2010

06/04/2010

SRS: SYSTEM REQUERIMENTS SPECIFICATION



**LAURA ARIAS PRADA**

**NESTOR DIAZGRANADOS**

**ANDREA FAJARDO**

**WILLIAM JIMÉNEZ**

**GERMÁN MORALES**

**DAVID SUAREZ**

HISTORIAL DE CAMBIOS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versión | Fecha | Sección del documento modificada | Descripción de cambios (corta) | Responsable (S) |
| 0.1.0 | Marzo 12/2010 | Sección 1,2,3 | Alteración de las sub secciones a nuestro proyecto | Todos |
| 0.2.0 | Marzo 13/2010 | Sección 2.5, 2.7 | Avance de modelo de dominio y distribución de requerimientos | Andrea Fajardo, Arquitecto; Néstor Diazgranados, Diseñador Gráfico. |
| 0.3.0 | Marzo 14/2010 | Sección 1.6, 1.5 | Descripción del modelo en espiral  Presentación de la apreciación global del hito. | Laura Arias, Director de proyectos.  William Jiménez, Administrador de configuraciones y documentación. |
| 0.4.0 | MARZO 15/2010 | SECCIÓN 2.7, 3 | ESTRUCTURACIÓN SECCIÓN 3, CORRECCIONES SECCIÓN 2.7 | Andrea Fajardo, Arquitecto. |
| 0.4.1 | Marzo 15/2010 | Sección 3.1 | Reestructuración de acuerdo a los títulos | Laura Arias, Director de proyectos. |
| 0.5.0 | Marzo 16/2010 | Secciones 1.1 y 1.2 | Avance de las secciones mencionadas | David Suárez, Director de Calidad y Manejo de riesgos |
| 0.5.1 | Marzo 16/2010 | Tabla de contenido | Inclusión de los títulos, corrigiendo la numeración | Laura Arias, Director de proyectos. |
| 0.6.1 | Marzo 16/2010 | Secciones 3.1.5, 3.1.6, 3.2 | Explicación y desarrollo de las secciones mencionadas | Andrea Fajardo  Arquitecto. |
| 0.7.1 | Marzo 17/2010 | Secciones  3.1.1 – 3.1.4 | Explicación y desarrollo de las secciones mencionadas | Laura Arias, Director de proyectos. |
| 0.8.0 | Marzo 19/2010 | Todas las secciones | Estructuración del SRS al modelo espiral | Todos los integrantes de Alimnova® |
| 0.9.0 | Marzo 21/2010 | Sección 3.2.3 | Especificación completa de las sección 3.2.3 | William Jiménez  Administrador de configuraciones y documentación. |
| 1.0.0 | Marzo 20 /2010 | Sección 1.5 , 2.1.1.3, 2.1.1.4 | Redacción secciones 1.5, 2.1.1.3, 2.1.1.4 del documento | Andrea Fajardo |

Tabla : Historial De Cambios

CONTENIDO

[HISTORIAL DE CAMBIOS 2](#_Toc256726602)

[CONTENIDO 3](#_Toc256726603)

[LISTA DE TABLAS 5](#_Toc256726604)

[LISTA DE ILUSTRACIONES 6](#_Toc256726605)

[1. INTRODUCCIÓN 7](#_Toc256726606)

[1.1 PROPÓSITO 7](#_Toc256726607)

[1.2 ALCANCE 8](#_Toc256726608)

[Ilustración 1: Adaptación del juego Monopoly® a T-Monopoly 9](#_Toc256726609)

[Ilustración 2: Tablero T-Monopoly 9](#_Toc256726610)

[1.3 DEFINICIONES, ACRÓNIMOS, Y ABREVIACIONES 10](#_Toc256726611)

[1.4 REFERENCIAS 10](#_Toc256726612)

[1.5 APRECIACIÓN GLOBAL 10](#_Toc256726613)

[Ilustración 3: Apreciación Global 11](#_Toc256726614)

[INTRODUCCION 13](#_Toc256726615)

[1.6 MODELO DE ESPIRAL PARA EL PROCESO DE INGENIERÍA DE REQUERIMIENTOS 16](#_Toc256726616)

[1.6.1 Elicitación de Requerimientos: 16](#_Toc256726617)

[1.6.2 Análisis De Requerimientos 17](#_Toc256726618)

[1.6.3 Especificación De Requerimientos De Software 17](#_Toc256726619)

[1.6.4 Validación De Los Requerimientos 18](#_Toc256726620)

[2. PLAN DE REQUERIMIENTOS 18](#_Toc256726621)

[2.1 Levantamiento de requerimientos 18](#_Toc256726622)

[2.1.1 Fuentes de Requerimientos 18](#_Toc256726623)

[2.1.2. Técnicas de Recolección 18](#_Toc256726624)

[2.2 Análisis de Requerimientos 19](#_Toc256726625)

[2.2.1. Riesgos Generales De Los Requerimientos 19](#_Toc256726626)

[2.2.2. Clasificación De Los Requerimientos 19](#_Toc256726627)

[2.2.3 Modelo Conceptual 24](#_Toc256726628)

[2.2.4 Diseño Arquitectónico y Asignación de Requerimientos 24](#_Toc256726629)

[2.2.5 Modelo Del Dominio 25](#_Toc256726630)

[3. Especificación De Requerimientos 25](#_Toc256726631)

[3.1. Documento de Especificación de Requerimientos 25](#_Toc256726632)

[3.2. Especificación de las Categorías Funcionales 25](#_Toc256726633)

[Requerimientos Funcionales 25](#_Toc256726634)

[3.2.1. Casillas Especiales 25](#_Toc256726635)

