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.**3** y modelo del dominio | Corrección y revisión de las secciones 3.3 y 3.1 y corrección del modelo del dominio | Andrea Fajardo, Arquitecto. |
| 2.2.4 | Abril 5/2010 | Sección 4.2 Trazabilidad | Profundización de dicha sección | Laura Arias, Director de proyectos. |
| 2.2.5 | Abril 6/2010 | Referencias | Corrección de las referencias acorde a formato IEEE | Andrea Fajardo, Arquitecta |
| 2.2.6 | Abril 7/2010 | Secciones 2 Plan de requerimientos y 3.2 Especificación de las categorías funcionales de los requerimientos | Correcciones propuestas a las secciones 2 y 3.2 | William Jiménez, Administrador de configuraciones y documentación |
| 2.2.7 | Abril 10/2010 | Sección 2 excepto modelo del dominio | Corrección | Laura Arias, Director de proyectos. |
| 2.2.8 | Abril 11/2010 | Sección 3.2 | Revisión | Laura Arias, Director de proyectos. |
| 2.2.9 | Abril 11/2010 | Sección 3.4 | Corrección | Laura Arias, Director de proyectos. |
| 2.3.0 | Abril 11/2010 | Secciones 1.3.2, 1.4, 2, 3.2. | Adición de definiciones y acrónimos, bibliografía en formato IEEE y corrección de las referencias de las secciones mencionadas. | William Jiménez, Administrador de configuraciones y documentación. |
| 2.4.0 | Abril 11/2010 | Referencias , Sección 3.1, 3.3 y modelo del dominio | Corrección de referencias y las secciones 3.1, 3.3 y modelo del dominio | Andrea Fajardo, Arquitecta |

Tabla : Historial De Cambios

CONTENIDO

[HISTORIAL DE CAMBIOS 2](#_Toc258760086)

[CONTENIDO 6](#_Toc258760087)

[LISTA DE TABLAS 8](#_Toc258760088)

[LISTA DE ILUSTRACIONES 10](#_Toc258760089)

[1. Introducción 11](#_Toc258760090)

[1.1 Propósito 11](#_Toc258760091)

[1.2 Alcance 12](#_Toc258760092)

[1.3 Definiciones, Acrónimos, Abreviaciones 13](#_Toc258760093)

[1.3.1 Diccionario del Juego 13](#_Toc258760094)

[1.3.2 Diccionario Técnico 14](#_Toc258760096)

[1.4 Referencias 15](#_Toc258760097)

[1.5 Apreciación Global 17](#_Toc258760098)

[1.6 Modelo De Espiral Para El Proceso De Ingeniería De Requerimientos 20](#_Toc258760099)

[1.6.1 Recolección de Requerimientos: 20](#_Toc258760100)

[1.6.2 Análisis De Requerimientos 21](#_Toc258760101)

[1.6.3 Especificación De Requerimientos De Software 22](#_Toc258760102)

[1.6.4 Validación De Los Requerimientos 22](#_Toc258760103)

[2. PLAN DE REQUERIMIENTOS 23](#_Toc258760104)

[2.1 Levantamiento de requerimientos 23](#_Toc258760105)

[2.1.1 Fuentes de Requerimientos 23](#_Toc258760106)

[2.1.2. Técnicas de Recolección 28](#_Toc258760107)

[2.2 Análisis de Requerimientos 30](#_Toc258760108)

[2.2.1. Riesgos Generales De Los Requerimientos 31](#_Toc258760109)

[2.2.2. Clasificación De Requerimientos 33](#_Toc258760110)

[2.2.3 Modelo Conceptual 39](#_Toc258760114)

[2.2.4 Modelo Del Dominio 40](#_Toc258760115)

[3. Especificación De Requerimientos 43](#_Toc258760124)

[3.1. Documento de Especificación de Requerimientos 43](#_Toc258760125)

[3.2. Especificación de las Categorías Funcionales de requerimientos 43](#_Toc258760126)

[Requerimientos Funcionales 43](#_Toc258760127)

[3.2.1 Casillas Especiales 43](#_Toc258760128)

