operación de respaldo de información. Además, se especifica qué sistema de recuperación y respaldo se va a realizar.
* Funciones de soporte a procesamiento de datos. Indicar cuándo el sistema utiliza otros productos de software para el procesamiento de datos (transacciones bancarias, procesamiento de información en servidores de alto desempeño, etc.).

## Funciones del producto

**Propósito**

Esta sección es una descripción general de las funciones que debe cumplir el producto de software. 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 los 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.

Contenido

* Diagramas de casos de uso
* Descripción de los casos de uso o de las historias de usuario.

## Características del usuario

**Propósito**

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

**Contenido**

Mencionar cada usuario y describir sus características pertinentes. Tomar como referencia la Tabla 1.

|  |  |
| --- | --- |
| 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.  Indicar 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 . Descripción de las características del usuario

## Restricciones

**Propósito**

Debe especificar las características o restricciones del sistema.

**Contenido**

* Restricciones generales: las reglas que rigen la aplicación, la tolerancia a fallos, los idiomas, etc.
* Restricciones de software: las licencias del software, alguna máquina virtual, algún API, los lenguajes de programación utilizados, etc.
* Restricciones de hardware: los requisitos mínimos del computador donde se ejecute la aplicación, especialmente, la resolución de la pantalla, la velocidad de transferencia del adaptador de red, etc.

## Modelo de dominio

**Propósito**

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.

Según Larnman, un modelo del domino podría considerarse como un diccionario visual de las abstracciones relevantes, vocabulario del dominio e información del dominio.

**Contenido**

* Diagrama de clases que representa el modelo del dominio
* Plantilla para documentar los elementos en el diagrama del modelo de dominio. Tomar como referencia la Tabla 2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ID*** |  | | ***Elemento del dominio*** |  | |
| ***Descripción*** |  | | | | |
| ***Atributos*** | | | | | |
| ***Nombre*** | | ***Descripción*** | | | ***Tipo de dato*** |
|  | |  | | |  |
|  | |  | | |  |
| ***Objetivo*** |  | | | | |

Tabla . Plantilla para documentar el modelo de dominio

La información requerida por la Tabla 2 es la siguiente:

* 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 se debe crear este elemento dentro 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: información 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 atributo.
* Objetivo: descripción global acerca del elemento documentado con el fin de exponer su funcionalidad en el sistema.

## Suposiciones y dependencias

**Propósito**

Son aquellos factores que afectan los requisitos. 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, o si las dependencias no se cumplen.

**Contenido**

Listado de todas aquellas suposiciones y dependencias que puedan llegar a afectar los requisitos.

## Distribución de requisitos

**Propósito**

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

**Contenido**

En la Figura 1 se propone un formato sencillo para el mapeo y la distribución de los requisitos. La figura ilustra un proyecto de póker en donde se manejaron diferentes módulos y se utilizaron casos de uso para identificar los requisitos.

Es importante tener en cuenta que cuando se realizan diagramas con imágenes, manejo de colores y demás componentes gráficos, 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 requisitos planeados para futuras versiones del sistema con una breve descripción de cada uno.

|  |
| --- |
|  |

Figura . Distribución de requisitos

# Requisitos específicos

Esta sección es en cierto sentido la más importante del proyecto. 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 requisitos 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 requisitos que se ubiquen en esta sección deben cumplir con las siguientes características:

* Atómicos: todos los requisitos deben ser atómicos y no compuestos; por lo tanto, deben tener solo un propósito y su funcionalidad no debe poder ser divida en más de una parte.
* Correctos: los requisitos 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*. Requisitos 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 requisitos finales deben tener solo una interpretación, no importa quién los lea. Para facilitar el cumplimiento de esta característica se deben emplear términos que se refieran a una única función en particular. Sin embargo, el uso de términos requiere su inclusión en el glosario de los documentos SPMP, SRS y SDD.
* Completos: para que un requisito sea completo es preciso que incluya toda la información necesaria para su perfecto entendimiento; esto es, descripción, razón y criterio de medición, entre otros (para más información remitirse a la Tabla 7: Documentación de Requisitos. La idea principal de que un requisito 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.
* Consistentes: cada uno de los requisitos debe estar pensado para que no contradiga otro requisito u otra sección de alguno de los documentos. De esta forma, a la hora de implementar el sistema no se presentarán errores de redundancia o conflictos de intereses. Además, es importante utilizar el mismo tipo de lenguaje entre requisitos.
* Clasificados por importancia: los requisitos 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.
* Verificables: todos los requisitos deben tener una forma con la cual se pueda saber si se están cumpliendo con los objetivos, o mejor si la funcionalidad de dicho requisito está presente en la implementación.
* Modificables: cualquier cambio a los requisitos debe poder hacerse de manera sencilla, manteniendo la estabilidad del sistema y al mismo tiempo su estructura y estilo. La redundancia de requisitos no permite su fácil ubicación y mucho menos su cambio.
* Trazables: un requisito es trazable si su origen es claro y además si su relación con otro tipo de requisitos es fácilmente mapeable. Cuando un sistema entra en su fase de mantenimiento, la trazabilidad de los requisitos es fundamental para llevar a cabo una fácil transición.
* Asociados a una versión: los requisitos deben estar ordenados de acuerdo a la evolución y deben estar relacionados con el producto que satisfacen
* No redundante: los requisitos no deben estar en más de una parte del documento ni deben repetirse.
* Precisos: los requisitos precisos no dejan lugar a malas interpretaciones ni necesitan ser explicados.