[3.2.2 Jugadas Especiales 27](#_Toc256726636)

[3.2.3 Transacciones 28](#_Toc256726637)

[3.2.4 Partida 28](#_Toc256726638)

[3.3 Especificación de las Categorías No Funcionales 28](#_Toc256726639)

[Requerimientos No Funcionales 28](#_Toc256726640)

[3.3.1 Restricciones 28](#_Toc256726641)

[3.3.2 Especificaciones 28](#_Toc256726642)

[Ilustración 20: División por Funcionalidades 29](#_Toc256726643)

[Funcionalidad 1: Jugar 29](#_Toc256726644)

[Funcionalidad 2: Registro y Autenticación 30](#_Toc256726645)

[Ilustración 23: Características de Confiabilidad 31](#_Toc256726646)

[Ilustración 24: Características de Disponibilidad 32](#_Toc256726647)

[Ilustración 25: Características de Seguridad 32](#_Toc256726648)

[Ilustración 26: Características de Mantenibilidad 33](#_Toc256726649)

[Ilustración 27: Portabilidad del Sistema 33](#_Toc256726650)

[Ilustración 28: Características Bases de Datos 34](#_Toc256726651)

[4. Validación de requerimientos 34](#_Toc256726652)

[4.1. Revisión de conceptos de documentación 34](#_Toc256726653)

[4.2. Análisis de Trazabilidad 34](#_Toc256726654)

[4.3. Evaluación de Requerimientos de Software 34](#_Toc256726655)

[ANEXOS 36](#_Toc256726656)

[REFERENCIAS 37](#_Toc256726657)

LISTA DE TABLAS

[Tabla 1: Historial de cambios 1](#_Toc176967938)

[Tabla 2: Acrónimos 7](#_Toc176967939)

[Tabla 3: Interfaces con el Software 13](#_Toc176967940)

[Tabla 4: Descripción de las Características del Usuario 17](#_Toc176967941)

[Tabla 5: Definiciones del Modelo de Dominio 19](#_Toc176967942)

[Tabla 6: Formato de documentación del Modelo del Dominio 20](#_Toc176967943)

[Tabla 7: Documentación de Requerimientos 27](#_Toc176967944)

LISTA DE ILUSTRACIONES

Ilustración 1: Propósito 5

Ilustración 2: Alcance 6

Ilustración 3: Apreciación Global 7

Ilustración 4: Tipos de productos de software 8

Ilustración 5: Interfaces con el usuario 10

Ilustración 6: Operaciones 15

Ilustración 7: Tips para definir funciones del producto 16

Ilustración 8: Características del Usuario 17

Ilustración 9: Restricciones 18

Ilustración 10: Limitaciones 18

Ilustración 11: Descripción documentación del modelo del dominio 20

Ilustración 12: Suposiciones 21

Ilustración 13: Dependencias [1] 21

Ilustración 14: Distribución de requerimientos 23

Ilustración 15: Características de los Requerimientos 26

Ilustración 16: Documentación de Requerimientos 28

Ilustración 17: Interfaces con el Usurario 29

Ilustración 18: Interfaces de Hardware 30

Ilustración 19: Interfaces con el Software 31

Ilustración 20: División por Funcionalidades 32

Ilustración 21: Ejemplo Enunciado Requerimientos 34

Ilustración 22: Atributos de Calidad a tener en cuenta 35

Ilustración 23: Características de Confiabilidad 36

Ilustración 24: Características de Disponibilidad 37

Ilustración 25: Características de Seguridad 37

Ilustración 26: Características de Mantenibilidad 38

Ilustración 27: Portabilidad del Sistema 38

Ilustración 28: Características Bases de Datos 39

# 1. INTRODUCCIÓN

## 1.1 PROPÓSITO

El documento de requerimientos de software (SRS por sus siglas en ingles) pretende describir de una forma muy especifica el producto a realizar (T-Monopoly) por Alimnova®. El fin de esto es que todos los Stakeholders se empapen del producto a y de esta forma poder desarrollar de una mejor manera la aplicación. Adicionalmente el SRS será un documento que exprese al cliente lo que Alimnova®

El SRS pretende describir el producto completamente, es decir, poder recoger todas las funcionalidades acordadas con el cliente, y describirlas de forma que sean claras para todos los Stakeholders involucrados en el proyecto y de esta forma hacer que el proceso de desarrollo sea más sencillo y consistente con los requerimientos definidos.

## 1.2 ALCANCE

El producto a realizar será T-Monopoly, este es un producto basado en Monopoly, ya que la base del juego es la misma pero cambiaran algunas de la reglas como se muestra a continuación:

|  |  |  |
| --- | --- | --- |
| MODIFICACIÓN | TIPO | DESCRIPCIÓN |
| Tablero | Interfaz | Cambio de temática de las propiedades. **[Ver figura siguiente]** |
| Cartas de arca comunal y casualidad | Temática | Dado que la temática es la zona T las gratificaciones o penitencias serán de acuerdo a dicha temática. |
| Parada libre | Regla | En parada libre según las reglas, es un sitio de descanso la idea es que haya un hippie al que deban comprarle una manilla. |
| Repartición de propiedades | Funcionalidad | Al inicio del juego se repartirán 2 tarjetas de propiedad a cada jugador. |
| Fichas | Interfaz | Las fichas serán diferentes tipos de automóviles, desde una zorra hasta el auto más costoso. |
| Tres pares seguidos | Funcionalidad | En el juego original al sacar 3 pares seguidos, el jugador va a la cárcel, se modificara dando la oportunidad de comprar una propiedad disponible a mitad de precio. |

Tabla : Modificaciones de las reglas del Monopoly® aplicado a T-Monopoly

A continuación se hará una breve descripción del tablero del T-Monopoly, por medio de un mapa mental donde se puede ver la adaptación que se hizo del juego Monopoly® a la aplicación que Alimnova® quiere desarrollar.