[3.2.2 Jugadas Especiales 45](#_Toc258760129)

[3.2.3 Transacciones 46](#_Toc258760130)

[3.2.4 Partida 50](#_Toc258760131)

[3.3 Requerimientos No Funcionales 50](#_Toc258760132)

[3.3.1 Restricciones 51](#_Toc258760133)

[3.3.2 Especificaciones 51](#_Toc258760134)

[3.4 Requerimientos De Negocio 59](#_Toc258760138)

[3.4.1 Persistencia 59](#_Toc258760139)

[3.4.2. Comunicación Y Arquitectura Cliente-Servidor 62](#_Toc258760141)

[3.4.3. Interfaz Gráfica 62](#_Toc258760142)

[4. Validación de requerimientos 62](#_Toc258760143)

[4.1. Revisión de conceptos de documentación 63](#_Toc258760144)

[4.2. Análisis de Trazabilidad 63](#_Toc258760145)

[4.3. Evaluación de Requerimientos de Software 65](#_Toc258760146)

[5. Anexos 66](#_Toc258760147)

[5.1 Encuestas 66](#_Toc258760148)

[5.2 Especificación De Requerimientos 66](#_Toc258760149)

[5.3 Documento De Trazabilidad 67](#_Toc258760150)

[5.4 Listas De Chequeo 67](#_Toc258760151)

[5.5 Propiedades T-Monopoly 67](#_Toc258760152)

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: Comparación del juego Monopoly® y T-Monopoly. Imágenes Tomadas de [1],[2] 11

Ilustración 2: Apreciación Global del documento SRS 18

Ilustración 3: Actividades de la ingeniería de requerimientos 19

Ilustración 4: modelo espiral para el proceso de ingeniería de requerimientos. Tomado de[8]. 20

Ilustración 5: levantamiento de requerimientos 23

Ilustración 6: Stakeholders T-Monopoly 25

Ilustración 7: Funcionalidad del producto 27

Ilustración 8: Caricatura que expresa la desventaja de la implementación de prototipos. Tomada de [31]. 29

Ilustración 9: Un guión gráfico dibujado a mano que utilizan los desarrolladores para elaborar una secuencia de pantallas. Las anotaciones explican las opciones que el usuario realiza o procesos efectuados por el sistema. Tomada de [31]. 30

Ilustración 10: Descripción de encuestas realizadas 31

Ilustración 11: Pirámide de requerimientos . Tomada de [32]. 33

Ilustración 12: Descripción de los niveles de la pirámide de requerimientos. Adaptadas de [32]. 34

Ilustración 13: Clasificación preliminar de requerimientos según la pirámide de requerimientos. 35

Ilustración 14: Clasificación de requerimientos y relación de categorías 37

Ilustración 15: Explicación categorías de requerimientos definidas por Alimnova®. 38

Ilustración 16: Modelo conceptual T-Monopoly®. 40

Ilustración 13: Modelo del dominio T-Monopoly. 41

Ilustración 17: Negociación T-Monopoly®. 47

Ilustración 18: Subasta T-Monopoly®. 48

Ilustración 19: Especificaciones de persistencia 65

Ilustración 20: Formato archivo de perfiles. 66

Ilustración 21: Formato archivo de ranking de jugadores. 66

Ilustración 22: Formato archivo historial del juego. 67

Ilustración 25: Razón de ser de la trazabilidad, a través del tratamiento de requerimientos. Tomado de [11]. 69

Ilustración 26: Visualización de la trazabilidad en la pirámide de requerimientos. Tomado de [22]. 70

