2010

06/04/2010

SRS: SOFTWARE 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, Arquitecto. |
| 1.1.0 | Marzo 24 / 2010 | Sección 3.3.2.4 y 3.3.2.5 | Redacción y estructura de las secciones Sección 3.3.2.4 y 3.3.2.5 | Andrea Fajardo, Arquitecto. |
| 1.1.1 | Marzo 24 / 2010 | Sección 3.3.2.4 y 3.3.2.5 | Duda acerca de sección 3.2.2.5 | Andrea Fajardo, Arquitecto. |
| 1.2.1 | Marzo 24/2010 | Sección 2.2.1, 4 | Documentación de la sección | Laura Arias, Director de proyectos. |
| 1.3.0 | Marzo  25/2010 | 2.2.2 , 2.2.3 | Documentación de las secciones mencionadas | William Jiménez administrador de configuraciones y documentación. |
| 1.4.0 | Marzo  25/2010 | 3.3.2.2 | Documentación de la sección mencionada | Germán Morales  Director de desarrollo. |
| 1.5.0 | Marzo 25/2010 | 3.3.2.9 – 3.3.2.11, 1.3 | Documentación y definición de los diccionarios | Laura Arias, Director de proyectos. |
| 1.6.0 | Marzo 25/2010 | 2.1.1, 3.3.1, 3.2.4, 3.3.3.12 | Adición de las secciones mencionadas | David Suárez, Director de Calidad. |
| 1.7.0 | Marzo 25/2010 Hora: 11:52 pm | 5 anexos | Inclusión de anexos | Laura Arias, Director de proyectos. |
| 1.8.0 | Marzo 25/2010 | Sección 3.3.2.5 a la Sección 3.3.2.9 | Redacción y estructura de las secciones | Andrea Fajardo |

Tabla : Historial De Cambios

CONTENIDO

[1.3 DEFINICIONES, ACRÓNIMOS, Y ABREVIACIONES 12](#_Toc257323985)

[1.3.1 Diccionario del Juego 13](#_Toc257323986)

[Tabla X: Diccionario del juego 14](#_Toc257323987)

[1.3.2 Diccionario Técnico 14](#_Toc257323988)

[1.4 REFERENCIAS 15](#_Toc257323989)

[1.5 APRECIACIÓN GLOBAL 15](#_Toc257323990)

[1.6 MODELO DE ESPIRAL PARA EL PROCESO DE INGENIERÍA DE REQUERIMIENTOS 19](#_Toc257323994)

[1.6.1 Elicitación de Requerimientos: 19](#_Toc257323995)

[1.6.2 Análisis De Requerimientos 20](#_Toc257323996)

[1.6.3 Especificación De Requerimientos De Software 20](#_Toc257323997)

[1.6.4 Validación De Los Requerimientos 21](#_Toc257323998)

[2. PLAN DE REQUERIMIENTOS 21](#_Toc257323999)

[2.1 Levantamiento de requerimientos 21](#_Toc257324000)

[2.1.1 Fuentes de Requerimientos 21](#_Toc257324001)

[2.1.2. Técnicas de Recolección 24](#_Toc257324003)

[2.2 Análisis de Requerimientos 24](#_Toc257324004)

[2.2.1. Riesgos Generales De Los Requerimientos 24](#_Toc257324005)

[2.2.2. Clasificación De Los Requerimientos 27](#_Toc257324013)

[2.2.3 Modelo Conceptual 33](#_Toc257324016)

[2.2.4 Diseño Arquitectónico y Asignación de Requerimientos 35](#_Toc257324018)

[2.2.5 Modelo Del Dominio 35](#_Toc257324021)

[3. Especificación De Requerimientos 35](#_Toc257324022)

[3.1. Documento de Especificación de Requerimientos 35](#_Toc257324023)

[3.2. Especificación de las Categorías Funcionales 35](#_Toc257324024)

[Requerimientos Funcionales 35](#_Toc257324025)

[3.2.1. Casillas Especiales 35](#_Toc257324026)

[3.2.2 Jugadas Especiales 38](#_Toc257324027)

[3.2.3 Transacciones 38](#_Toc257324028)

[3.2.4 Partida 42](#_Toc257324030)

[3.3 Especificación de las Categorías No Funcionales 43](#_Toc257324031)

[Requerimientos No Funcionales 43](#_Toc257324032)

[3.3.1 Restricciones 43](#_Toc257324033)

[3.3.2 Especificaciones 43](#_Toc257324034)

[4.1. Revisión de conceptos de documentación 52](#_Toc257324042)

[4.2. Análisis de Trazabilidad 53](#_Toc257324043)

[4.3. Evaluación de Requerimientos de Software 54](#_Toc257324044)

[5. Anexos 55](#_Toc257324045)

[5.1 Encuestas 55](#_Toc257324046)

[5.2 Especificación De Requerimientos 55](#_Toc257324047)

[5.3 Documento De Trazabilidad 55](#_Toc257324048)

[5.4 Listas De Chequeo 55](#_Toc257324049)

[REFERENCIAS 55](#_Toc257324050)