Ilustración : Adaptación del juego Monopoly® a T-Monopoly



Ilustración : Tablero T-Monopoly

Las dos ilustraciones anteriores pretenden describir a grandes rasgos como va ha ser la aplicación a desarrollar por Alimnova®.

## 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. Para mayor claridad revisar la sección 3 de la plantilla del SPMP [2]. 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](http://es.wikipedia.org/wiki/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 : Acrónimos

## 1.4 REFERENCIAS

Remítase a la sección 2 de la plantilla del SPMP [2] donde se encuentra explicado de manera detallada el manejo de referencias.

## 1.5 APRECIACIÓN GLOBAL

Para realizar la documentación de SRS (System Requirements Specification) Alimnova® decidió manejar el proceso de obtención de requerimientos y desarrollo de estos requerimientos de la misma forma que el ciclo de vida que tenemos para el desarrollo del proyecto, el cual es el modelo en espiral, es por esto que decidimos estructurar este documento de forma tal que permita que el desarrollo de este muestre claramente el proceso llevado a cabo por el modelo de espiral para el proceso de ingeniería de requerimientos [[Sección 1.6](#_1.6_MODELO_DE)] de forma que sea claro para Alimnova® el desarrollo de este proceso relacionado con los requerimientos y para nuestro cliente Miguel Torres la lectura del mismo, facilitando así el seguimiento de los requerimientos, ya que de este forma podemos identificar el origen de cada requerimiento y su posterior ubicación en el desarrollo del juego y del proyecto como tal.

Es así como decidimos estructurar el documento en 3 grandes temas, en los que está: el plan de requerimientos, especificación de requerimientos y validación de requerimientos.

A continuación se muestra la estructura correspondiente a la sección 2, denominada Plan de Requerimientos:

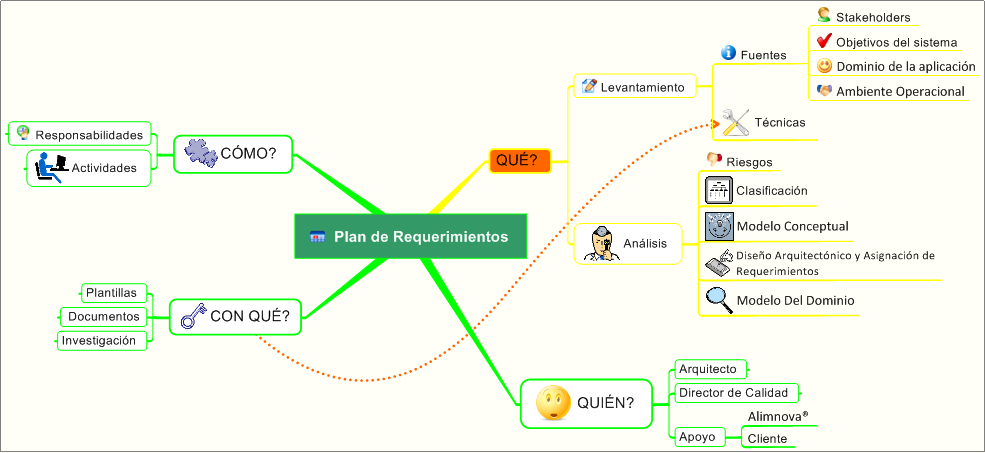


Ilustración : Apreciación Global PLAN REQUERIMIENTOS

Respecto a la estructura de la sección 3, denominada especificación de requerimientos. Es importante resaltar que para realizar esta especificación Alimnova® ha decido por facilidad de documentación y claridad del documento realizar una división más diciente y completa que la división de los requerimientos sólo por funcionales y no funcionales, ya que esta división por sí sola, no es completa ni muy útil tanto para el entendimiento, como para el desarrollo posterior basado en este documento, es por esto que dentro de la división más general de los requerimientos funcionales y no funcionales, Alimnova® agrupó dentro de los requerimientos no funcionales las especificaciones y restricciones dadas por el cliente y por las reglas de la aplicación, las cuales ya inicialmente han sido dadas y son conocidas por cada uno de los integrantes de Alimnova®.

Respecto a los requerimientos funcionales Alimnova® decidió relacionarlos directamente con una funcionalidad del juego y el comportamiento ó respuesta que debe dar el sistema respecto a determinada operación necesaria para el correcto desarrollo de juego, con el fin de tener claridad de la razón de ser de cada requerimiento y su directa aplicación con el desarrollo del proyecto, permitiendo así claridad y fácil ubicación de cada uno de los requerimientos identificados.

La clasificación descrita corresponde a la siguiente gráfica:

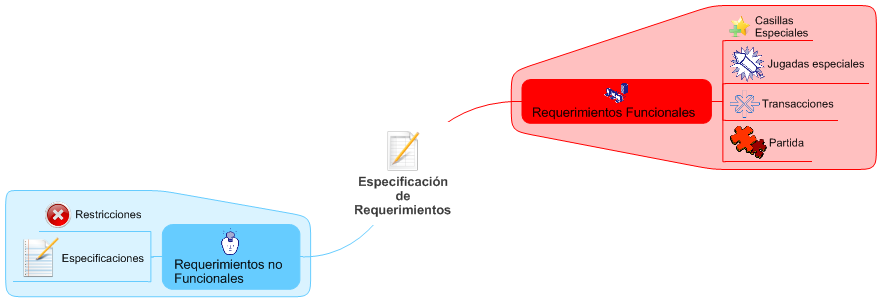


Ilustración 4: Apreciación Global especificación de requerimientos

En cuanto a la sección de validación de requerimientos, Alimnova® considera que es muy importante mantener una continua validación y retroalimentación de los requerimientos, ya que a pesar que por parte de nuestro cliente no van a existir modificaciones, puede darse que a medida que se desarrolla el proyecto, las prioridades, ideas ó conceptos se modifiquen y con ello los requerimientos pueden verse afectados, es por eso que esta sección es importante para mantener esta evaluación de los resultados de cada procesos , para de esta forma poder mantener claro el objetivo y los pasos para llegar a este.