# 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

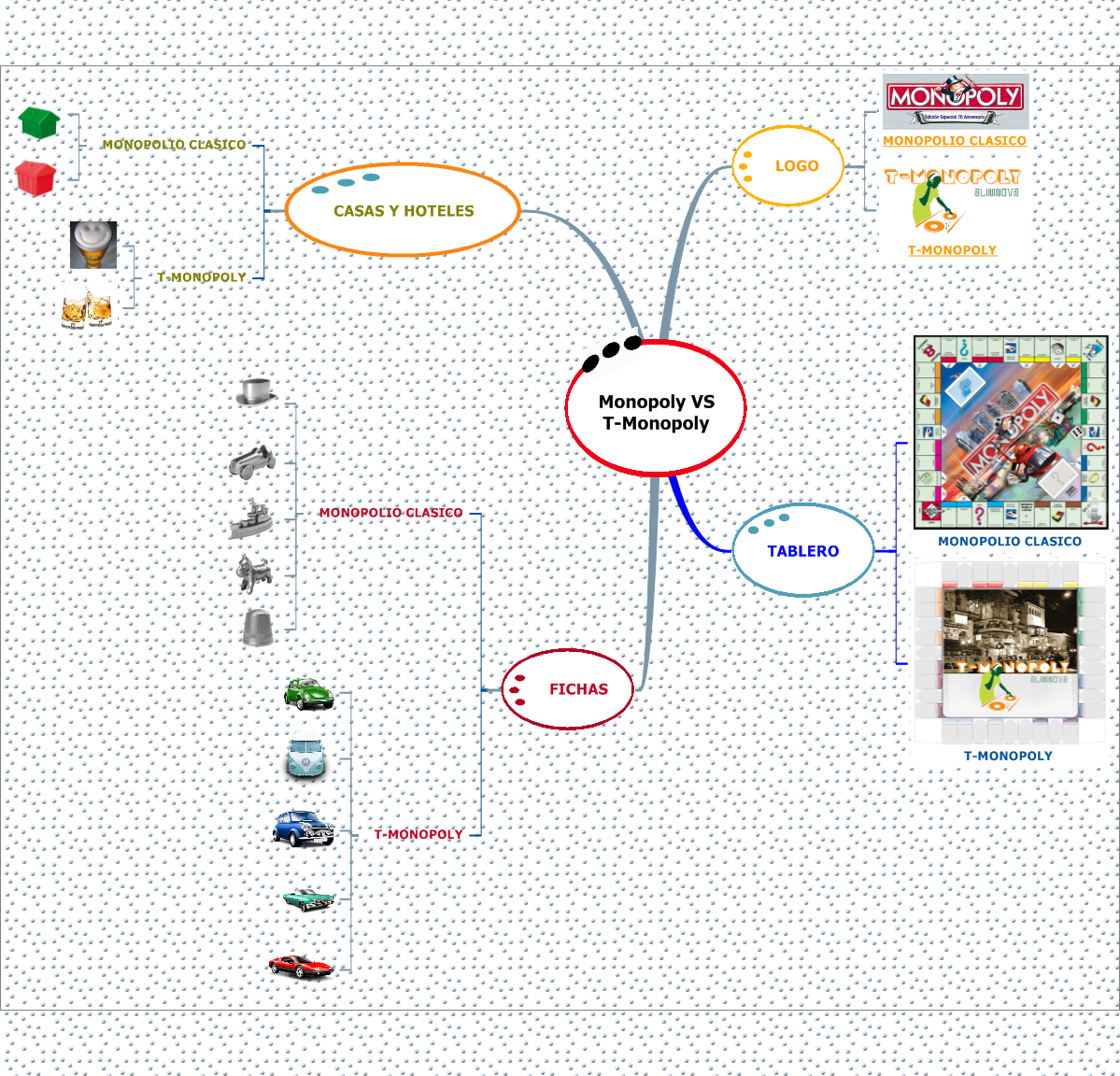
El producto a realizar por Alimnova® será T-Monopoly® el cual es una adaptación del clásico juego Monopolio de Hasbro. El objetivo principal del juego es llegar a ser el jugador más rico comprando, alquilando y vendiendo propiedades. La esencia de T-Monopoly® será las misma, sin embargo, cambiara la 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. * SUELDO: Monto que paga el Banco a los jugadores al pasar por la casilla de salida. |
| 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. [38] |
| H | * **Halfduplex:** Significa que el método o protocolo de envío de información es bidireccional pero no simultáneo. [37] |
| I | * **IBM:** International Bussiness Machine. |
| J | * **JAVA RMI**: Java Remote Method Invocation * **JVM:** Java Virtual Machine. |
| K |  |
| L |  |
| M | * **Modelo del dominio:** |
| N |  |
| O |  |
| P | * **Paper Prototyping**: Paper Prototyping es una variación de las pruebas de usabilidad en donde usuarios representativos desarrollan tareas reales por medio de la interacción con una versión en papel de la interfaz la cual es manipulada por una persona “corriendo en una maquina” [31]. |
| 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. [40] |
| 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. [41] |
| 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. Rational Software Corporation, *Rational RequisitePro User’s Guide,* Versión: 2003.06.00, 1998 – 2003.
23. E. Gottesdiener, “*How Agile Practices Reduce Requirements Risk*”, Inside Analysis.
24. Imagen que contiene riesgos de los requerimientos: <http://images.google.com.co/imgres?imgurl=http://www.stsc.hill.af.mil/crosstalk/2006/12/0612Young_Tab5.jpg&imgrefurl=http://www.stsc.hill.af.mil/crosstalk/2006/12/0612young.html&usg=__b9s3kHCBQz3YhInlDGRTbNcB7RU=&h=729&w=525&sz=106&hl=es&start=21&sig2=2fJKCRLdNsMJdgV6UacY2w&um=1&itbs=1&tbnid=J2u61Hk3D03qxM:&tbnh=141&tbnw=102&prev=/images%3Fq%3Drequirements%2Brisk%26start%3D18%26um%3D1%26hl%3Des%26sa%3DN%26ndsp%3D18%26tbs%3Disch:1&ei=tiS6S6ivOIH48Ab4t81E>
25. J. Bach, “*Risk and Requirements-Based Testing*”, Independent Consultant, IEEE Computer Society 1999.
26. J. P. Flynt, O. Salem, *Software Engineering* *for game developers*, Premier Press, 2005.