La Tabla 3, basada en la plantilla de Volere, presenta un posible formato que permite especificar los requisitos de manera que se cumpla con las características anteriores.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***# Requisito*** |  | ***Tipo de requisito*** | |  | ***Casos de uso asociados*** | |  |
| ***Descripción*** |  | | | | | | |
| ***Razón*** |  | | | | | | |
| ***Autor*** |  | | | | | | |
| ***Criterio de medición*** |  | | | | | | |
| ***Prioridad*** |  | | ***Módulo asociado*** | | |  | |
| ***Versión*** |  | | ***Fecha*** | | |  | |

Tabla . Plantilla para la documentación de requisitos

La información requerida por la Tabla 3 es la siguiente:

* # Requisito: indica el número identificador del requisito el cual debe ser único.
* Tipo de requisito: este campo especifica la clase de requisito. Pueden ser funcionales o no funcionales, entre otras clases que se hayan definido en el proyecto.
* Casos de uso asociados: se especifican los identificadores de los casos de uso que se ven afectados por este requisito.
* Descripción: explicacióndel requisito, exponiendo situaciones en las que debe manifestarse en el sistema.
* Razón: justificación de la existencia del requisito. ¿Por qué se contempló?
* Autor: persona que realizó el levantamiento del requisito. Usualmente el levantamiento lo realiza un integrante del grupo de trabajo.
* Criterio de medición: especifica la forma en que el requisito va a ser evaluado una vez haya sido implementado en el sistema.
* Prioridad: indica el grado de importancia de este requisito para el funcionamiento de la aplicación.
* Módulo asociado: Indica la ubicación del requisito dentro de los módulos o subsistemas diseñados en el sistema.
* Versión: identifica la versión del requisito para controlar el historial de cambios que se le ha realizado.
* Fecha: indica la fecha en la que fue realizada esta versión del requisito.

## Requisitos de interfaces externas

**Propósito**

Las siguientes secciones se encuentran estrechamente relacionadas con la sección 7.1. 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 requisitos de interfaces externas.

**Contenido**

Ver subsecciones

### Interfaces con el usuario

**Propósito**

En esta sección se debe explicar la forma en que el sistema o que la aplicación permitirá la comunicación con el usuario. 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.

**Contenido**

Proveer información lógica de la interfaz, por ejemplo, formatos de pantalla requeridos (1024 x 768, 800 x 600 o 600 x 400), distribución de elementos en la pantalla o si tiene accesos rápidos con el teclado, etc.

Algunas posibles interfaces son: pantalla, teclado, mouse, GUI, tarjeta gráfica, tarjeta de red, etc.

### Interfaces con el hardware

**Propósito**

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

**Contenido**

Detallar las interfaces con el hardware. Por ejemplo, definir cómo se va a llevar a cabo la comunicación entre las máquinas de los usuarios participantes

Algunas de estas interfaces son: protocolos de comunicación (TCP, UDP), puertos, dispositivos de red, etc.

### Interfaces con el software

**Propósito**

En esta sección se especifica cómo se comunicará el sistema a desarrollar con otros productos de software como por ejemplo sistemas gestores de bases de datos (DBMS), sistema operativo, máquinas virtuales, etc. Para ellos es necesario tener algunos requisitos mínimos del software, esto es versión del sistema operativo, servidores de bases de datos y en general todos los productos de software para la correcta utilización del sistema desarrollado.

**Contenido**

La Tabla 4 muestra un formato que puede usarse para explicar estas interfaces.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | Producto de software | Producto 1 | Producto 2 | Producto n | | Descripción |  |  |  | | Propósito de uso |  |  |  | | Versión |  |  |  | | Fuente |  |  |  | | Comentarios adicionales |  |  |  | |

Tabla . Interfaces con el software

### Interfaces de comunicaciones

**Propósito**

En esta sección se especifican todas las interfaces de comunicación. 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, por ejemplo, comunicación por serial y los mensajes son únicos para el sistema, es necesario especificar que significa cada bit de la trama.