Es por ello que para realizar la sección 4 los miembros de Alimnova® decidieron tener en cuenta:

Ilustración 5: Apreciación Global Validación de requerimientos

## 1.6 MODELO DE ESPIRAL PARA EL PROCESO DE INGENIERÍA DE REQUERIMIENTOS

Nuestro modelo general del proyecto es espiral. La grafica que se mostrara a continuación describirá las iteraciones durante el proceso de requerimientos.

Ilustración : Modelo espiral para el proceso de ingeniería de requerimientos. Tomado de SWEBOK: Software Requirementes Engineering Knowledge Area Description

La descripción de cada fase se hará a continuación:

### 1.6.1 Elicitación de Requerimientos:

También se denomina descubrimiento o adquisición de requerimientos. Se preocupa por definir de donde vienen los requerimientos y como pueden ser clasificados. Es el primer paso para la comprensión del problema.

Este paso se divide en 2 tópicos:

Fuentes de los requerimientos: Objetivos por los cuales se hacen los requerimientos. Dominio del conocimiento: se debe tener conocimiento de la aplicación a realizar. Ambiente operacional: la aplicación en que software se aplicara, cuáles son sus tiempos reales.

Ambiente organizacional: los requerimientos no deben afectar los procesos de negocio de quien desea poseer la aplicación

Técnicas de elicitación: Entrevistas al cliente. Escenarios que consiste en diseñar un modelo que permita generar preguntas de que pasaría si alguna cosa no se cumple. Prototipos. Observación.

### 1.6.2 Análisis De Requerimientos

Se concentra en la detección y resolución de conflictos entre los requerimientos. Descubrir los límites del sistema. Transformar los requerimientos de sistema a requerimientos de software. Se estudian las siguientes características:

#### 1.6.2.1 Clasificación de requerimientos:

¿Es funcional o no es funcional?

¿Es derivado de otro(s) requerimiento(s)?

¿Cuál es la prioridad del requerimiento?

El ámbito del requerimiento, es decir, la categoría

La estabilidad del requerimiento que no afecte si se modifica o elimina.

#### 1.6.2.2 Modelo conceptual:

El desarrollo del modelo conceptual es fundamental en el análisis de requerimientos, ya que es el inicio de la solución. Este modelo incluye los flujos de control, los datos, la interacción de los usuarios. La selección del modelo depende de varios factores entre ellos:

La naturaleza del problema

La experiencia del analista de requerimientos

Los requerimientos del proceso del cliente

Disponibilidad de los métodos y herramientas

#### 1.6.2.3 Diseño arquitectónico y asignación de requerimientos:

Incluye la descripción de subsistemas y componentes que cumplan con los requerimientos especificados. Permite un análisis detallado de las necesidades. Los requerimientos asignados a un componente específico.

### 1.6.3 Especificación De Requerimientos De Software

Este tópico está ligado a la estructura, calidad y verificación del documento de requerimientos. Existen 2 documentos esenciales y otros recursos:

Documentación de los requerimientos. Es decir las plantillas

SRS

IEEE std. 830-1998

IEEE std. 1362-1998

### 1.6.4 Validación De Los Requerimientos

Verificar por medio de listas de chequeo si los requerimientos están bien estructurados. Las listas de chequeo que utilizaremos serán las de CxOne.

#### 1.6.4.1 Conducto de las revisiones de los requerimientos

Se debe revisar constantemente los requerimientos, esto con el fin de que no afecten en el momento de la implementación, también cuando se desean hacer cambios a estos. Se deben revisar detenidamente los errores cometidos.

Este tema está muy relacionado con la trazabilidad, es decir determinar cuál es el origen, el destino.

# 2. PLAN DE REQUERIMIENTOS

El plan de requerimientos está basado en el modelo espiral para el proceso de ingeniería de requerimientos, especificado en la [[sección 1.6](#_1.6_MODELO_DE)].

## 2.1 Levantamiento de requerimientos

[[Sección 1.6.1](#_1.6.1_Elicitación_de)].

### 2.1.1 Fuentes de Requerimientos

#### 2.1.1.1 Objetivos del sistema

Se describen las formas o procesos llevados a cabo para deducir los objetivos

#### 2.1.1.2 Dominio de la aplicación

¿Qué es lo que realmente la aplicación debería hacer?

#### 2.1.1.3 Stakeholders

Las personas que estarán relacionadas directa y continuamente en el proceso del desarrollo de T-Monopoly serán los miembros de Alimnova®, ya que son los encargados de recoger y analizar la información, para luego desarrollar satisfactoriamente el juego T-Monopoly. Cada uno de los miembros de Alimnova® tiene un rol asignado y con este sus responsabilidades con el grupo y con su rol.

Otra persona que estará muy relacionada con este proceso, será el cliente y profesor, Miguel Torres, quien es el guía para Alimnova® y será el encargado de asesorarnos en el trabajo y el desarrollo de cada documento, todo esto con el fin de tener un buen ejercicio de aprendizaje en la materia Ingeniería de Software, con buenos resultados durante el proceso de desarrollo del juego T-Monopoly.

