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 subsecciones 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 |
| 1.9.1 | Marzo 26/2010 | Sección 2.2.4, 2.2.2, 3.3.2.1 , 2.1.2 | Adición y revisión de las secciones mencionadas. | William Jiménez, Administrador de Configuraciones y David Suárez, director de Calidad. |
| 2.0.0 | Marzo 26/2010 | Sección REFERENCIAS | Estructuración de la sección de referencias, sin IEEE. | William Jiménez Administrador de configuraciones y documentación |
| 2.1.0 | Marzo 30/2010 | Sección 3, Sección 1 | Restructuración sección 3, Modificación sección 1 | Laura Arias, Director de proyectos |
| 2.1.1 | Marzo 31/2010 | Sección 1.5, 1.6.1, referencias 1 - 9 | Modificación de las secciones, Estándar IEEE para referencias> | Laura Arias, Director de proyectos. |
| 2.1.2 | Abril 2/2010 | Sección 1.6.1 – 1.6.4, referencias 10- 16, sección 4. | Modificación de las secciones mencionadas y Estándar IEEE para las referencias. | Laura Arias, Director de proyectos. |
| 2.1.3 | Abril 5/2010 | Sección 4.2 | Modificación sección | Laura Arias, Director de proyectos. |
| 2.2.3 | Abril 5/2010 | Sección 3.1, 3.1 y modelo del dominio | Corrección y revisión de las secciones 3.3 y 3.1 y corrección del modelo del dominio | Andrea Fajardo , Arquitecta |

Tabla : Historial De Cambios

CONTENIDO

[HISTORIAL DE CAMBIOS 2](#_Toc257749411)

[CONTENIDO 5](#_Toc257749412)

[LISTA DE TABLAS 7](#_Toc257749413)

[LISTA DE ILUSTRACIONES 8](#_Toc257749414)

[1. INTRODUCCIÓN 9](#_Toc257749415)

[1.1 PROPÓSITO 9](#_Toc257749416)

[1.2 ALCANCE 10](#_Toc257749417)

[1.3 DEFINICIONES, ACRÓNIMOS, Y ABREVIACIONES 11](#_Toc257749418)

[1.3.1 Diccionario del Juego 11](#_Toc257749419)

[1.3.2 Diccionario Técnico 12](#_Toc257749421)

[1.4 REFERENCIAS 13](#_Toc257749423)

[1.5 APRECIACIÓN GLOBAL 14](#_Toc257749424)

[1.6 MODELO DE ESPIRAL PARA EL PROCESO DE INGENIERÍA DE REQUERIMIENTOS 17](#_Toc257749425)

[1.6.1 Elicitación de Requerimientos: 17](#_Toc257749426)

[1.6.2 Análisis De Requerimientos 18](#_Toc257749427)

[1.6.3 Especificación De Requerimientos De Software 18](#_Toc257749428)

[1.6.4 Validación De Los Requerimientos 19](#_Toc257749429)

[2. PLAN DE REQUERIMIENTOS 19](#_Toc257749430)

[2.1 Levantamiento de requerimientos 19](#_Toc257749431)

[2.1.1 Fuentes de Requerimientos 19](#_Toc257749432)

[2.1.2. Técnicas de Recolección 22](#_Toc257749434)

[2.2 Análisis de Requerimientos 23](#_Toc257749435)

[2.2.1. Riesgos Generales De Los Requerimientos 23](#_Toc257749436)

[2.2.2. Clasificación De Los Requerimientos 25](#_Toc257749437)

[2.2.3 Modelo Conceptual 31](#_Toc257749443)

[2.2.4 Modelo Del Dominio 33](#_Toc257749445)

[3. Especificación De Requerimientos 35](#_Toc257749454)

[3.1. Documento de Especificación de Requerimientos 35](#_Toc257749455)

[3.2. Requerimientos Funcionales 35](#_Toc257749456)

[3.2.1.Casillas Especiales 35](#_Toc257749457)

[3.2.2 Jugadas Especiales 37](#_Toc257749458)

[3.2.3 Transacciones 37](#_Toc257749459)

[3.2.4 Partida 42](#_Toc257749461)

[3.3 Requerimientos No Funcionales 43](#_Toc257749462)

[3.3.1 Restricciones 43](#_Toc257749463)

[3.3.2 Especificaciones 43](#_Toc257749464)

[3.4 Requerimientos De Negocio 49](#_Toc257749468)

[3.4.1 Persistencia 50](#_Toc257749469)

[3.4.2. Comunicación 50](#_Toc257749471)

[3.4.3. Restricciones De Diseño 51](#_Toc257749472)

[4. Validación de requerimientos 53](#_Toc257749475)

[4.1. Revisión de conceptos de documentación 53](#_Toc257749476)

[4.2. Análisis de Trazabilidad 53](#_Toc257749477)

[4.3. Evaluación de Requerimientos de Software 55](#_Toc257749478)

[5. Anexos 55](#_Toc257749479)

[5.1 Encuestas 55](#_Toc257749480)

[5.2 Especificación De Requerimientos 55](#_Toc257749481)

[5.3 Documento De Trazabilidad 56](#_Toc257749482)

[5.4 Listas De Chequeo 56](#_Toc257749483)

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 propósito del documento SRS es describir de manera detallada el producto T-Monopoly® de forma que el cliente este informado del producto que va a recibir.

Dado lo anterior el SRS pretende describir el producto, donde se incluyen los requerimientos funcionales y no funcionales, al igual que la razón por la cual están dichos requerimientos y la especificación de la clasificación de los mismos. Al igual que las fuentes de donde se tomaron los requerimientos. Todo este proceso fue llevado a cabo, basados en el modelo espiral para el proceso de requerimientos.