27. K. E. Wiegers, *More About Software Requirements Thorny and Practical Advice,* Microsoft Press, 2006.
28. IEEE std 830-1998, IEEE Recomended Practice for Software Requirements Specifications.
29. B. Appleton's Documents, "A Software Design Specification Template"; <http://www.cmcrossroads.com/bradapp/docs/>.
30. K. E. Kendall y J. E. Kendall, *ANÁLISIS Y DISEÑO DE SISTEMAS*, Pearson education, 2005.
31. C. Snyder, *PAPER PROTOTYPING The Fast and the Easy way to Design*, Elsevier, 2003.
32. P. Zielczynski, *Requirements Management Using IBM Rational Requisite Pro*, IBM Press, 2008.
33. P. F. Tiako, *Software Applications: Concepts, Methodologies, Tools, and Applications*, Information Science Reference, 2009.
34. M. Genero, M Piattini y C. Calero, *METRICS FOR SOFTWARE CONCEPTUAK MODELS*, Imperial College Press, 2006.
35. Hasbro, “MONOPOLY Parker Brother´s Real Estate Trading Game”, <http://www.hasbro.com/common/instruct/Monopoly(Spanish).pdf>.
36. BogoWiki, "Zona T"; <http://bogowiki.org/zona%20t>.
37. Wikipedia, "Half Duplex"; es.wikipedia.org/wiki/Half-duplex.
38. Definición de GUI, Diccionario de Informática [En línea]. Disponible: <http://www.alegsa.com.ar/dic/gui.php>. [Consultado el 4 de Marzo de 2010]
39. Half-duplex, Babylon [En línea]. Disponible: <http://www.babylon.com/definition/half-duplex/Spanish>. [Consultado el 4 de Marzo de 2010]
40. TCP/IP, Kioskea [En línea]. Disponible: <http://es.kioskea.net/contents/internet/tcpip.php3> . [Consultado el 4 de Marzo de 2010]
41. Francys Castillo. Creación de un Cyber, Definicón de conceptos considerados centrales. [En línea]. Disponible: <http://www.mailxmail.com/curso-creacion-cyber/definicion-conceptos-considerados-centrales>. [Consultado el 4 Marzo de 2010]
42. Universidad Nacional Sede Manizales. Dirección Nacional de Servicios Académicos Virtuales. Introducción al diseño. [En línea]. Disponible: <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060030/lecciones/Capitulo%205/introduccion.htm>
43. Discovering real business requirements for software project success. Robin F. Goldsmith. Artec House, Inc 2004.
44. Software requirements Styles and techniques. Soren Lauesen. Pearson Education Limited 2002. First published in Great Britain in 2002
45. *Anne Marie Smith,**Business Requirements Gathering - An Overview,* January 1, 2000.
46. Información de la librería Look and Feel de java, <https://substance.dev.java.net/learn.html>