Además de las anteriores personas Alimnova® contará con el apoyo de los profesores de la facultad de Ingeniería de Sistemas, para asesorarnos acerca de temas como la arquitectura, el diseño, la implementación y demás aspectos en los que necesitemos asesoría.

**Ilustración 5: Stakeholders T-Monopoly**

#### 2.1.1.4 Ambiente Operacional

La sustentación y muestra de T-Monopoly será en la sala A de la facultad de Ingeniería, por lo tanto es necesario que casa una de las aplicaciones deben funcionar en esta sala, por lo tanto es importante que Alimnova® se asegure que esto suceda antes de que sea el día de realizar la presentación final.

También es importante tener en cuenta que tenemos determinadas restricciones dadas por el mismo cliente que son igual de importantes para el desarrollo de la aplicación, estas restricciones se convierten el ambiente operacional ya que es el medio en que se va a desarrollar la aplicación y si esta se tiene en cuenta, seguramente los resultados serán mucho más satisfactorios para Alimnova®

### 2.1.2. Técnicas de Recolección

Las técnicas de recolección depende la sección descrita anteriormente [[sección 2.1.1](#_2.1.1_Fuentes_de)], en ella interviene el arquitecto, donde se extrae información relevante de las fuentes. A partir de las diferentes técnicas tales como:

* Encuestas
* Escenarios
* Prototipos
* Reuniones entre los Stakeholders
* Observación

## 2.2 Análisis de Requerimientos

### 2.2.1. Riesgos Generales De Los Requerimientos

Presenta una lista de los riesgos más frecuentes que pueden ocurrir cuando los requerimientos ya se han especificado. Por ejemplo que haya conflicto entre requerimientos.

### 2.2.2. Clasificación De Los Requerimientos

Esta sección describe las categorías básicas y especificas que se manejaran en la documentación de los requerimientos, así como los criterios de priorización de los mismos.

Para realizar la clasificación de los requerimientos tuvimos en cuenta la pirámide de requerimientos, la cual a partir de los recursos y las características comunes permite dividir los requerimientos en diferentes tipos, a partir de:

http://www.ibmpressbooks.com/articles/article.asp?p=1152528&seqNum=2

La imagen a continuación muestra cómo se distribuyen los aspectos antes explicados, lo cuales muestran claramente las características que comparten en cada nivel de la pirámide:



http://ptgmedia.pearsoncmg.com/images/chap1\_9780321383006/elementLinks/01fig01.gif

Basándonos en el modelo anterior hemos decidido realizar la clasificación de los requerimientos funcionales en este SRS para el desarrollo del juego T- Monopoly, ya que asocia las necesidades de cada uno de los agentes involucrados relacionándolos con las actividades encaminadas para lograr la realización de cada una de estas tareas, de esta forma el proceso de desarrollo y diseño de la arquitectura será mucho más manejable, por la claridad que tiene cada uno de los procesos y actividades que se buscan realizar y así como también acerca de quién las realizará, permitiendo que realizar los diagramas, el diseño y posterior código sea consistente con los requerimientos planteados y con ello con las necesidades del cliente final.

Otro aspecto importante de esta clasificación es que permite relacionar los casos de uso de una forma clara y concisa, facilitando de esta forma la identificación de los casos de uso que faltan y de los que sobran acorde a las necesidades y características identificadas para el juego T-Monopoly, lo cual finalmente se traduce en requerimientos necesarios para el desarrollo de la aplicación.

Para realizar la clasificación tuvimos en cuenta los aspecto que muestra la pirámide, teniendo en cuenta que en las necesidades del cliente están las mismas restricciones planteadas por el cliente, Miguel Torres las cuales son: Interfaz gráfica fuerte, Persistencia y Arquitectura Cliente / Servidor. *[REFERENCIA: IBM REQUERIMENTS MANAGEMENT USING IBM RATIONAL REQUISITE PRO]*

http://www.ibm.com/developerworks/rational/library/04/r-3217/index.html

Agregamos los componentes que consideramos pertinentes y necesarios para el entendimiento y desarrollo del juego acorde al desarrollo que Alimnova®, los componentes identificados fueron:

**Jugadas especiales:** Corresponde a la tarea a realizar cuando un jugador saca más de tres pares seguidos, ya que si esto ocurre el puede escoger una propiedad

**Casillas especiales:** Corresponde a las casillas que tienen propiedades especiales, como por ejemplo, el arca comunal, los parqueaderos, los servicios públicos, causalidad.

**Transacciones:** Representa las operaciones realizadas por los jugadores o el sistema, entre las cuales están la negociación, la subasta, la venta y compra de tragos y/o propiedades

**Partida:** Representa el inicio, finalización de la partida y las actividades llevadas a cabo por el banco, durante el desarrollo del juego

A partir de estas tres grandes bases se despliegan las características correspondientes a cada restricción, de forma que a cada restricción la relacionamos con su labor en el juego, el desempeño que va a tener en este ó las actividades que esta desarrolla bien sea durante el juego, respecto o a la funcionalidad y la relación con la visualización que tiene el usuario del juego ó para poder crear y mantener el juego como tal, a partir de estas posibles relaciones que se puede tener en el juego generamos las características que tendrían cada una de las restricciones planteadas.

Luego de tener las características de las relaciones podremos identificar los casos de uso asociados a cada uno, las relaciones que pueden tener y cómo se van a desarrollar dentro del sistema y el desarrollo del juego como tal, respecto tanto a funcionalidad del juego, la forma en que el usuario ve el juego, la parte correspondiente al sistema; crear conexiones, realizar partidas, gestionar dinero, y demás aspectos correspondientes con la lógica del juego, cada componente de esta forma tendrá un caso de uso asociado a partir de la tarea que realice en el juego y que contribuya a asegurar el buen funcionamiento del mismo.