## 1.2 Alcance

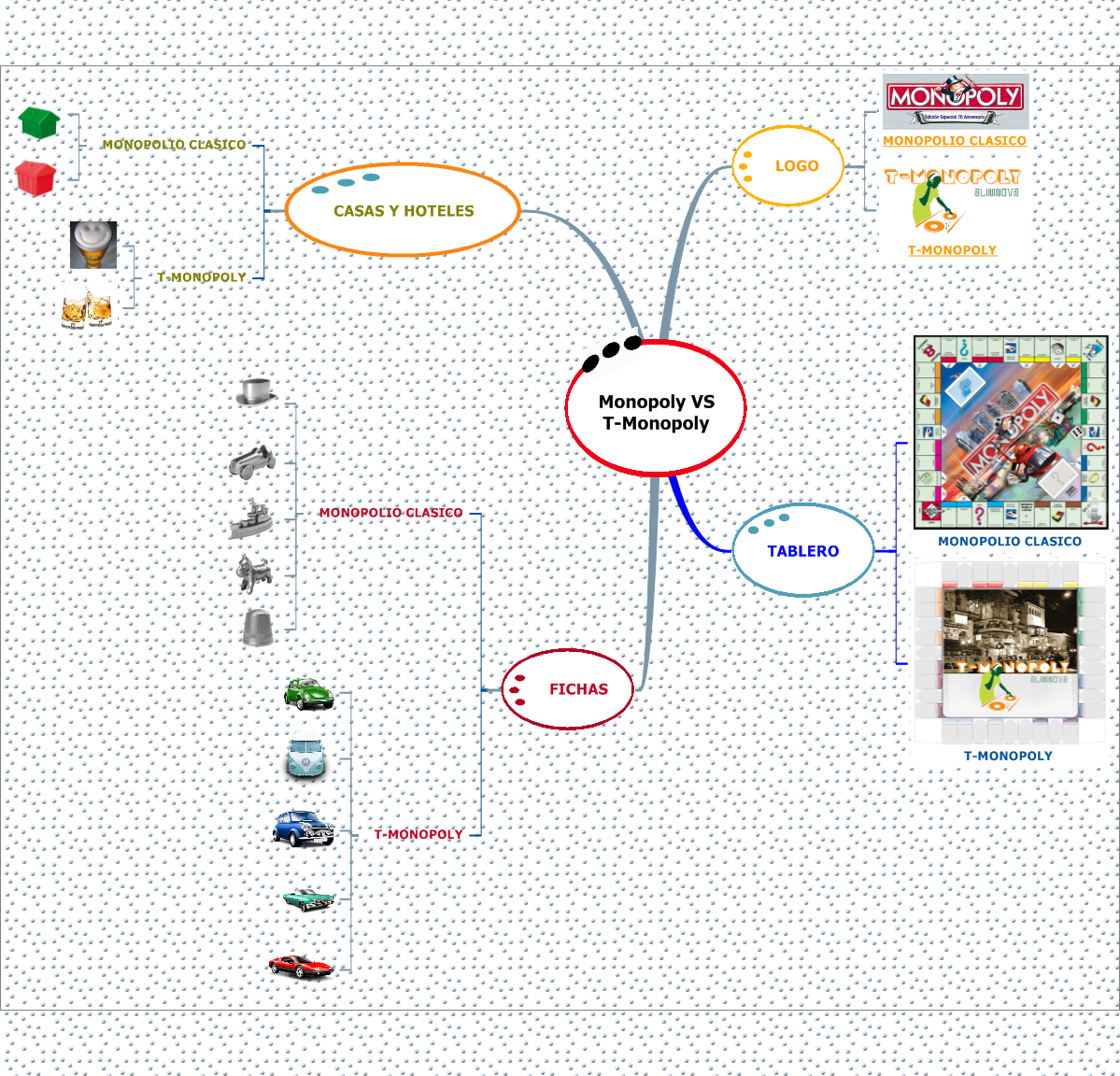
El producto a realizar por Alimnova® será T-Monopoly® el cual es una adaptación del clásico juego Monopolio de Hasbro. El objetivo principal del juego es llegar a ser el jugador más rico comprando, alquilando y vendiendo propiedades. La esencia de T-Monopoly® será las misma, sin embargo, cambiara la temática la cual está relacionada con la zona T de Bogotá, Colombia. En la siguiente figura se muestra la comparación entre ambos monopolios.

Ilustración : Comparación del juego Monopoly® y T-Monopoly. Imágenes Tomadas de ,

## 1.3 Definiciones, Acrónimos, Abreviaciones

### 1.3.1 Diccionario del Juego

|  |  |
| --- | --- |
| LETRA | PALABRA:SIGNIFICADO |
| 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 : Diccionario del juego

### 1.3.2 Diccionario Técnico

|  |  |
| --- | --- |
| LETRA | PALABRA: SIGNIFICADO |
| A |  |
| B |  |
| C |  |
| D |  |
| E |  |
| F |  |
| G | * **GUI:** (Graphic User Interface o Interfaz Gráfica de Usuario). Conjunto de formas y métodos que posibilitan la interacción de un sistema con los usuarios utilizando formas [gráficas](http://www.alegsa.com.ar/Dic/grafico.php) e [imágenes](http://www.alegsa.com.ar/Dic/imagen.php). Con formas gráficas se refiere a botones, [íconos](http://www.alegsa.com.ar/Dic/icono.php), [ventanas](http://www.alegsa.com.ar/Dic/ventana.php), [fuentes](http://www.alegsa.com.ar/Dic/fuente.php), etc. los cuales representan funciones, acciones e información.   http://www.alegsa.com.ar/Dic/gui.php |
| H | * **Halfduplex:** Significa que el método o protocolo de envío de información es bidireccional pero no simultáneo. http://www.babylon.com/definition/half-duplex/Spanish |
| I |  |
| J | * **JAVA RMI**: Java Remote Method Invocation |
| K |  |
| L |  |
| M | * **Modelo del dominio:** |
| N |  |
| O |  |
| P |  |
| Q |  |
| R | * **Requerimiento**: Especificación de lo que debería ser implementado. Son descripciones de cómo el sistema se debería comportar. Ellos podrían ser una restricción del proceso de desarrollo del sistema[17]. |
| S | * **SRS:** Software Requirements Specification. * **Stakeholders:** todas las personas involucradas con el proyecto. |
| T | * **TCP/IP:** Es un conjunto de [protocolos](http://es.kioskea.net/contents/internet/protocol.php3). La sigla TCP/IP significa "**Protocolo de control de transmisión/Protocolo de Internet**". Representa todas las reglas de comunicación para Internet y se basa en la noción de dirección IP, es decir, en la idea de brindar una [dirección IP](http://es.kioskea.net/contents/internet/ip.php3) a cada equipo de la red para poder enrutar paquetes de datos. http://es.kioskea.net/contents/internet/tcpip.php3 |
| U | * **CABLE UTP:** (unshielded twisted pair) par trenzado sin apantallar, es el soporte físico más utilizado en las redes de área local, tanto su costo como el costo de instalación es barato y sencillo. Por él se pueden enviar señales tanto analógicas como digitales. Consiste en un mazo de conductores de cobre (protegido cada conductor por un dieléctrico), que están trenzados de dos en dos para evitar al máximo la diafonía. http://www.mailxmail.com/curso-creacion-cyber/definicion-conceptos-considerados-centrales |
| V |  |
| W |  |
| X |  |
| Y |  |
| Z |  |

Tabla : 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. Imágenes Monopolio Original: <Http://www.worldofmonopoly.Com/Fansite/Index.Php>
2. Imágenes Fichas T-Monopoly: <Http://www.iconspedia.Com/Cat/Cars/>
3. K. Wiegers, *more about software requirements thorny issues and practical advice,* Microsoft press 2006.
4. K. Wiegers, *software requirements,* segunda edición, Microsoft Press 2003.
5. R. Rucker, *software engineering and computer games, Addison Wesly,* December 17/2002.
6. E. Hochmüller, “*Requirements Classification as a first Step to Grasp Quality Requirements*”, Universität Klagenfurt, Austria.
7. S. Lauesen, S*oftware Requirements styles and techniques ,A*ddison Wesley 2002-
8. P. Sawyer, G.kontoya, “S*webok: software requirements Engineering Knowledge Area Descripcion*”,Versión 0.6, United Kingdom.
9. B. Berenbach, D. Paulish, J. Kazmeier, A. Rudorfer, *Software and Systems Requirements Engineering In Practice,* Mc Graw Hill.
10. *Rational Requirements Management with Use Cases,* Versión 5.5, Rational Software.
11. I. Alexander, R. Stevens, *Writing Better Requirements,* Pearson 2002.
12. R. Young, *The Requirements Engineering Handbook,* Artech House 2004.
13. E. Hull, K. Jackson, J. Dick, *Requirements Engineering,* Springer 2005,Second Edition.
14. R. Lutowski, *Software Requirements Encapsulation, Quality and Reuse,* Auerbach. Publications 2005.
15. M. Ramachandran, R. Atem, *Handbook of Research on Software Engineering and Productivity Technologies: Implications of Globalization,* Engineering Science Reference.
16. Construx Software. CxCheck Requirements.
17. I.Sommerville, *Ingeniería de Software 7th ed*, Romo MM. Madrid: Pearson Educación. S.A.2005.
18. P. Zielczynski, *Requirements Management Using IBM Rational RequisitePro,* IBM Press 2008.
19. R. Goldsmith, *Discovering Real Business Requirements for Software Project Success,* Artech House Computing Library 2004.
20. Gotel, Finkelstein, *An Analysis of the Requirements Traceability Problem, Proc. Of 4th International Conference on Requirements Engineering,* 1994.
21. Ramesh, Jarke, *Toward Reference Models for Requirements Traceability, IEEE Transactions on Software Engineering,* Vol. 21, No. 1, January 2001.
22. How Agile Practices Reduce Requirements Risk.
23. Risk and Requirements-Based Testing, James Bach, Indepent Consultant.
24. Exploring Requirements: Quality Before Design (Dorset House, 1989).
25. Lista de chequeo: <http://www.incose.org/tgcc/LIBRARY/CAI%20Checklist%20of%20Risk%20Factors.pdf>
26. Lista de chequeo <http://www.stsc.hill.af.mil/crosstalk/2006/12/0612Young_Tab5.jpg>