**Contenido**

Listado y explicación de las interfaces de comunicaciones

## Características del producto software

**Propósito**

Los requisitos 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 forma "*El sistema debe…*".

**Contenido**

Para esta sección es aconsejable utilizar la Tabla 3 con el fin de hacer los requisitos 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 requisitos 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.

Tanto para esta sección como para la siguiente es muy importante tener en cuenta que todos los requisitos 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 requisito.

## Requisitos de desempeño

**Propósito**

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

Es importante darle criterios de medición a estos requisitos de desempeño, de forma que en las fases de verificación y validación se pueda comprobar que el desempeño del sistema, basado en los requisitos de esta sección, cumpla con las necesidades del cliente o con las especificaciones finales acordadas en el SPMP

**Contenido**

Algunos atributos de calidad (requisitos no funcionales) con respecto al desempeño del sistema que deben tenerse en cuenta a la hora de documentar las restricciones de diseño son:

* Latencia: intervalo de tiempo para responder a una petición.
* Tasa de procesamiento: número de respuestas a peticiones en un intervalo de tiempo.
* Capacidad: cantidad de trabajo que un sistema puede soportar antes de que se sobrepase la latencia.
* Disponibilidad: cantidad de tiempo real en que el sistema puede ser usado.

## Restricciones de diseño

En este numeral se tienen en cuenta las diferentes limitaciones del proyecto en cuestiones de diseño.

Contenido

Mencionar restricciones de diseño para elementos como:

* Lenguajes de programación.
* Herramientas CASE
* Restricciones de hardware y software teniendo en cuenta el paradigma y el lenguaje de programación.
* Diseño de la arquitectura final del sistema.
* 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.

## Atributos del sistema software

**Propósito**

Describir los principales atributos de calidad, entre los que se pueden incluir:

* 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. Ejemplos de este tipo de requisitos son el manejo de transacciones concurrentes, la recuperación o tolerancia a fallos y la persistencia de los datos.
* Disponibilidad. Porcentaje de tiempo al día o a la semana en 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 requisitos de disponibilidad.
* Seguridad. Este atributo depende de la información que se maneje en el sistema. Usualmente los métodos de seguridad más utilizados son los de permisos para usuarios o la creación de cuentas. Sin embargo, si es necesario también podrían implementarse sistemas de encriptación o registros de acciones ejecutadas por los usuarios autenticados en el sistema. Si la aplicación es susceptible a ataques de denegación de servicio, mecanismos como la replicación de servidores o los balanceadores de carga también pueden ser tenidos en cuenta.
* 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 requisito 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.
* 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.

**Contenido**

Para esta sección se sugiere abordar los ítems anteriores y documentarlos usando la información solicitada en la Tabla 3.

## Requisitos de la base de datos

**Propósito**

Especificar los requisitos de la base de datos.

**Contenido**

Es necesario documentar la siguiente información:

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, consultadas e ingresados desde los *Data Access Object* (DAO) para evitar la introducción de sentencias malintencionadas.

Indexación de los datos. La eficiencia de las consultas complejas puede 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 la 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.

# Proceso de ingeniería de requisitos

**Propósito**

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

Esta sección debe ser complementaria a la sección de *Control de Requisitos* del SPMP, la cual presenta precisamente cómo se gestionarán los cambios propuestos a los requisitos, basados en los mecanismos de gestión (localización, y trazabilidad) propuestos en el presente documento.

**Contenido**

Ver detalle de las subsecciones.

## Planeación

**Propósito**

La planeación tiene como objetivo determinar cuáles son las actividades para realizar un proceso de ingeniería de requisitos, desde la levantamiento, especificación, validación y verificación de los requisitos.

**Contenido**

Es necesario documentar la siguiente información:

* Los encargados
* La frecuencia de planeación
* La puesta en marcha de la planeación, o la serie de actividades a seguir para el desarrollo del proceso de ingeniería de requisitos realizados por el equipo

## Levantamiento

**Propósito**

Determinar el procedimiento para levantar requisitos.

**Contenido**

Es necesario documentar la siguiente información:

* Los encargados
* La frecuencia de levantamiento de los requisitos
* Proceso de levantamiento de requisitos, describir los diferentes puntos de vista o clasificaciones seguidos para los requisitos levantados

## Especificación

**Propósito**

Realizar la especificación de los requisitos. Dejar explícita todas las características del requisito para que pueda ser utilizado y comprendido por todo el equipo de desarrollo, facilitar su ubicación y su impacto en el proyecto y artefactos desarrollados dentro de este, además de llevar un control y seguimiento de su avance.