## 1.5 Apreciación Global

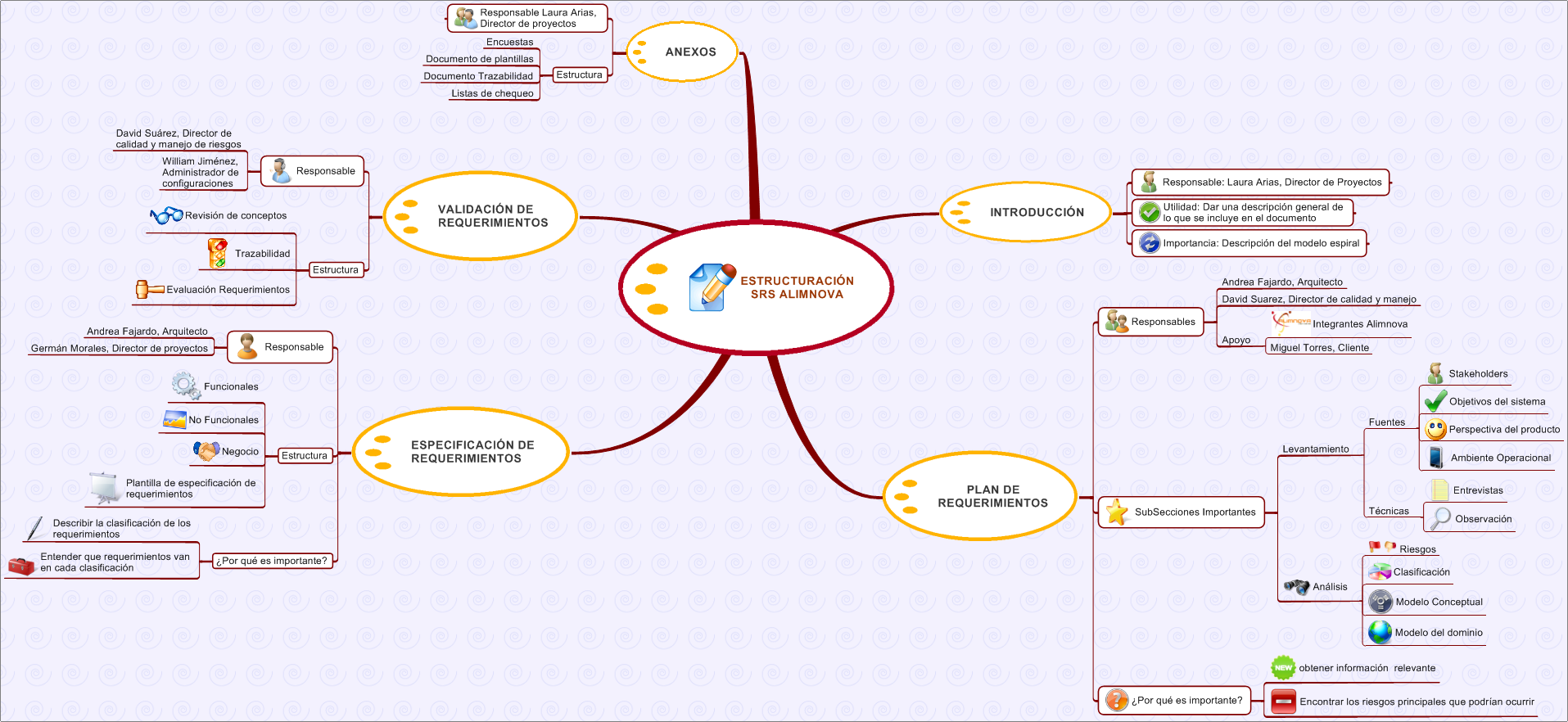
El Documento SRS presentado por Alimnova® está basado en el modelo espiral para el proceso de ingeniería de requerimientos que se explicara más adelante [[**Sección 1.6**](#_1.6_MODELO_DE)] y también por el Estándar IEEE Std. 830. La razón principal por la cual se llevaron las modificaciones del SRS 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