1. <http://www.ibmpressbooks.com/articles/article.asp?p=1152528&seqNum=2>
2. IBM,.Requirements.Management.Using.IBM.Rational.RequisitePro.(2007).
3. http://ptgmedia.pearsoncmg.com/images/chap1\_9780321383006/elementLinks/01fig01.gif
4. http://www.ibm.com/developerworks/rational/library/04/r-3217/index.html

1. <http://www.bogota-dc.com/rest/bog-res4.htm>
2. <http://books.google.com.co/books?id=3XSL7kS8GEgC&pg=PA80&dq=REQUIREMENT+PYRAMID+SOFTWARE&ei=tOWaSYWhPLIBPmyyOMK&cd=3#v=onepage&q=REQUIREMENT%20PYRAMID%20SOFTWARE&f=false>.
3. <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>
4. http://books.google.com.co/books?id=HVvwMGGQqpoC&pg=PA145&dq=REQUIREMENT+PYRAMID&ei=mOWaS9XNJpT4zATBs4T9Cg&cd=1#v=onepage&q=REQUIREMENT%20PYRAMID&f=false
5. [*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=Mjn53zPoDQIC&pg=PA212&dq=REQUIREMENT+PYRAMID&ei=mOWaS9XNJpT4zATBs4T9Cg&cd=3#v=onepage&q=REQUIREMENT%20PYRAMID&f=false)
6. [*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)
7. [*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)

1. *[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" \l "v=onepage&q=DIAGRAMAS%20DE%20NECESIDADES&f=false)*
2. http://www.cauvin.biz/RequerimensConcepts.png
3. http://www.scribd.com/doc/2929604/Reglas-MonopolySpanish
4. Redes de computadoras Tanembaum 4ta edición
5. PC hardware Interfaces: A Develoer’s Reference

1. <http://www.adobe.com/es/products/flashplayer/productinfo/systemreqs/>

1. <http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp>

## 1.5 Apreciación Global

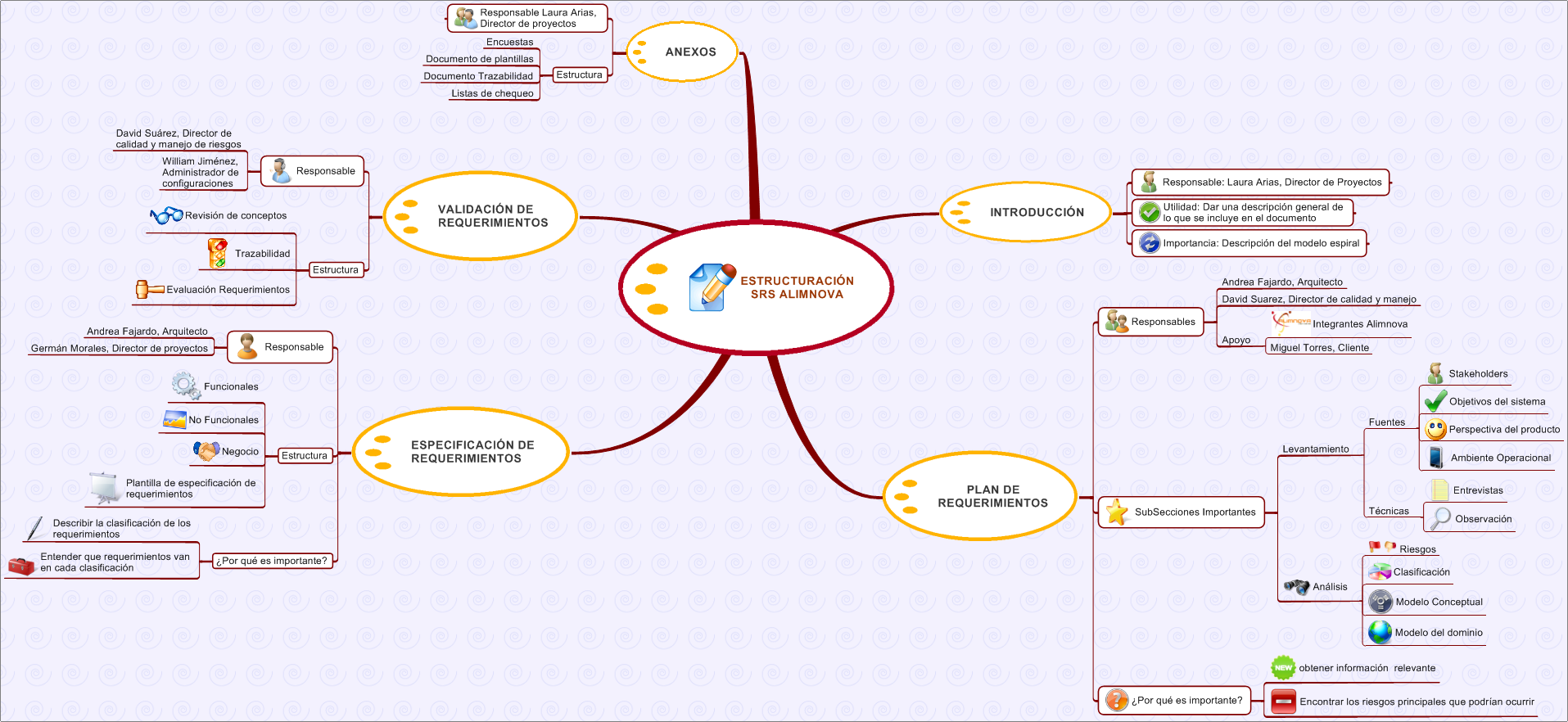
El Documento SRS presentado por Alimnova® está basado en el modelo espiral para el proceso de ingeniería de requerimientos que se explicara más adelante [**Sección 1.6**] y también por el Estándar IEEE Std. 830. La razón principal por la cual se llevaron las modificaciones del SRS original a la propuesta de Alimnova® para el SRS es que es importante para los integrantes de Alimnova® llevar a cabo el modelo espiral durante todo el desarrollo de T-Monopoly®. Otra razón es que se tendrá presente el manejo de riesgos a lo largo del proyecto. En la siguiente ilustración se mostrara la estructuración del documento, incluyendo encargados de cada etapa, las principales etapas, etc.