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

### 1.3.1 Diccionario del Juego

|  |  |
| --- | --- |
| LETRA | PALABRAS |
| A | * ANFITRIÓN: jugador que crea una partida * ARCA COMUNAL: tarjeta que contiene sentencias que se le deben aplicar a un jugador |
| B | * BASE INICIAL: el precio base con el que se iniciara la subasta * BANCA ROTA: cuando un jugador tiene todas sus propiedades hipotecadas y no tiene dinero para pagar el cover |
| C | * CAI: En el juego original de monopolio es la cárcel * CASILLAS ESPECIALES: propiedades o tarjetas que se tratan de manera diferente, es decir, su funcionalidad es diferente. * CASUALIDAD: tarjeta que contiene sentencias que se le deben aplicar a un jugador * CERVEZAS: En el juego original de monopolio es el referente a las casas * COVER: en el juego original de monopolio significa alquiler de una propiedad |
| D | * DUEÑO: jugador que posee una propiedad. |
| e |  |
| F | * FICHA: el objeto que identificara a un jugador. |
| g |  |
| H | * HIPOTECAR: vender sus propiedades a mitad de precio al banco, con el fin de obtener dinero. |
| I | * INTERCAMBIAR: hacer un trueque de propiedades entre 2 jugadores. * INVITADO: jugador que se une a una partida |
| J | * JUGADOR: sinónimo de participante, quien jugara en una partida |
| k |  |
| l |  |
| M | * Mayor de edad: el jugador que posee todas las propiedades de un mismo color. |
| n |  |
| o |  |
| P | * PARQUEADERO: en el juego original del monopolio se refiere a los ferrocarriles * PERFIL: tiene atributos de nombre, contraseña. * PUJAR: aumentar el valor de la subasta |
| q |  |
| R | * RANKING: nombre del jugador, jugadas ganadas, perdidas y empatadas de todos los jugadores. |
| S | * SUBASTAR: el banco vende la propiedad al mayor postor, todos los jugadores que tengan el dinero de la subasta podrán participar. Al igual que se podrá retirar de la subasta. El banco colocara la base inicial. |
| t |  |
| u |  |
| v |  |
| w | * WHISKY: en el juego original del monopolio referente a los hoteles |
| X |  |
| y |  |
| z |  |

Tabla X: Diccionario del juego

### 1.3.2 Diccionario Técnico

|  |  |
| --- | --- |
| LETRA | PALABRAS |
| A |  |
| B |  |
| C |  |
| D |  |
| e |  |
| F |  |
| g |  |
| H |  |
| I |  |
| J |  |
| k |  |
| l |  |
| M |  |
| n |  |
| o |  |
| P |  |
| q |  |
| R |  |
| S |  |
| t |  |
| u |  |
| v |  |
| w |  |
| X |  |
| y |  |
| z |  |

Tabla X: Diccionario de técnico

## 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 (Software 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:

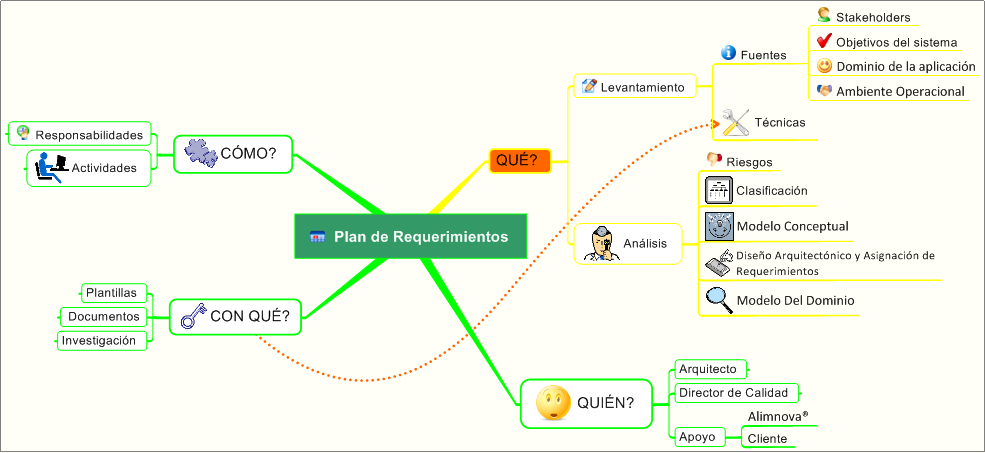


Ilustración : Apreciación Global PLAN REQUERIMIENTOS

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:

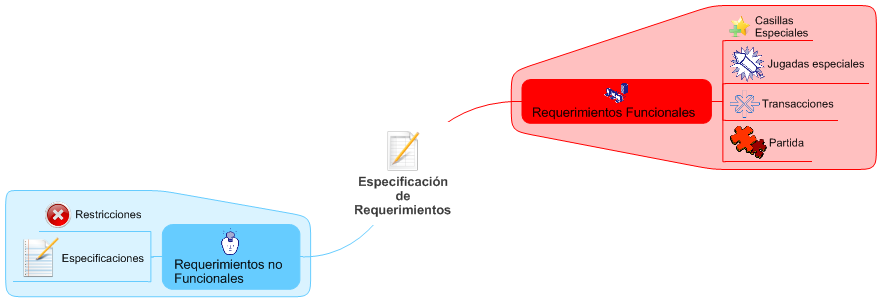


Ilustración 4: Apreciación Global especificación de requerimientos

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

Cuando se describen los objetivos que debe cumplir la aplicación a fin de llenar alguna necesidad de negocio, por lo tanto es una muy importante fuente de requerimientos. Sin embargo en el caso de Alimnova® el principal objetivo del desarrollo de esta aplicación es el aprendizaje del proceso y en coherencia con esto no se tomaría en cuanta en el proceso de levantamiento de requerimientos.