**Contenido**

Es necesario documentar la siguiente información:

* Los encargados.
* La frecuencia de especificación de los requisitos.
* Plantilla de especificación de los requisitos. A continuación, se mencionan algunos de los campos que tiene la plantilla:

**ID:** este ID es el encargado de diferenciar cada requisito puesto que la codificación de los requisitos debe ser única. Cada ID tendrá la siguiente estructura:

**R (Categoría) (Tipo) (número de requisito del tipo)**, donde primero se pone una R de requisito, a continuación, se pone el número de categoría a la cual pertenece el requisito, después se pone la o las iniciales del tipo al que pertenezca (funcional, no funcional)

**Requisito:** se enuncia el requisito.

**Especificación y restricciones:** este campo tendrá una descripción detallada de los aspectos del requisito si la necesita para aclarar alguna parte o si posee detalles adicionales que en el campo anterior no se explican.

**Origen del requisito:** acá se dirá cuál es la fuente de la cual se sacó la información para desarrollar el requisito, ya sea que surja por los clientes, por restricciones del sistema, por decisión del equipo, etc.

**Tipo:** este campo tendrá el tipo al que corresponde el requisito, ya sea funcional o no funcional.

**Caso de uso asociado:** se pondrá en caso de que el requisito está asociado con algún o algunos casos de uso, será llenado con el nombre del caso de uso asociado.

**Prioridad:** se establece la prioridad que tiene el requisito, la cual se realizará por medio del esquema de priorización.

**Estado:** se selecciona el estado en el que se encuentre el requisito al momento de diligenciar. El estado del requisito determina el avance que se ha hecho para cumplir el requisito en el producto de software. Esto implica como se encuentra este requisito y así mismo facilitar el control de este, así como la toma de decisiones en caso que se haya excedido el tiempo para realizarse. Debido a que se tratan de requisitos específicos, ya paso por un primer proceso de validación. Se definen estados de avance del requisito:

* + Validado: El requisito ha pasado exitosamente por un proceso de validación para continuar con su especificación
  + Especificado: Si el requisito ha sido especificado, priorizado y se le ha establecido trazabilidad horizontal y verticalmente.
  + Verificado: Si el requisito ha sido representado en artefactos de diseño y de código.
  + Implementado: Una vez que el sistema haya pasado por pruebas y proceso de verificación para determinar el estado de cumplimiento del requisito.

**Aspecto de aceptación**

Se define el criterio con el cual se acepta el requisito descrito, se debe decir a grandes rasgos de qué manera será verificable el requisito.

**Versión**

Los requisitos comenzarán siempre con una versión 0.1 hasta que sean presentados a los clientes académicos. Cada vez que se presenten a los clientes académicos, se les sumará uno al primer número y cada vez que el requisito sufra cambios se le sumará uno al segundo.

* Utilizar un esquema de priorización para asignar la prioridad a los requisitos.
* La trazabilidad: busca que los requisitos se vean reflejados en las diferentes actividades, esto con el fin que se pueda hacer una posterior verificación del requisito una vez se encuentra en desarrollo o implementación.

# Proceso de verificación y validación

**Propósito**

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

**Contenido**

* Validación de requisitos. Realizar el proceso de validación de cada uno de los requisitos identificados a partir de la utilización de criterios de aceptación, algunos de ellos pueden ser:
  + Atómico
  + Correctos
  + No ambiguos
  + Completos
  + Consistente
  + Importancia
  + Verificable
  + Modificable
  + Trazable
  + Asociados a versión
  + No redundante
  + Precisos
* Verificación de requisitos. Realizar el proceso de verificación a cada uno de los requisitos. Identificar las características que debe cumplir cada requisito, así como actividades para realizar una verificación constante y consistente con la validación realizada anteriormente.
* Incluir diagramas de flujo de los procesos de validación y verificación.

# Anexos

Agregue aquí cualquier información adicional relevante para el proyecto para que sea referenciada desde las secciones previas.

Es importante tener en cuenta que hasta antes de esta sección, el documento debe ser autocontenido, es decir, el lector no debe necesitar leer ningún documento adicional o archivo externo para entender el proyecto. La información que se coloque aquí es simplemente un complemento, en caso de que el lector requiriera más detalles sobre algún tema.

# 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 (http://[www.zotero.org)](http://www.zotero.org)) o JabRef (<http://www.jabref.org>).