Ilustración : Apreciación Global del documento SRS

En general la estructuración del SRS está basada en la siguiente ilustración, puesto que son las actividades de ingeniería de requerimientos. Se hizo énfasis en la primera rama de la ingeniería de requerimientos denominada desarrollo de requerimientos. Más adelante se explicaran estas actividades.



Ilustración : Actividades de la ingeniería de requerimientos

La sección 2 corresponde a la primera fase del ciclo de vida espiral para el levantamiento y análisis de requerimientos. El levantamiento incluye las acciones que se llevan a cabo para entender a los usuarios y descubrir sus necesidades . El análisis incluye ir a un nivel de detalle la información recibida de los usuarios distinguiendo los objetivos funcionales de los no funcionales, las reglas de negocio, soluciones, etc. Otros aspectos que se incluyen son las prioridades, riesgos, confiabilidad técnica. El análisis provee una retroalimentación que refina el entendimiento de la actividad de levantamiento .

La sección 3 describe de manera detallada la clasificación de los requerimientos. La razón de ser de esta sección es que el desarrollo del producto de software comienza con una pregunta o solicitud que es el requerimiento y la especificación es la respuesta o solución .La clasificación de los requerimientos de acuerdo al problema a resolver es importante ya que funcionalidad no es la única dimensión relevante en el desarrollo del software, ya que podría incurrir en riesgos del proyecto como la cancelación del mismo .

La sección 4 es la última fase del ciclo de vida de la ingeniería de requerimientos. La pregunta principal sería ¿Cómo asegurar que todo está como debería ser? Que los Stakeholders obtendrán lo que desean, que haya consistencia en lo que se especifico.La respuesta está en verificar y validar los requerimientos. Por esta razón Alimnova® considero incluir la sección de validación de los 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 **4**: modelo espiral para el proceso de ingeniería de requerimientos. Tomado de.

La descripción de cada fase se hará a continuación:

### 1.6.1 Recolección 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. Levantamiento también es el proceso de identificar las necesidades y reducir las diferencias de opiniones entre los Stakeholders, este proceso también involucra tener en cuenta las restricciones impuestas para el proyecto. En pocas palabras el levantamiento es la interacción con los Stakeholders para entender sus necesidades .

Se divide en 2 grupos:

* **Fuentes de los requerimientos:** Objetivos por los cuales se hacen los requerimientos. Planes de negocio. Entre las fuentes se encuentran:

Stakeholders: Identificar quien debería estar involucrado en el sistema, no es un proceso sencillo ya que se deben tener en cuenta todas las personas involucradas. Existen diferentes tipos de Stakeholders cada uno tiene su conjunto de requerimientos y diferentes puntos de vista.

Dominio del problema: se debe tener conocimiento de la aplicación a realizar.

Ambiente operacional: la aplicación en que software se aplicara, cuáles son sus tiempos reales.

Diagrama de casos de uso: Son útiles para entender con el cliente que es lo que el sistema hará, identifica las interfaces que el sistema debe tener. Ayuda a verificar que no se pierdan requerimientos y finalmente verificar que los desarrolladores comprenden los requerimientos .

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

* **Técnicas de levantamiento:** Es necesario seleccionar técnicas apropiadas para obtener los requerimientos dadas las fuentes identificadas.

Entrevistas con los Stakeholders: Es importante tener en cuenta los siguientes pasos para las entrevistas: Introducción, Razón de la entrevista, manejo de tiempos, preguntas a realizar, materiales que usara.

Experiencia como usuario: es útil cuando el proyecto no es tan complejo o peligroso para ser realizado por personas sin experiencia.

Prototipos: Asegurar que los prototipos muestran algunas ideas del producto final al hacer esto se puede estimular al usuario a decir lo que realmente desea.

Un problema que puede ocurrir en esta fase de levantamiento de requerimientos es separar el contexto del requerimiento, ya que los Stakeholders pueden confundir la necesidad de lo que es de fondo . Es decir, un Stakeholder puede decir lo que necesita profundizando en la necesidad pero en realidad es un solo requerimiento.

### 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. Consiste en asegurar que los requerimientos están bien definidos , de manera que se refinen los requerimientos, incluyendo detalles. 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)? .

¿Analizar la factibilidad de los requerimientos? Incluyendo el costo del requerimiento.

Diccionario de datos: incluir todas las definiciones asociadas con el sistema .

¿Cuál es la prioridad del requerimiento? .

El ámbito del requerimiento, es decir, la categoría .

#### 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 Modelo del dominio

**[LIBRO DE GERMÁN]**

### 1.6.3 Especificación De Requerimientos De Software

Esta sección está ligada a la estructura, calidad y verificación del documento de requerimientos. Esta especificación se relaciona con la plantilla SRS la cual provee una estructura consistente para describir los requerimientos, está basada en el estándar IEEE std. 830-1998, esta platilla incluye especificaciones formales que definen requerimientos en lenguajes lógicos y formales. El uso de plantillas para la especificación de los requerimientos también es fundamental, Alimnova® se baso en la plantilla de la referencia página 78, para la especificación de los requerimientos.

### 1.6.4 Validación De Los Requerimientos

Esta sección es importante para Alimnova® puesto que se relaciona con la calidad de los entregables de T-Monopoly®.

Hay 3 criterios para tener en cuenta :

- Utilidad: Consiste en que los requerimientos tienen una razón de ser, criterio de completitud del requerimiento para el buen desarrollo de este . Este criterio está definido en la plantilla de especificación de requerimientos [[**sección 1.6.3**](#_1.6.3_Especificación_De)]. Al igual que la revisión del documento de especificación.

- Trazabilidad: Cada requerimiento está asociado a los artefactos de software y las relaciones entre dichos requerimientos. Este criterio habilita el entendimiento claro de los requerimientos de software, el diseño y la implementación. La trazabilidad de los requerimientos funcionales está ligada a los casos de uso.

- Medición: Verifica la calidad del software. Existen diferentes criterios de medición entre ellos se encuentran: confiabilidad, funcionalidad, usabilidad, administración, Mantenibilidad, etc. . Para este proceso se utilizaran listas de chequeo de CxOne .

# 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

Ilustración **5**: Plan de levantamiento de requerimientos

### 2.1.1 Fuentes de Requerimientos

#### 2.1.1.1Objetivos 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 está 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 **6**: 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 sección 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 1Encuestas Realizadas**](#_5.1_Encuestas)]
  + Alimnova realizara varias encuestas con el fin de recolectar información para la especificación de los requerimientos. Las encuestas serán llenadas por todos los miembros de Alimnova®. De acuerdo a lo anterior se definieron tres encuestas, como se muestra a continuación, donde cada una se llenará para todos los requerimientos.:
* Reuniones entre Stakeholders
  + En las reuniones entre Stakeholders se propondrán nuevos requerimientos a partir de la explicación de cómo entiende cada miembro del grupo Alimnova®, las funcionalidades del sistema. Esto permitirá un mejor entendimiento del problema y adicionalmente saldrán nuevos requerimientos por los diferentes puntos de vista que tiene cada stakeholder.
* Observación
  + se realizara una partida de el juego Monopoly®, tanto de versión para computador como para la de juego de mesa, como una técnica de recolección de requerimientos ya que por medio de esto se genera una observación en general del juego y permite la adaptabilidad a nuestra aplicación T-Monopoly.