#### 2.1.1.2 Dominio de la aplicación

Esta fuente de requerimientos implica el amplio conocimiento de la aplicación(T-Monopoly) a realizar, en este caso para Alimnova® es muy importante esta sección en la medida en que la aplicación esta basada en un juego existente y por lo tanto toda la aplicación y una gran fuente de requerimientos son las reglas descritas por el juego Monopoly®, sin que esto quiera decir que Alimnova® no pueda modificar o agregar algunas reglas, como de hecho sucede. Adicionalmente se permite así que se tenga una visión mucho mas clara de lo que la aplicación debe hacer. El proceso desarrollado por Alimnova para cumplir con este pleno entendimiento de la aplicación fueron varias reuniones con el fin de que todos lo integrantes jugaran varias partidas de Monopoly® de Hasbro, el juego de mesa. De esta forma se entendieron las reglas generales del juego y se hizo un levantamiento inicial de requerimientos principalmente funcionales ya que están directamente relacionados con lo que es el juego en si.

También se hicieron varias reuniones con el fin de jugar en la aplicación Monopoly® realizada por Hasbro, esta ves se hizo con el fin de recolectar ideas para la interfaz grafica de Usuario(GUI) y un mejor entendimiento en general de cómo funcionaria el juego en un computador y como seria la interacción entre dos usuarios para llevar a cabo una negociación.

#### 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.

Finalmente, es importante mencionar que todos los Stakeholders relacionados con el proyecto actual son una fuente importante de requerimientos ya que aportan de forma directa o indirecta a restricciones, requerimientos de negocio o requerimientos del juego en general, es decir estos Stakeholders mencionados anteriormente ayudaran en el proceso de mejorar la descripción de requerimientos y de esta forma tener un mejor entendimiento de la aplicación a desarrollar, T-Monopoly®

**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 [[ver seccion 3.3.1](#_3.3.1_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®.

#### 2.1.1.5 Casos de uso

Los casos de uso realizados por Alimnova® en el primer hito (SPMP) serán la principal fuente de requerimientos, ya que en estos se describen las funcionalidades que debe tener T-Monopoly. Con esto se quiere indicar que los requerimientos funcionales se tomaran de los casos de uso que se hicieron anteriormente, siempre y cuando se tenga una buena descripción de casos de uso.

### 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, Andrea Fajardo, donde se extrae información relevante de las fuentes. A partir de las diferentes técnicas tales como:

* Encuestas [[**Anexo 1** **Encuestas Realizadas**](#_5.1_Encuestas)].

## 2.2 Análisis de Requerimientos

### 2.2.1. Riesgos Generales De Los Requerimientos

Los riesgos más frecuentes en cualquier proyecto de software están relacionados con el proceso de levantamiento de requerimientos, razón por la cual se deben tener muy en cuenta.

El encargado de administrar los riesgos de los requerimientos, será David Suarez, Director de calidad y manejo de riesgos, a través de la herramienta RiskyProject Professional 3.2.

A continuación se describirán los principales riesgos que se podrían presentar en la elaboración de los requerimientos de T-Monopoly®:

RRP1 = Riesgos de los Requerimientos Podrían ocurrir

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id Riesgo | Riesgo | Descripción | Tratamiento | Solución Particular |
| RRP1 | Análisis pobre del impacto | No entender que los requerimientos cambian y afectan a un conjunto de estos requerimientos | Mitigación | Trazabilidad de los requerimientos **[**[**Sección 4.2**](#_4.2._Análisis_de)**]** |
| RRP2 | No tener definido el objetivo del requerimiento | ¿Cuál es la razón por la que el requerimiento está presente? | Mitigación | En la plantilla de especificación de los requerimientos hay un campo denominado “RAZÓN” **[**[**Sección 3.1]**](#_3.1._Documento_de) |
| RRP3 | Los requerimientos deben ser escritos en términos que se puedan verificar. | Un requerimiento debe ser verificable, de esta manera no sería un requerimiento | Mitigación | En la plantilla de especificación de los requerimientos hay un campo denominado “Método V&V” **[**[**Sección 3.1**](#_3.1._Documento_de)**]** donde también se especificaron los métodos de V&V que existen. |
| RRP4 | Los requerimientos no cumplan con las características mínimas. | Las características son: no ambiguos, precisos, concisos, completos, únicos. | Mitigación | Al finalizar la especificación de requerimientos se realizara la lista de chequeo de CxOne denominada CxOne\_Requeriments |
| RRP5 | Que los requerimientos cambien o se eliminen. | Un requerimiento puede no ser aprobado y necesite ser cambiado o eliminado. | Mitigación | Trazabilidad de los requerimientos **[**[**Sección 4.2**](#_4.2._Análisis_de)**]** |
| RRP6 | Requerimientos inválidos | Requerimientos que se salgan de los limites y que no tengan que ver con lo que el cliente requiere | Mitigación | Semanalmente Laura Arias, Director de proyectos tiene reuniones con el cliente con el fin de informar. |