Teniendo los casos de uso podemos identificar los escenarios relacionados con cada uno de ellos y generar las pruebas para desarrollar la evaluación de cada uno de ellos, facilitando así el proceso de evaluación también, puesto que a partir del diagrama generado a partir de estas relaciones podemos identificar las necesidades que debe suplir y cómo lo hace.

Otro componente importante son las especificaciones:

**Software:** Relacionado con el sistema operativo y demás herramientas que empleemos y necesitemos durante el desarrollo de este juego

**Hardware:** Corresponde a los equipos que emplearemos y en los que debe correr nuestra aplicaciones, acorde a las restricciones dadas por Miguel Torres.

**Desempeño:** Relacionado con el logro de los objetivos propuestos por Alimnova® acerca de lo que debe hacer el juego

Acorde a lo antes expuesto la adaptación de la pirámide acorde a nuestros componentes, quedaría así:



Ilustración 6: Clasificación de requerimientos

http://books.google.com.co/books?id=3XSL7kS8GEgC&pg=PA80&dq=REQUIREMENT+PYRAMID+SOFTWARE&ei=tOWaS-YWhPLIBPmyyOMK&cd=3#v=onepage&q=REQUIREMENT%20PYRAMID%20SOFTWARE&f=false

<http://books.google.com.co/books?id=h4pPpXp-xrEC&pg=PA230&dq=REQUIREMENT+PYRAMID+SOFTWARE&ei=tOWaS-YWhPLIBPmyyOMK&cd=1#v=onepage&q=&f=false>

http://books.google.com.co/books?id=HVvwMGGQqpoC&pg=PA145&dq=REQUIREMENT+PYRAMID&ei=mOWaS9XNJpT4zATBs4T9Cg&cd=1#v=onepage&q=REQUIREMENT%20PYRAMID&f=false

*<http://books.google.com.co/books?id=Mjn53zPoDQIC&pg=PA212&dq=REQUIREMENT+PYRAMID&ei=mOWaS9XNJpT4zATBs4T9Cg&cd=3#v=onepage&q=REQUIREMENT%20PYRAMID&f=false>*

[*http://books.google.com.co/books?id=MSYLHYhoU3cC&pg=PA105&dq=NEED+DIAGRAM&ei=LeaaS\_-RFovKzgS0lvWMCw&cd=1#v=onepage&q=NEED%20DIAGRAM&f=false*](http://books.google.com.co/books?id=MSYLHYhoU3cC&pg=PA105&dq=NEED+DIAGRAM&ei=LeaaS_-RFovKzgS0lvWMCw&cd=1#v=onepage&q=NEED%20DIAGRAM&f=false)

[*http://books.google.com.co/books?id=e7ZhVD3JejAC&printsec=frontcover&dq=REQUIREMENTS&ei=yuiaS42PHoPUzQSH9aj8DQ&cd=2#v=onepage&q=&f=false*](http://books.google.com.co/books?id=e7ZhVD3JejAC&printsec=frontcover&dq=REQUIREMENTS&ei=yuiaS42PHoPUzQSH9aj8DQ&cd=2#v=onepage&q=&f=false)

[*http://books.google.com.co/books?id=Bg4NAQAAIAAJ&pg=PA6&dq=DIAGRAMAS+DE+NECESIDADES&ei=FuaaS7ajBpT4zATBs4T9Cg&cd=1#v=onepage&q=DIAGRAMAS%20DE%20NECESIDADES&f=false*](http://books.google.com.co/books?id=Bg4NAQAAIAAJ&pg=PA6&dq=DIAGRAMAS+DE+NECESIDADES&ei=FuaaS7ajBpT4zATBs4T9Cg&cd=1#v=onepage&q=DIAGRAMAS%20DE%20NECESIDADES&f=false)

### 2.2.3 Modelo Conceptual

Es una representación gráfica de todo el problema, identificando y analizando las amenazas.

MODELO TENTATIVO <http://www.cauvin.biz/RequirementsConcepts.png>

### 2.2.4 Diseño Arquitectónico y Asignación de Requerimientos

Se realizara un diagrama de componentes.

DIA. TENTATIVO <http://www.agilemodeling.com/images/models/componentDiagramUML2.jpg>

DIAG. TENTATIVO <http://xml.netbeans.org/text-edit/uml/completion/components-diagram.gif>

### 2.2.5 Modelo Del Dominio

#### 2.2.5.1 Diagrama

#### 2.2.5.2 Documentación de las Clases Conceptuales

#### 2.2.5.3 Documentación de las Relaciones

# Especificación De Requerimientos

## 3.1. Documento de Especificación de Requerimientos

LINK AL DOCUMENTO DE ESPECIFICACIÓN DE REQUERIMIENTOS

## 3.2. Especificación de las Categorías Funcionales

De acuerdo a la clasificación descrita en la sección 2.2.2 [[sección 2.2.2](#_2.2.2._Clasificación_De)] se describirá cual es la razón de ser de la clasificación.

Requerimientos Funcionales

### 3.2.1. Casillas Especiales

En el tablero del monopolio existen diferentes tipos de casillas; a las cuales se les debe dar un trato especial dentro de los requerimientos. La división de requerimientos funcionales incluye una categoría denominada casillas especiales.