## 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.  **[E. Gottesdiener, “How Agile Practices Reduce Requirements Risk”]** | 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. **[E. Gottesdiener, “How Agile Practices Reduce Requirements Risk”]** | ¿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.  **[Referencia Imagen]** | 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. **[Referencia J. Bach, “*Risk and Requirements-Based Testing*”]** | 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. [**Referencia Imagen**] | 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. [**Referencia Imagen**] | 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 : 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.  **[E. Gottesdiener, “How Agile Practices Reduce Requirements Risk”]** | El cliente Miguel Torres, definió únicamente 3 restricciones, estas están aterrizadas. |
| RRN2 | No involucrar al cliente en el proceso de construcción de requerimientos. **[E. Gottesdiener, “How Agile Practices Reduce Requirements Risk”]** | Semanalmente Laura Arias, Director de proyectos tiene reuniones con el cliente con el fin de informar avances o cambios. |

Tabla : 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.

### 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 las necesidades primarias del cliente permite dividir los requerimientos en diferentes tipos así:

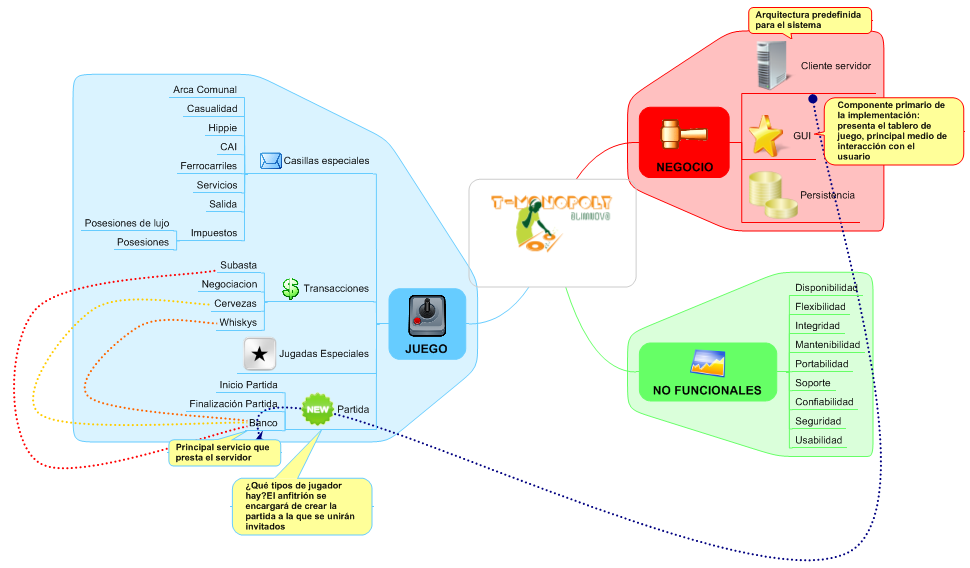


Ilustración **7**: Clasificación de requerimientos y relación de categorías

Alimnova® ha realizado una clasificación de requerimientos de acuerdo a los componentes más globales y al mismo tiempo más relevantes que la organización considera para la implementación del sistema.

Ilustración **8**: Explicación categorías de requerimientos definidas por Alimnova®.

Esta clasificación a su vez se puede ver mapeada en la pirámide de requerimientos la cual nos brinda un primer vistazo de la trazabilidad:

Ilustración **9**: Explicación categorías de la pirámide de requerimientos.

Consultas: , .

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:



Ilustración **10**: Pirámide de requerimientos.

Consultas: .

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 así como su responsable, permitiendo realizar el diseño y posteriormente una implementación consistente.

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 casos de uso que faltantes y sobrantes acorde a las necesidades y características de 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 , , .

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 requeridas por el juego podremos identificar los casos de uso asociados a cada una, los cuales contribuirán 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.

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

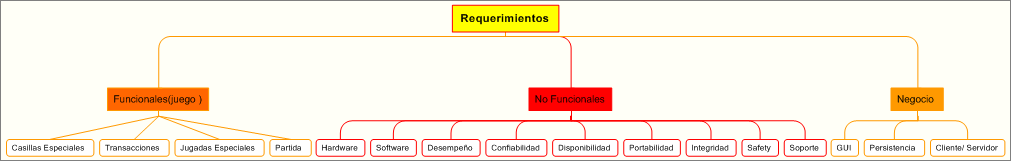


Ilustración **11**: Primera aproximación de la clasificación de requerimientos.

### 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.

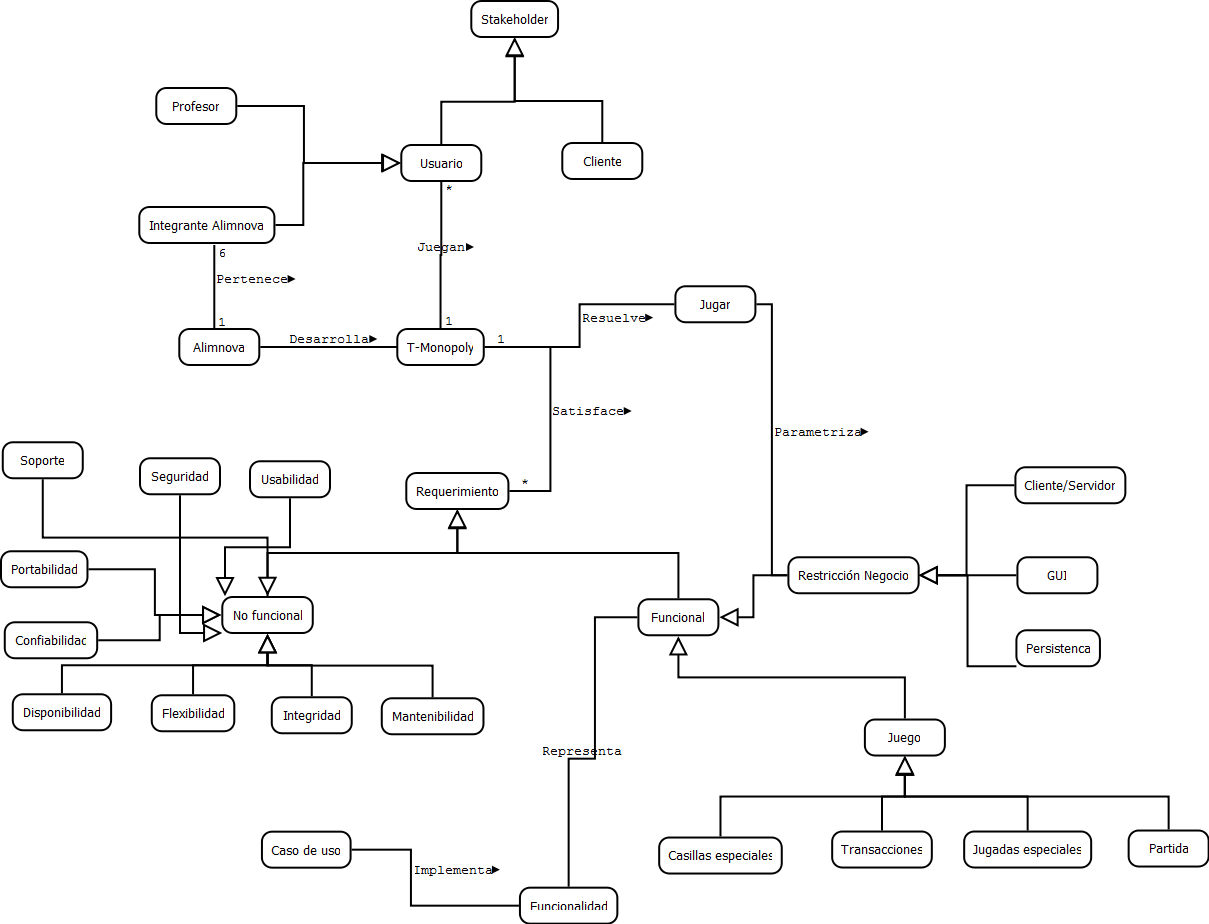


Ilustración **12**: Modelo conceptual para T-Monopoly.