Tabla X: Riesgos más frecuentes de los requerimientos

Al igual que los riesgos que hay poca probabilidad de que ocurran en la elaboración de los requerimientos de T-Monopoly®:

RRN1 = Riesgos de los Requerimientos No ocurrirán.

|  |  |  |
| --- | --- | --- |
| Id Riesgo | Riesgo | Razón |
| RRN1 | Expectaciones No Realistas del Cliente | El cliente Miguel Torres, definió únicamente 3 restricciones, estas están aterrizadas. |
| RRN2 | No involucrar al cliente en el proceso de construcción de requerimientos | Semanalmente Laura Arias, Director de proyectos tiene reuniones con el cliente con el fin de informar avances o cambios. |

Tabla X: Riesgos poco frecuentes de los requerimientos

En la sección 4.2 se tiene una lista de chequeo que mantiene al día los riesgos de los requerimientos.

**REFERENCIA RIESGOS NO PROBABLES** How Agile Practices Reduce

Requirements Risk

by Ellen Gottesdiener

Referencia: Risk and

Requirements-

Based Testing

**James Bach, Independent Consultant**

I define requirements as the set of ideas

that collectively define quality for a particular product.

product

**BUSCAR LIBRO:** *Exploring Requirements: Quality*

*Before Design* (Dorset House, 1989

**LISTA DE CHEQUEO :** [**http://www.incose.org/tgcc/LIBRARY/CAI%20Checklist%20of%20Risk%20Factors.pdf**](http://www.incose.org/tgcc/LIBRARY/CAI%20Checklist%20of%20Risk%20Factors.pdf)

[**http://www.stsc.hill.af.mil/crosstalk/2006/12/0612Young\_Tab5.jpg**](http://www.stsc.hill.af.mil/crosstalk/2006/12/0612Young_Tab5.jpg)