#### 3.2.1.1 Arca comunal

Corresponde a las tarjetas que muestran leyendas que se deben aplicar al jugador cuando caen en una casilla como esta. El número de cartas de arca comunal es 12.

|  |  |
| --- | --- |
| IDENTIFICADOR | LEYENDA |
| 1 | Utiliza esta tarjeta para salir del CAI. |
| 2 | Vas directo al CAI, por robarte el dinero del Hippie [[Ver sección 3.1.3](#_3.1.3_Hippie)]. |
| 3 | Pasa por la PROPIEDAD X, si pasas por la salida cobra $200.000 |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |

Tabla X: Descripción de las leyendas de Arca comunal

#### 3.2.1.2. Casualidad

Corresponde a las tarjetas que muestran leyendas que se deben aplicar al jugador cuando caen en una casilla como esta. El número de cartas de casualidad es 12. Es similar al de arca comunal, pero difiere en que las leyendas de esta tarjeta 9 son buenas para el jugador, mientras que en Arca Comunal el número de leyendas buenas se reduce a 6.

|  |  |
| --- | --- |
| IDENTIFICADOR | LEYENDA |
| 1 | Ganas $300.000 como mejor disfraz en Opera. |
| 2 | Valido para salir del CAI. |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |

Tabla X: Descripción de las leyendas de Casualidad

#### 3.2.1.3 Hippie

Esta casilla en el tablero del Monopolio original es denominada PARADA LIBRE. Alimnova® decidió modificar dicha regla de manera que este adaptado a la temática de T-Monopoly®. Al caer en la casilla de Hippie el banco cobrara el valor correspondiente a una manilla el cual es $10.000.

#### 3.2.1.4 CAI

En la casilla del Monopolio original corresponde a la CÁRCEL. Se divide en 2 categorías:

1. Ida al CAI, se presentan 3 situaciones para que ocurra este hecho:
   1. Caer en la casilla ir al CAI
   2. Que aparezca en una leyenda de la tarjeta de Arca Comunal.
   3. Que aparezca en una leyenda de la tarjeta de Casualidad.

Aclaración: en el juego original del monopolio hay otra situación en la que un jugador va a la cárcel, esta corresponde cuando el jugador saca 3 pares, pero Alimnova® modifico esta regla [[Ver sección 3.2](#_3.2_JUGADAS_ESPECIALES)].

1. Salida del CAI, se presentan 4 situaciones para que ocurra este hecho:
   1. El jugador cuando se encuentre en su turno saque pares, para este hecho tendrá 3 oportunidades.
   2. Que aparezca en una leyenda de la tarjeta de Arca Comunal.
   3. Que aparezca en una leyenda de la tarjeta de Casualidad.
   4. El jugador si desea pagará una fianza de $50.000

#### 3.2.1.5 Salida

Este aspecto corresponde a la casilla de salida del juego, en ver la cual se encuentran inicialmente las fichas de todos los jugadores y luego es donde los jugadores piden dinero al banco, cada que pasen por esa casilla, siempre y cuando cumplan con los requisitos para poder hacerlo, (durante esa vuelta no vieron estar en la cárcel). El dinero cobrado corresponde a $200.000 para T-Monopoly®.

http://www.scribd.com/doc/2929604/Reglas-MonopolySpanish

#### 3.2.1.6 Impuestos

Corresponde a la cantidad de dinero que debe cancelar el jugador por las propiedades o tragos que tenga en el momento que caiga en esa casilla, es por esto que los impuestos están divididos entre:

3.2.1.6.1 Posesiones De Lujo

Corresponde al dinero que debe cancelar el jugador por cada trago que tenga en el momento que cae en esta casilla, denominada en el tablero como: “Impuesto sobre posesiones de lujo”.

3.2.1.6.2 Posesiones

Corresponde a la cantidad de dinero que debe pagar el jugador, acorde al número de por propiedades que tiene en el momento de caer en esa casilla, denominada en el tablero como:”Impuestos sobre ingresos”.

### 3.2.2 Jugadas Especiales

Corresponde a la situación en la que el jugador lanza los dados y obtiene pares en tres lanzamientos, esto tiene un tratamiento especial y es una variación de la regla del Monopoly clásico, ya que en vez de ser llevado a la cárcel, el jugador puede escoger una propiedad para que esta le sea asignada.

### 3.2.3 Transacciones

#### 3.2.3.1 Negociación

Las reglas de Monopolio tradicional en la sección de venta de propiedades proponen lo siguiente:

Todo jugador que sea dueño de solares sin edificar, ferrocarriles y servicios públicos (pero no edificios) podrá venderlos a cualquier otro jugador en una operación privada y por la cantidad que convengan ambos. No obstante, no podrá venderse ningún solar que pertenezca a un grupo de color en el que haya algún otro solar que contenga edificios. Antes de que el dueño pueda vender un solar de dicho grupo de color, tendrá que vender al Banco dichos edificios.

<http://www.hasbro.com/common/instruct/Monopoly(Spanish).pdf>

Para efectos de nuestra adaptación y con el fin de definir un marco para la definición de requerimientos de la categoría de Negociación a continuación se presenta el alcance que definimos para ella la cual es análoga a la establecida por las reglas del Monopoly tradicional:

La negociación se basa en dos aspectos:

#### 3.2.3.2 Subasta

En cuanto a la subasta las reglas del Monopoly tradicional proponen lo siguiente:

Si el jugador opta por no comparar, la propiedad es puesta inmediatamente a la venta en subasta por el Banco y es vendida al mejor postor. El comprador pagará al Banco la cantidad estipulada en la licitación y recibirá la carta correspondiente de Escritura de Propiedad. Todos los jugadores pueden licitar, incluso el jugador que no acepto la opción de compra al precio impreso en la propiedad. Cualquier precio puede servir de base para comenzar la licitación.