Consultas: .

### 2.2.4 Modelo Del Dominio

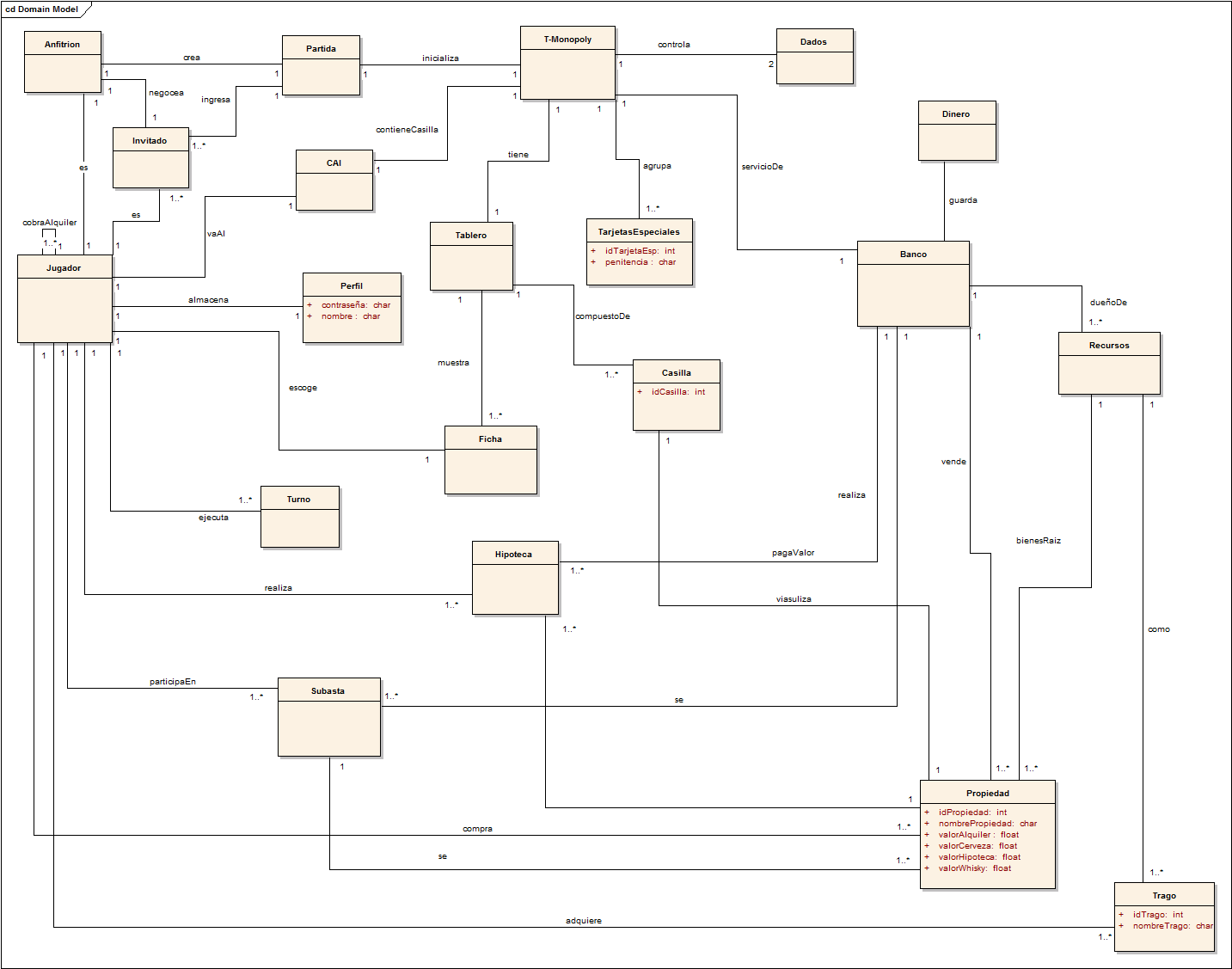
****

Ilustración **13**: Modelo del dominio T-Monopoly.

**JUGADOR**

Representa la persona que va a interactuar con el juego T-Monopoly, el cual puede ser:

* **INVITADO:** Corresponde a la persona que se une a una partida creada
* **ANFITRIÓN:** Corresponde a la persona que crea la partida y accede primero al juego

**PARTIDA**

Corresponde a la actividad que realiza el anfitrión, para poder jugar luego T-Monopoly, lo cual corresponde a escoger la ficha que desee, para luego mostrar el tablero, las fichas, etc.

* **TURNO:** Representa el momento que tiene el jugador para realizar un lanzamiento de dados y realizar diferentes actividades (compra, venta, negociación, etc)

**PERFIL**

Representa los datos ingresados por el jugador que son almacenados en el archivo para posteriores jugadas.

**CAI**

Representa la casilla de la cárcel, a la cual va el jugador bien sea porque al lanzar los dados termina en la casilla de ir al CAI ó porque ha sacado una tarjeta que tenga la penitencia de ir al CAI

**T-MONOPOLY**

Representa la lógica del juego de Monopoly, todo lo relacionado con reglas, componentes propios del juego, etc.

* **TABLERO:** Representa el tablero como tal , que se muestra en pantalla al jugador el cual tiene:
  + **FICHA**: Representa la ficha escogida por el jugador, para ser representado en el tablero de juego.
  + **CASILLA:** Representa las diferentes divisiones que tiene el tablero, en donde cada casilla tiene una propiedad asociada, una casilla de penitencia ó un espacio para tarjetas especiales.
* **TARJETASESPECIALES:** Corresponden a un componente del juego, en donde se presenta en cada tarjeta una determinada situación que debe afrontar el jugador, bien sea que gana dinero, que debe ir a la cárcel, etc.
* **DADOS:** Representan los dados que permiten principalmente moverse a los jugadores, acorde al número obtenido por ellos, aunque también permite realizar cobrar alquileres, entre otros aspectos que pueden ser determinados por los dados.
* **BANCO :** Representa el coordinador del juego ya que es quién dentro del juego, maneja:
  + **DINERO:** Representa el dinero para que sea asignado a los jugadores inicialmente, el dinero para pagar eventuales casos de las tarjetas especiales, pagar el paso por la casilla de inicio, etc.
  + **RECURSOS:** Representa que el banco inicialmente es dueño de los tragos y de las propiedades y que a medida que se desarrolla por medio de transacciones se le asigna a un jugador, bien sea por compra directa o por otras transacciones, entre los recursos están:
    - **TRAGO:** Es el equivalente a las casas y hoteles en el Monopoly, en donde las cervezas son las casa y el whisky representa los hoteles
    - **PROPIEDADES:** Representa las propiedades del Monopoly, para T-Monopoly son bares y restaurantes de la zona T de Bogotá

**SUBASTA:** Representa la subasta que se genera, en el momento que un jugador no quiere o no puede realizar la compra de la propiedad en la que cayó en su turno.