### 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:

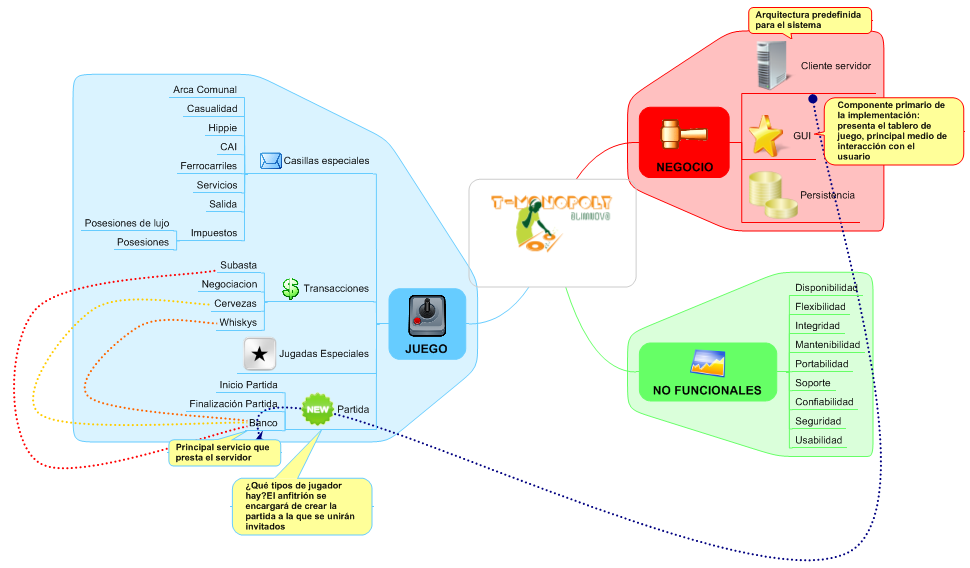


Ilustración X: Clasificación de requerimientos y relación de categorías

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>*

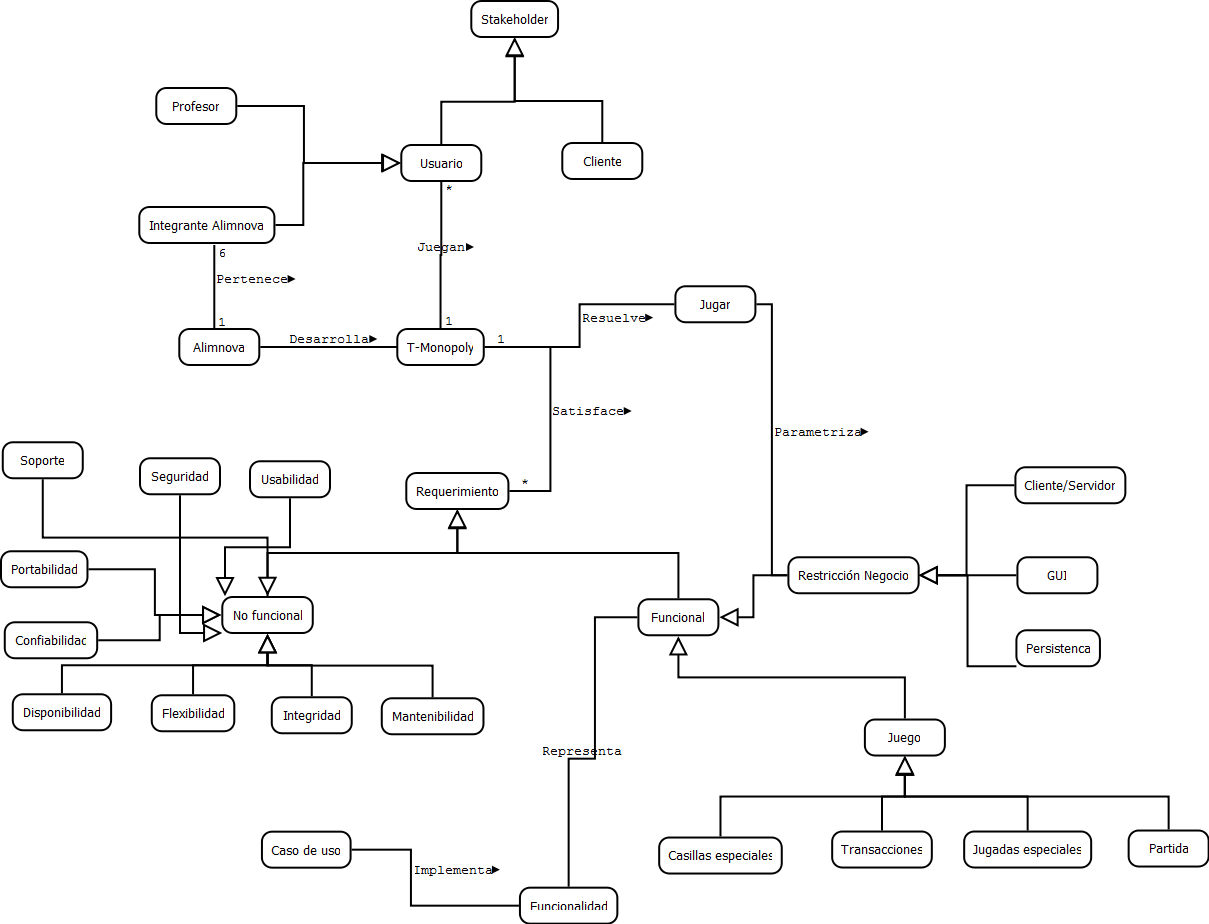
[*http://books.google.com.co/books?id=MSYLHYhoU3cC&pg=PA105&dq=NEED+DIAGRAM&ei=LeaaS\_-RFovKzgS0lvWMCw&cd=1#v=onepage&q=NEED%20DIAGRAM&f=false*](http://books.google.com.co/books?id=MSYLHYhoU3cC&pg=PA105&dq=NEED+DIAGRAM&ei=LeaaS_-RFovKzgS0lvWMCw&cd=1#v=onepage&q=NEED%20DIAGRAM&f=false)

[*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

Con el fin de tener una visión global del hito de requerimientos del proyecto se ha optado por desarrollar un modelo conceptual de este. En este modelo se presentan las relaciones identificadas a partir del entendimiento del problema, utilizando los factores mas relevantes que intervienen como stakeholders, tareas a realizar y la jerarquía de requerimientos a desarrollar.



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

[[**Documento de Especificación de Requerimientos**](#_5.2_Especificación_De)].

## 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

Este corresponde a como se hará el inicio de una partida. Alimnova® definió que para poder iniciar una partida, el primer paso ha realizar es que un jugador que ya ha creado su perfil, cree un partida, convirtiéndose así en anfitrión y permitiendo entonces que los demás jugadores conectados al servidor se puedan agregar a la partida. Para el desarrollo de esta aplicación se ha definido que el mínimo numero de jugadores para poder iniciar un partida serán de 3 jugadores.

#### 3.2.4.2 Finalización Partida

En cuanto a la finalización de una parida Alimnova® realizo una modificación a las reglas del juego original Monopoly®, ya que se definió que una partida finaliza cuando un jugador queda en bancarrota y en tal caso el ganador de la partida será aquel jugador que tenga la mayor cantidad de dinero, representado bien sea en dinero en efectivo como en los bienes que aquel jugador tenga(Propiedades, Cervezas y Whisky).

Adicionalmente un jugador se declara en bancarrota cuando la diferencia entre el total de activos (bienes y dinero) y lo que debe pagar en determinado turno es negativa.

Una variación de la finalización de la partida, será en el caso especifico en que solo quede un jugador activo a razón del abandono, ya sea voluntario o por un problema en la conexión con el servidor, de una partida por parte de los demás jugadores.