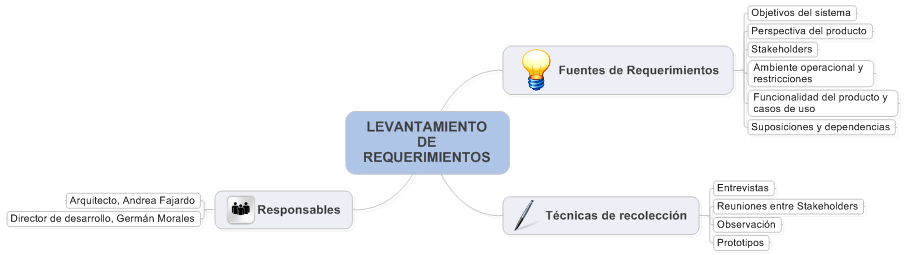


Ilustración **5**: levantamiento de requerimientos

### 2.1.1 Fuentes de Requerimientos

#### 2.1.1.1Objetivos del sistema

Mediante la definición de los objetivos del sistema Alimnova® se asegura de llenar las expectativas que la organización tiene con el producto final. Mediante esta definición de objetivos Alimnova® se asegura de una identificación clara de requerimientos globales y de algunas especificaciones concretas establecidas por el cliente.

Un ejemplo claro en este contexto es la especificación de comunicación que el cliente requiere. Es claro que T-Monopoly® será una aplicación multi jugador, este es un objetivo especifico del sistema el cual abre paso al descubrimiento de requerimientos funcionales y no funcionales relacionados con la comunicación.

Con este proceso Alimnova® asegura una fuente confiable y verificable de requerimientos pues se centra específicamente en el sistema a desarrollar.

#### 2.1.1.2 Perspectiva del producto

Esta fuente de requerimientos implica un amplio conocimiento de la aplicación (T-Monopoly®) a realizar. Para Alimnova® es muy importante este aspecto en la medida en que la aplicación está basada en un juego existente, por esta razón las reglas de Monopoly® son un gran aporte al descubrimiento y recolección de requerimientos sin que esto quiera decir que Alimnova® se apegará completamente a dichas reglas. Según la temática de T-Monopoly®, Alimnova® modificara y adaptara las reglas de Monopoly a la aplicación con el fin de que no se pierda la esencia del juego. Adicionalmente esto permite que se tenga una visión mucho más clara de lo que la aplicación debe hacer.

El proceso desarrollado por Alimnova® para cumplir con el pleno entendimiento de la aplicación fue una serie de reuniones con el fin de que todos los integrantes jugaran varias partidas del Monopoly de Hasbro, el juego de mesa a pesar de que todos los jugadores ya conocían el juego y lo habían jugado anteriormente lo que cuenta como una fuente adicional correspondiente a experiencias previas [26]. 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 el argumento del juego.

Además los integrantes de Alimnova® hicieron varias reuniones con el fin de jugar la aplicación Monopoly también de Hasbro esto con el fin de recolectar ideas para la interfaz grafica de Usuario (GUI), casos de uso 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