**HIPOTECA:** Representa la hipoteca que se realiza sobre determinada propiedad escogida por un jugador, cuando este no tiene dinero para cancelar un alquiler, impuesto, etc. Ya que por este medio puede pedir dinero prestado al banco, acorde al valor de la hipoteca de esa propiedad.

# 3. Especificación De Requerimientos

## 3.1. Documento de Especificación de Requerimientos

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

## 3.2. Requerimientos 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.

### 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 : 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 : 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®.

#### 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:

Ilustración **14**: Explicación subasta de T-Monopoly.

#### 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.

Ilustración **15**: Imagen Zona T Bogotá, Colombia. Tomada de <http://bogowiki.org/zona%20t>

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 interceptan en forma de T. Está ubicada en frente al centro comercial Andino entre la “Calle del sol” y la calle 82, entre las carreras 12 y 13.

Ilustración **16**: Descripción de las propiedades T-Monopoly.

Consultas:,[22] .

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 el nombre, precio en el mercado, cover, 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.

|  |  |  |  |
| --- | --- | --- | --- |
| 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 : 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 Requerimientos No Funcionales

### 3.3.1 Restricciones

En esta sección se definirán las principales restricciones identificadas por Alimnova®, identificadas a partir de las condiciones dadas por el cliente, de las necesidades del juego, entre otros aspectos que estén relacionados con el desarrollo del proceso y que contribuyan al buen funcionamiento del juego.

A partir de las restricciones impuestas por el cliente, realizamos la división realizada a los requerimientos en este documento, las cuáles son: Arquitectura Cliente/Servidor, Interfaz gráfica fuerte, y Persistencia de datos , además de estas restricciones Alimnova® ha definido:

* Alimnova® en el juego T-Monopoly, mantendrá las mismas reglas básicas del clásico juego de Monopoly®, con variación de algunas reglas.
* El sistema debe funcionar en las salas de la facultad de ingeniería de la Pontificia Universidad Javeriana.
* El lenguaje de desarrollo de T-Monopoly es Java.
* Cada uno de los miembros de Alimnova® tendrá el mismo entorno de desarrollo, el cual es Eclipse, con el fin de facilitar el trabajo y actualización de cada una de las etapas de desarrollo.

### 3.3.2 Especificaciones

#### 3.3.2.1 Hardware

T-Monopoly será una aplicación basada en la arquitectura cliente servidor por lo que utilizará infraestructura de redes como: cables, puertos entre otros. Además de esto la aplicación interactuará con los componentes hardware de PCs, por lo cual es necesario especificar los requerimientos mínimos con los que se deberá contar para la ejecución de esta.

En esta sección se describirán los artefactos y dispositivos hardware a utilizar para las restricciones de redes, interfaz gráfica y persistencia.

3.3.2.1.1. Hardware de Comunicación

Debido a que en el juego T-Monopoly se manejarán turnos, correspondientes a cada jugador, el protocolo usado será halfduplex ya que se envían mensajes en ambas direcciones pero no al mismo tiempo, es decir, uno a la vez, en cada turno el jugador enviará información acerca de su estado, compra de propiedades, entre otras actividades.

Este envió de datos se realizará a través del protocolo TCP/IP. El medio de transmisión serán cables UTP.

Además de esto para el desarrollo y las pruebas preliminares se utilizaran los PCs de las salas de la facultad de Ingeniería de la Pontificia Universidad Javeriana.

3.3.2.1.2. Hardware de GUI

 Para GUI lo fundamental es que la máquina sobre la cual se instalará la aplicación cumpla con las especificaciones mínimas para flash y para el entorno de java.

Además que tenga instalado flash y Java en el momento de usar la aplicación, ya que si no se tienen instalados, la máquina no reconocerá los archivos del programa.

Para que la aplicación se ejecute de manera óptima se proponen los siguientes requisitos:

* Procesador Intel® Pentium® 4 a 2,33 GHz o equivalente
* 128 MB de RAM
* 64 MB de RAM

Consultas: [30].

3.3.2.1.3 Hardware de persistencia

Base de datos IconEn la ejecución de T-Monopoly se generarán archivos que actuarán como salidas del sistema y que luego se emplearán para verificar determinada información de acceso al juego. Un ejemplo claro son los perfiles de los jugadores los cuales serán almacenados en archivos planos, por lo cual no se ocupará mucho espacio de almacenamiento y el manejo será simple y eficaz. Estos archivos permitirán luego evaluar si el jugador tiene acceso al juego, por medio del perfil creado.

Además de almacenar perfiles, también se almacenarán las partidas una vez se hayan terminado, pero estos archivos tan solo serán consultados por el jugador, dado el caso que así lo desee.

Para este ámbito la exigencia es que la máquina donde se instalará la aplicación tenga como mínimo 3Gb de espacio disponible en disco, teniendo también en cuenta el tamaño de la aplicación y de la JVM.

#### 3.3.2.2 Software

Las siguientes restricciones de software fueron sacadas por 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 pantallas, acorde a cada actividad que quiere llevar a cabo el jugador.

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

Dado el caso que el jugador ya tenga su perfil, se le muestra la pantalla de registro, para que ingrese los datos correspondientes a su perfil, de forma que si está almacenado dicho perfil pueda ingresar a la partida, de lo contrario, se le informa que revise la información ingresada o que debe registrase .

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. 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.

#### 3.3.2.6 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.

#### 3.3.2.7 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

#### 3.3.2.8 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

#### 3.3.2.9 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 montará la aplicación.
* Especificar bajo qué sistema operativo estará la aplicación.
* Cuál es el lenguaje sobre el cual será implementada la aplicación.

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

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

Por lo anterior la aplicación no es portable ya que se desarrollará para un sistema operativo únicamente, llevando a que quién quiera usarla debe tenerlo y conocer dicha especificación y saber que debe tener en cuenta el lenguaje en el que se va a desarrollar. Con el fin que tenga un buen funcionamiento en la máquina que se quiera ejecutar y que cumpla con las condiciones anteriores.

## 3.4 Requerimientos De Negocio

### 3.4.1 Persistencia

La persistencia en T-Monopoly® se manejará 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

### 3.4.2. 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-Monopoly. Adicionalmente se utilizara Java RMI [25] 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.

### 3.4.3. Restricciones De Diseño

http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060030/lecciones/Capitulo%205/introduccion.htm

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

Ilustración 18: Restricciones DEL CLIENTE PARA EL 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.

Debido a que el sistema manejará tan solo dos datos para acceder al juego, decidimos usar archivos planos, ya que una base de datos no justifica su empleo para tan solo un nombre y una contraseña, en por esto que por medio de los archivos planos, con un formato definido permitirá evaluar y realizar esta actividad de validación de usuario para acceder al sistema. **[**[**3.4.1 Persistencia**](#_3.4.1_Persistencia)**]**

El formato a manejar en los archivos es:



Este formato aplica tanto para saber cómo realizar la consulta de cada perfil y para poder almacenar el perfil cuando sea creado uno nuevo, esto con el fin de facilitar ambas actividades (almacenamiento, consulta).