#### 3.2.4.3 Banco

El banco guarda, además del dinero del banco, las cartas de las propiedades de todo el juego, cervezas y Whiskys antes de que estas sean adquiridas por los jugadores. El banco paga sueldos, vende y subasta propiedades a los jugadores; les vende cervezas y Whiskys cuando el jugador sea mayor de edad y les presta dinero sobre hipotecas de propiedades.

Se le paga al Banco, todas las multas, prestamos e intereses y el precio de todas las propiedades que venda y remate.

El banco no se arruina nunca, si se queda sin dinero puede emitir toda la moneda propia que necesite.

## 3.3 Especificación de las Categorías No Funcionales

Requerimientos No Funcionales

### 3.3.1 Restricciones

En esta sección se definirán las principales restricciones identificadas por Alimnova®, las cuales a su ves serán una fuente de requerimientos de negocio. A continuación se mencionaran estas restricciones, las cuales son impuestas por el cliente, y reflejan las tres divisiones que se hicieron para sacar estos requerimientos: Cliente/Servidor, GUI y Persistencia.

* La aplicación debe ser en una arquitectura cliente/servidor.
* La aplicación a pesar de ser desarrollada por Alimnova® como T-Monopoly, mantendrá las mismas reglas que el clásico juego de Monopoly®, con excepción de algunas reglas.
* E l sistema tiene interfaz grafica de usuario.
* El sistema tiene persistencia de datos.
* El sistema debe funcionar en las salas de la facultad de ingeniería de la Pontificia Universidad Javeriana.

### 3.3.2 Especificaciones

#### 3.3.2.1 Hardware

#### 3.3.2.2 Software

Las siguientes restricciones de software fueron sacadas por el Germán Morales (director de desarrollo) teniendo en cuenta los recursos brindados en la sala A del primer piso de la facultad de ingeniería y las necesidades básicas del grupo Alimnova® para llevar a cabo el desarrollo de la aplicación T-Monopoly.

#### 3.3.2.3 Desempeño

#### 3.3.2.4 Interfaces con el usuario

Respecto a las interfaces de usuario, tendremos varias para cada actividad que quiere llevar a cabo el jugador.

Es así como para registrarse se le muestra al jugador un pantalla con el espacio correspondiente para ingresar sus datos de perfil y un botón que le habilite la opción de guardar tales datos.

Luego de realizar su registro se le habilita al usuario la pantalla con las fichas disponibles, permitiéndole al jugador escoger su ficha.

Una vez el usuario ha seleccionado su ficha, se le habilita al usuario el tablero de T-Monopoly, con las propiedades y las fichas en la posición inicial, así como también el historial y la información de los demás jugadores.

Finalmente cuando el usuario decide salir de la aplicación tan solo se le deshabilita la vista del tablero, al cerrar la ventana.

Ilustración : Interfaces con el Usurario

#### 3.3.2.5 Restricciones De Diseño

Los aspectos para tener en cuenta para el diseño del programa es tener en cuenta las restricciones dadas por el cliente:

Ilustración 18: Restricciones de Diseño

Respecto a la persistencia Alimnova® manejará archivos planos, en donde va a almacenar la información del jugador, como sus perfiles, las partidas ganadas, las partidas jugadas y las partidas perdidas.

Para poder generar un historial a medida que se desarrolla el juego, el almacenamiento será durante la ejecución, de forma tal que al terminar un turno se almacena la información del jugador y se muestra inmediatamente el resultado del turno terminado (compra de propiedades, pago de dinero por alquiler, ganancia ocasional, etc.).

Generar el historial es diferente al manejo de perfiles de usuario y de partidas, ya que estos últimos si permanecen almacenados y serán cargados cada que comience una partida, para poder verificar que el usuario existe y le permita el ingreso a la aplicación.

Para cumplir con la restricción de tener una interfaz fuerte, Alimnova® desarrollará la interfaz con diferentes herramientas que permitan hacerla más llamativa y agradable al usuario, es por eso que emplearemos flash, para realizarlas, el encargado de dirigir este procesos, será el diseñador Nestor Diazgranados , el cual es conocedor de la herramienta.

Una restricción dada por el cliente acerca de la muestra de la misma es que el programa debe funcionar en la sala A de la facultad de Ingeniería de la Pontificia Universidad Javeriana, por lo cual debemos tener en cuenta que todos los programas que usemos se puedan emplear posteriormente en dicha sala, es decir, que sean compatibles con el sistema operativo que es Windows XP, además que es necesario también realizar pruebas al programa en la sala, con el fin de asegurarnos que realmente funcione y lo haga bien.

Finalmente respecto a la restricción que debe tener arquitectura Cliente-Servidor, tendremos en cuenta que existe un proveedor del juego, para este caso será el servidor, el cual además de contener las funcionalidades del juego, la interfaz del mismo, también permite la conexión de otros usuarios para iniciar una partida, que este caso mínimo deben ser dos, para que se pueda realizar un juego, ya que una persona no puede jugar contra el computador.