Tal vez una de las fuentes de información más valiosas para el levantamiento de requerimientos es el punto de vista del jugador. El equipo de diseño y de levantamiento de requerimientos de Alimnova® representado por el Arquitecto, Andrea Fajardo, se ha cuestionado sobre la funcionalidad del juego desde el punto de vista del mercado del jugador. La razón por la cual se ha prestado especial atención a este proceso es porque no hay nadie que tenga una perspectiva de cómo un juego se debe desempeñar y sentir un juego. Además de esto Alimnova® se propuso desde un principio desarrollar prototipos con el fin de que el usuario pueda probar la aplicación [26]. Para desarrollar esta estrategia Alimnova® intentará evitar el uso de “usuarios sustitutos” como por ejemplo expertos en el juego, desarrolladores de software o simulaciones de usuario. Para esto se propuso que cada miembro de jugara al menos dos partidas de Monopoly con personas ajenas al juego como familiares y amigos con el fin de acercar al máximo la VOC, por sus siglas en ingles VOICE OF THE CUSTOMER (voz del usuario) al EOD, EAR OF DEVELOPER (Oído del desarrollador) [27].

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 la aplicación. Cada uno de los miembros de Alimnova® tiene un rol asignado y con este sus responsabilidades con el grupo y con sí mismo.

Otra persona que estará relacionada con el proceso, será el cliente, Miguel Torres, quien es la guía principal de Alimnova® y será el encargado de asesorarnos en el proceso de documentación y el de desarrollo de la aplicación.

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 tanto técnicos como a nivel de diseño, además también serán considerados usuarios del producto final.

Ilustración **6**: Stakeholders T-Monopoly

#### 2.1.1.4 Ambiente Operacional y restricciones

El producto a realizar por Alimnova® posee unas precondiciones y/o restricciones establecidas por el cliente las cuales influirán en gran medida en el ambiente operacional de la aplicación y finalmente en la implementación de la misma.

Una de las condiciones dadas de antemano por el cliente corresponde a la instalación y sustentación de T-Monopoly® las cuales deberán realizarse en la sala A de la facultad de Ingeniería, por lo tanto es necesaria la compatibilidad entre el software dispuesto por la sala y el producto final. Para efectos de esta compatibilidad el equipo de desarrollo de Alimnova® encabezado por Germán Morales, Director de desarrollo, realizará pruebas del sistema, tanto parcial como en su totalidad, en las instalaciones de la sala A de la facultad de ingeniería con el fin de tener un grupo de requerimientos concreto de acuerdo al hardware en el que la aplicación se instalará.

También es importante tener en cuenta que existen restricciones dadas por el cliente que son igual de importantes para la identificación, análisis de requerimientos y desarrollo de la aplicación. Estas restricciones [[ver sección 3.3.1](#_3.3.1_Restricciones)] hacen parte del ambiente operacional ya que es el medio en que se va a desarrollar la aplicación. La más importante consiste en la modalidad multijugador del juego que exige un desarrollo según los lineamientos de una arquitectura cliente/servidor. Esto implica que el desarrollo de la aplicación estará también inmerso en una infraestructura de redes brindada también por la facultad de ingeniería [[ver sección 3.3.2.1](#_3.3.2.1_Hardware)].

#### 2.1.1.5 Funcionalidad del producto y Casos de uso

Con el fin de que Miguel Torres, cliente, tenga una visión global de esta fuente primaria de requerimientos se expondrá a grandes rasgos de lo que el producto significa en el contexto del usuario.