Para nuestra adaptación la subasta se dará en dos situaciones:

#### 3.2.3.3 Propiedades

En cuanto a propiedades, T-Monopoly manejará propiedades reales correspondientes a los bares, discotecas y restaurantes ubicados en la “Zona T” de Bogotá.

Ubicada en la localidad de Chapinero, La Zona T es un lugar bastante concurrido por su gran oferta de restaurantes y bares de diversos estilos. Consiste en dos calzadas peatonales que se intersectan en forma de T. Está ubicada en frente al centro comercial Andino entre la “Calle del sol” y la calle 82 y las carreras 12 y 13. Tomado de <http://bogowiki.org/zona%20t>

Consultas: <http://www.bogota-dc.com/rest/bog-res4.htm>, <http://bogowiki.org/zona%20t>

Al igual que en Monopoly, T-Monopoly se basara en la compra, venta, hipoteca y negociación de las propiedades anteriormente listadas. Cada propiedad cuenta con atributos que las identifican dentro del juego, los más destacados son su nombre, su precio en el mercado, su cover, su incremento de cover o impuesto adicional por posesión de Tragos (bienes sobre cada propiedad)y el precio de adquisición de dichos tragos.

#### 3.2.3.4 Trago

T-Monopoly emula también la posibilidad de crear distintas edificaciones sobre las propiedades del juego las cuales están denominadas como Tragos. Este es un componente del juego que requiere gran detalle en su especificación por lo que se presentará en paralelo con las reglas de Monopoly.

|  |  |
| --- | --- |
| Monopoly | T-Monopoly |
| http://oligopoly.files.wordpress.com/2008/12/monopoly-house-150.jpg Casa | http://gulmaros.files.wordpress.com/2009/08/cerveza.jpg **Cerveza** |
| http://www.swtourism.org.uk/files/imagelibrary/finance/hotel3.jpg Hotel | http://www.danielmcbee.com/whiskey.jpg **Whisky** |

Tabla X: Equivalencias de casas y hoteles entre Monopoly y T-Monopoly.

|  |  |  |  |
| --- | --- | --- | --- |
| HECHO | MONOPOLY | EQUIVALENCIA DE HECHO | ADAPTACIÓN A T-MONOPOLY |
| Las Casas | Cuando un jugador posee todas las propiedades de un grupo del mismo color puede comprar casas al Banco y levantarlas en dichas propiedades.  El jugador podrá comprar y construir en cualquier momento todas las casas que juzgue conveniente y le permita su situación económica. | **Las Cervezas** | La administración de las cervezas se tratará con la misma temática de las casas de Monopoly. |
| Los Hoteles | Antes de poder comprar un edificio de hotel, el jugador ha de tener cuatro casas en cada solar de un grupo completo de un solo color. Cuando lo logre podrá comprar del Banco un hotel para levantarlo en cualquier solar de dicho grupo de un solo color, entregándole al Banco a cambio del mismo las cuatro casas allí existentes y el precio del hotel que indique la Escritura de propiedad. En cada solar no puede construirse más de un hotel | **El Whisky** | La administración del Whisky se tratará con la misma temática de los hoteles de Monopoly. |
| Escasez de Edificios | Cuando el Banco no tenga casas para vender, los jugadores que deseen construir tendrán que esperar para hacerlo a que algún otro jugador devuelva o venda al Banco sus casas. Si se dispone de una pequeña cantidad de casas y hoteles y dos o mas jugadores desean comprar un número superior al que tiene el banco, las casas y hoteles tienen que venderse en pública subasta al mejor postor. | **Ley Seca** | La imposibilidad de adquisición de Tragos se manejará de manera análoga a la gestión de la Escasez de Edificios. |

Tabla X: Analogía de reglas y hechos con respecto a adquisición de bienes sobre propiedades.

### 3.2.4 Partida

#### 3.2.4.1 Inicio Partida

#### 3.2.4.2 Finalización Partida

#### 3.2.4.3 Banco

## 3.3 Especificación de las Categorías No Funcionales

Requerimientos No Funcionales

### 3.3.1 Restricciones

#### 3.3.1.1 CLIENTE/SERVIDOR

#### 3.3.1.2 PERSISTENCIA

#### 3.3.1.3 GUI

### 3.3.2 Especificaciones

#### 3.3.2.1 Hardware

#### 3.3.2.2 Software

#### 3.3.2.3 Desempeño

#### 3.3.2.4 Interfaces con el usuario

#### 3.3.2.5 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 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 : División por Funcionalidades

Funcionalidad 1: Jugar

Uso de la para la documentación de los requerimientos asociados a esta funcionalidad.

Funcionalidad 2: Registro y Autenticación

Uso de la para la documentación de los requerimientos asociados a esta funcionalidad.

Funcionalidad 3: Administración

Uso de la para la documentación de los requerimientos asociados a esta funcionalidad.

Funcionalidad 4: Consultas

Uso de la para la documentación de los requerimientos asociados a esta funcionalidad.

Cruce de funcionalidades

Uso de la 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.

#### 3.3.2.6 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.

#### 3.3.2.7. 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 : Características de Confiabilidad

#### 3.3.2.8 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 : Características de Disponibilidad

#### 3.3.2.8 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 : Características de Seguridad

#### 3.3.2.9 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 : Características de Mantenibilidad

#### 3.3.2.10 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 : Portabilidad del Sistema

#### 3.3.2.11 Persistencia

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

Ilustración : 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, consultadas 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. Validación de requerimientos

## 4.1. Revisión de conceptos de documentación

Revisión de los conceptos de la sección 3.

## 4.2. Análisis de Trazabilidad

## 4.3. Evaluación de Requerimientos de Software

Listas de chequeo.

## 

# 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 Specificacitions, 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