El lenguaje de programación que Alimnova® usará para la implementación de T-Monopoly será java, el cual integraremos con nuestra interfaz de usuario creada en flash, para este trabajo contaremos con el apoyo y asesoría de nuestro Director de Desarrollo, Germán Morales, quien será el encargado de dirigir este proceso.

Como herramienta de diseño Alimnova® usará con Enterprise Architect para a realizar diagramas de clase, diagramas de arquitectura; entre otros diagramas y modelos relevantes para el diseño del sistema y el entendimiento del mismo, en esta labor estarán relacionados la arquitecta Andrea Fajardo y el Director de Desarrollo, para generar los modelos necesarios, que permitan el buen desarrollo de la aplicación los cuales posteriormente serán evaluados por David Suarez (Director de calidad y manejo de riesgos ) y Laura Arias (Gerente de Proyecto).

Ilustración 19: Herramientas de diseño

#### 3.3.2.6. Confiabilidad

Con el fin de controlar quien va a ingresar al sistema se tendrá en cuenta la validación del usuario, por medio de su nombre y contraseña, de forma tal que debe estar registrado para poder acceder a la aplicación, de no estarlo tiene la opción de registrarse ó de cerrar la aplicación y no usarla.

Los fallos son inevitables, por lo tanto el sistema debe recuperarse luego de 10 segundos referencia documento requerimientos no funcionales. Confiabilidad , permitiendo de esta forma garantizar que si se genera un fallo, este se solucionará y el jugador podrá continuar con la partida.

Para manejar las versiones de cada documento o parte del código, respecto al proceso de documentación y al de desarrollo, manejaremos SVN Tortoise [[**3.3.2.5 Restricciones De Diseño**](#_3.3.2.5_Restricciones_De)], esta herramienta nos permite mantener un control de los documentos y posteriormente del código desarrollado por cada uno de los miembros de Alimnova®, para así mantener el avance y poder ver versiones anteriores dado el caso que así se requiera.

#### 3.3.2.7 Disponibilidad

La disponibilidad del sistema será durante la sustentación de este en la sala A, aproximadamente dos horas, sin embargo se almacenará en un CD, al cual tendrá acceso el profesor y él un futuro los estudiantes de Ingeniería a de Software de los próximos semestres, serán ellos quién tengan acceso a futuro de este programa y estará disponible para ellos, siempre que lo requieran y soliciten al profesor Miguel Torres (ver requerimiento no funcional disponibilidad).

Es importante también para el desarrollo del juego tener claridad que se requiere que una vez alguien quiera jugar, es decir quien crea la partida (Anfitrión), el sistema va a esperar cierto tiempo a que más jugadores se unan a esta partida, de forma que dado el caso que cumplido el tiempo nadie se una, el anfitrión no puede jugar, ya que es necesario que estén al menos dos personas para jugar.

#### 3.3.2.8 Seguridad

Para este juego, T-Monopoly el manejo de seguridad será encaminado hacia el registro de este usuario, de forma tal que de estar inscrito y almacenado el perfil del jugador, para que pueda acceder a la aplicación. Para el caso del anfitrión, este tiene el permiso de crear la partida, dado el caso que no haya una, de existir ya se volvería un jugador, que se une a una partida, de esta forma garantizamos que sólo exista una partida activa y quien quiera jugar se vea obligado a unirse a esta y no crear otras, que posiblemente generen inconvenientes en el inicio de juego.

Ilustración 20: Seguridad

#### 3.3.2.9 Mantenibilidad

Con el fin de permitir que se puedan realizar cambio y mejoras al programa, Alimnova tendrá en cuenta:

**Documentación del código:** En donde se especificarán cada una de las clases implementadas, su funcionalidad, atributos, métodos, etc. Lo cual permite que cualquier persona, conozca el funcionamiento del sistema y pueda cambiar, mejorar o resolver un problema de la aplicación.

**Documentación del proceso:** En donde se explica de una forma menos técnica y más detallada el sistema, por medio de diagramas y demás especificaciones de requerimientos, etc.

Tener en cuenta **patrones de desarrollo**, con el fin que se puedan realizar modificaciones, sin tener que cambiar todo el programa, sino permitir agregar diferentes funcionalidades, con pequeñas modificaciones. Contribuyendo así a la mejora ya avance de la aplicación.

Ilustración 21: Confiabilidad

#### 3.3.2.10 Portabilidad

Los objetivos típicos de un requisito de portabilidad incluyen:

* Especificar si la aplicación puede ser migrada a otros sistemas operativos, diferente al que se montara la aplicación.
* Especificar bajo qué sistema operativo estará la aplicación.
* Cuál es el lenguaje sobre el cual será implementada la aplicación.

<http://www.opfro.org/index.html?Components/WorkProducts/RequirementsSet/Requirements/AvailabilityRequirements.html~Contents>

Dados los puntos descritos anteriormente se especificaran para T-Monopoly®:

* El sistema operativo sobre el cual se montara la aplicación será Windows XP o posteriores.
* La aplicación no funcionara para sistemas operativos Mac y Linux.
* El lenguaje sobre el cual será implementada la aplicación es Java.