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 los perfiles serán consultados por el sistema cada que comience una partida, para poder verificar que el usuario existe y le permita el ingreso a la aplicación, mientras que el historial es generado por cada partida y puede ser consultado por el jugador cuando así lo desee.

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 realizar algunas animaciones, complementando con la librería de java look and feel https://substance.dev.java.net/learn.html, el encargado de dirigir el proceso de desarrollo en flash será el diseñador NestorDiazgranados , el cual es conocedor de la herramienta acompañado del desarrollador Germán Morales el cual está relacionado con la parte correspondiente a Java con la librería Look and feel.

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 **[**[**3.3.2.1.1. Hardware de Comunicación**](#_3.3.2.1.1._Hardware_de)**]** 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 jugadores, 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.

Alimnova® usará 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 Germán Morales, 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

# 4. Validación de requerimientos

La validación asegura que los requerimientos cumplen con las características de la especificación de requerimientos [4]. La validación debe asegurar el crecimiento del SRS en las diferentes iteraciones[4].

La calidad consiste en 2 componentes: ¿Qué Tanto? Cuantas funcionalidades importantes proporciona, y ¿Qué tan bien? Cuál es el nivel de capacidad que proporciona. Muchos requerimientos solo se concentran en el primer componente [19].

Escribir un requerimiento no es suficiente. Se debe asegurar que son correctos y completos para que los procesos de diseño, construcción, pruebas y gestión se lleven a cabo sin entrar en conflicto [4].

## 4.1. Revisión de conceptos de documentación

La revisión de los conceptos de documentación se basa en revisiones que incluyen: Pedirle a un colega que revise el producto de trabajo (revisiones cruzadas) [4], otra importante es que el cliente escriba comentarios del documento. Esta revisión de conceptos también es denominada inspección [4], esta inspección permitirá crear sistemas de calidad [4].

Los pasos de chequeo del documento son los siguientes [11]:

* Estructura del documento: Entender el propósito de la estructura.
* Tener claro que el sistema no está diseñado aún: Asegurar que si se muestran prototipos visuales no implica que estos hayan sido implementados.
* Comparar que lo escrito en cada sección tenga relación con lo que debe ir. Una herramienta de apoyo para este paso será la plantilla 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 de los requerimientos es la habilidad de describir y seguir la vida de un requerimiento, en ambas direcciones adelante y atrás, desde los orígenes, a través del desarrollo y especificación, al igual que el despliegue y uso, a través de periodos de refinamiento e iteración en cualquiera de estas fases[20].

La trazabilidad permite verificar en los requerimientos características como: ¿Cada requerimiento esta únicamente identificado? ¿Cada requerimiento funcional esta trazado por un requerimiento de más alto nivel o de un caso de uso? [4].

La trazabilidad de los requerimientos también está relacionada con la pirámide de requerimientos [[**ver** **ilustración 10**](#_2.2.2._Clasificación_De_1)]**.**

¿Por qué es importante tener trazabilidad en los requerimientos? Es importante para los Stakeholders verificar que el requerimiento está presente. Para los desarrolladores es importante porque se puede tener en cuenta si un requerimiento ha sido implementado o no, o que ha sido completado todo el ciclo de vida para ese requerimiento. La trazabilidad es una herramienta vital en la gestión de desarrollo de sistemas a través de requerimientos[11]. En términos generales la trazabilidad de los requerimientos es una característica del sistema donde los requerimientos están encadenados a las fuentes y artefactos creados durante el desarrollo del ciclo de vida del sistema[21].



Ilustración **25**: Razón de ser de la trazabilidad, a través del tratamiento de requerimientos. Tomado de [11].

Dadas las razones anteriores 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. Semanalmente Germán Morales, Director de desarrollo verificara en qué fase del ciclo de vida va cada requerimiento.

1. **Origen de los requerimientos funcionales**

Esta plantilla permite determinar de donde surgieron los requerimientos funcionales, es decir, de los casos de uso [3]. 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 [18]. 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**

**VAMOS A USARLA ¿??**

Finalmente se debe llevar a cabo cada requerimiento basado en el 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

Es claro que al realizar una especificación de requerimientos estos no estarán perfectos [11]. El propósito de la actividad chequeo de requerimientos es corregir errores en la especificación.

Para la actividad de chequeo cada integrante de Alimnova® se preguntara ¿Se verificara el requerimiento de manera individual o como conjunto de requerimientos? [11], ¿Se verificaran requerimientos funcionales o no funcionales?.

La lista de chequeo para requerimientos funcionales CxCheck\_Requirements contiene ambas formas de verificación. La primera parte que va de REQ-1 a REQ-13 verifica los requerimientos como conjunto, mientras que los que van de REQ-14 a REQ-25 verifica los requerimientos de manera individual.

La lista de chequeo para requerimientos no funcionales es CxCheck\_NonFunctionalRequirement. Esta lista de chequeo también sustenta la trazabilidad de los requerimientos no funcionales [[**sección 4.2**](#_4.2._Análisis_de_1)].

Todas las listas de chequeo 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**](file:///C:\Users\LaUrIta\Desktop\Rep_Alimnova\SRS%20ENTREGA\Encuestas.xlsx)]. Si el hipervínculo no abre favor revisar Documento de Excel denominado 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**](file:///C:\Users\LaUrIta\Desktop\Rep_Alimnova\SRS%20ENTREGA\DOCUMENTACION_REQ(1.5.0)LineaBase.docx)]. Si el hipervínculo no abre favor revisar documento de Word denominado DOCUMENTACION\_REQ(1.5.0).

La lista de los requerimientos funcionales la podrá ver en el siguiente hipervínculo: [[**Descripción Requerimientos Funcionales**](file:///C:\Users\LaUrIta\Desktop\Rep_Alimnova\SRS%20ENTREGA\DescripcionRequeFuncionales.xlsx)]. Si el hipervínculo no abre favor revisar documento de Word denominado DescripcionRequeFuncionales

La lista de los requerimientos no funcionales la podrá ver en el siguiente hipervínculo: [[**Descripción No Requerimientos Funcionales**](file:///C:\Users\LaUrIta\Desktop\Rep_Alimnova\SRS%20ENTREGA\DescripcionRequeNoFuncionales.xlsx)]. Si el hipervínculo no abre favor revisar documento de Word denominado DescripcionRequeNoFuncionales

## 5.3 Documento De Trazabilidad

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

[[**Trazabilidad de los requerimientos**](file:///C:\Users\LaUrIta\Desktop\Rep_Alimnova\SRS%20ENTREGA\TrazabilidadRequerimientos.xlsx)]. Si el hipervínculo no abre favor revisar el documento de Excel denominado TrazabilidadRequerimientos.

[**[Casos de uso asociados](C:\\Users\\LaUrIta\\Desktop\\Rep_Alimnova\\SRS ENTREGA\\ListaCasosUso v (0.0.1).xlsx)**]. Si el hipervínculo no abre favor revisar documentos de Excel denominado ListaCasosUso v (0.0.1)

## 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**](file:///C:\Users\LaUrIta\Desktop\Rep_Alimnova\SRS%20ENTREGA\ChequeoRequerimientos.xlsx)]. Si el hipervínculo no abre favor revisar documento de Excel denominado ChequeoRequerimientos.