#### 3.3.2.11 Persistencia

La persistencia en T-Monopoly® se manejara con archivos planos y no con bases de datos. Las razones principales son: la aplicación no es tan compleja y grande que requiera bases de datos. Otra razón es que solo se almacenara información no se reconstruirán partidas, por lo tanto es mucho más útil y sencillo el manejo de archivos planos para la adaptación a las necesidades del cliente.

Para la especificación de los requerimientos de persistencia, se tendrán en cuenta los siguientes aspectos:

Ilustración X: Especificaciones de persistencia

#### 3.3.2.12 Comunicación

La comunicación para el desarrollo de la aplicación T-Monopoly® será TCP/IP con el fin de proporcionar una comunicación segura, lo cual ayudaría en la prevención de fallos en la ejecución de T-Monpoly. Adicionalmente se utilizara Java RMI http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp para la comunicación entre el servidor y el o los clientes. Este método permite la invocación de objetos remotos de Java desde otras maquinas virtuales de java. Esta manera de comunicación permitirá que la aplicación este en el servidor y los clientes simplemente llamen los métodos de forma remota.

# 4. Validación de requerimientos

## 4.1. Revisión de conceptos de documentación

Con el fin de tener un documento de calidad, que la información sea la necesaria y correcta la revisión se realizara por medio de CxTemp\_SoftwareRequirementsSpecification. Donde cada miembro en su sección asignada revisara la correspondencia de la información escrita contra la que debería estar. Además de la colaboración de David Suarez, Director de calidad y manejo de riesgos, en cuanto a la revisión del documento. Para esto también se tendrá el diccionario correspondiente y de esta manera el documento sea compatible.

## 4.2. Análisis de Trazabilidad

La trazabilidad es una de las características más importantes que debe tener un requerimiento. Para esto se ha diseñado una plantilla en Excel [[**Trazabilidad de los requerimientos**](#_5.3_Documento_De)] que contiene 4 pestañas:

1. **Estado del requerimiento**

Se refiere a que fase del ciclo de vida del software va dicho requerimiento. Escribiendo la fecha en que se completo dicha fase del ciclo de vida. Existen unos estados para cada requerimiento, entre ellos se encuentran: concepto, desarrollo, integración, pruebas, despliegue e instalación. También se manejaran unas convenciones para determinar el avance de los estados. Las convenciones son: Rojo para incompleto, Amarillo para en proceso, Verde para completo y Azul para cancelado.

Esta plantilla es útil para Alimnova® ya que se puede verificar el estado del requerimiento, de manera que se pueda ver el avance de T-Monopoly® como línea de tiempo.

1. **Origen de los requerimientos funcionales**

Esta plantilla permite determinar de donde surgieron los requerimientos funcionales, es decir, de los casos de uso. Se realizo una matriz con las dos variables, el identificador de los requerimientos verticalmente y el identificador de los casos de uso horizontalmente, y se marca con una X la relación con ambas.

1. **Origen de los requerimientos no funcionales**

Igual que el anterior, pero esta matriz permite determinar el origen de los requerimientos no funcionales. Los cuales provienen de los requerimientos funcionales. El id de los requerimientos no funcionales se ubicaron verticalmente y el id de los requerimientos funcionales de manera horizontal.

1. **En qué fase del ciclo de vida de los requerimientos va cada requerimiento**

Finalmente se debe llevar a cabo cada requerimiento basadoenel modelo espiral para la construcción de requerimientos ya explicado [[**Sección 1.6**](#_1.6_MODELO_DE)], por eso se creó dicha plantilla. Identificando en qué fase del ciclo de vida va y en que iteración. Aplica tanto para los requerimientos funcionales como para los no funcionales.

## 4.3. Evaluación de Requerimientos de Software

Para la evaluación de requerimientos de Software Alimnova® se baso en las listas de chequeo: CxCheck\_NonFunctionalRequeriment y CxCheck\_Requeriments las cuales fueron llevadas a un documento de Excel. Donde se separaron las pestañas por listas de chequeo para requerimientos funcionales y no funcionales.

[[**Ver lista de chequeo**](#_5.4__Listas)]**.**

## 

# 5. Anexos

## 5.1 Encuestas

Se realizaron 3 encuestas entre los integrantes de Alimnova® las cuales fueron:

* Prioridad de los requerimientos: calificando de 1-10 la importancia de cada requerimiento.
* Alcance de los requerimientos: cada integrante de Alimnova® escribió para cada requerimiento si consideraba que se iba a hacer o no.
* Costo de los requerimientos: cada integrante definió en horas cual sería el costo de implementar un requerimiento. Aplica solo para funcionales.

[**Encuestas**].

## 5.2 Especificación De Requerimientos

El documento contiene la plantilla de los requerimientos, al igual que la especificación de cada requerimiento.

[**Documentación de Requerimientos**].

## 5.3 Documento De Trazabilidad

El contenido del documento ya fue explicado en la sección 4.2.

[**Trazabilidad de los requerimientos**].

## 5.4 Listas De Chequeo

Se tuvo en cuenta las listas de chequeo de CxOne para requerimientos funcionales y no funcionales.

[**Ver Lista de Chequeo**].

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