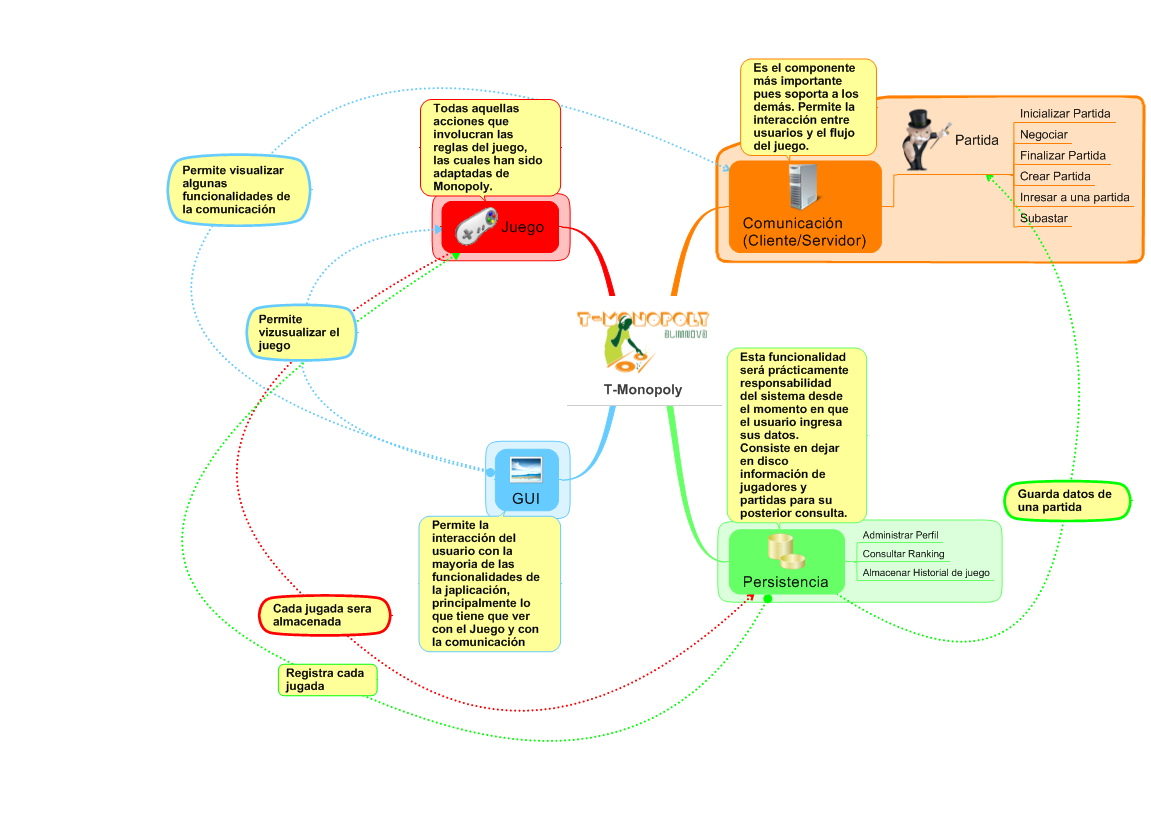


Ilustración **7**: Funcionalidad del producto

Los casos de uso definidos por Alimnova® en el primer hito del proyecto (SPMP) junto a las debidas correcciones realizadas durante este segundo hito son la principal guía para el levantamiento de requerimientos. Con esto se quiere indicar que el surgimiento de los requerimientos funcionales consistirá en un análisis detallado de las funcionalidades que deben caracterizar al producto final, información provista por la última versión de los casos de uso elaborada en el actual hito del proyecto. [Remítase al SPMP].

Para garantizar que los requerimientos provienen de los casos de uso, esta matriz esta especificada en la plantilla de trazabilidad [[**sección 4.2**](#_4.2._Análisis_de_1)].

#### 2.1.1.6 Suposiciones y dependencias

Como ya se había indicado la instalación y ejecución de T-Monopoly® se realizará en la Sala A de la facultad de ingeniería, por lo que las dependencias del producto se limitan únicamente al software y al hardware que se encuentre disponible en ese lugar durante el día de la presentación. Desde esta perspectiva, esta es una valiosa fuente de requerimientos ya que complementa a las demás y este la manera en que las complementa se podrá evidenciar con la estructuración de requerimientos no funcionales asociados a los funcionales extraídos de las demás fuentes.

A nivel de software es claro para Alimnova® que el producto será instalado bajo el sistema operativo Windows XP el cual se encuentra disponible en los PCs de la sala A [[ver sección 3.3.2.2](#_3.3.2.2_Software)]. En cuanto a hardware también sabemos que tenemos que atenernos a lo dispuesto por las maquinas de la sala A, para esto se establecen ciertos requisitos mínimos en este ámbito [[ver sección 3.3.2.1](#_3.3.2.1_Hardware)].

Otro aspecto a tener en cuenta es la aclaración que de antemano hizo el cliente acerca de las variaciones de la aplicación. Desde el principio se estableció que la aplicación y en general el proyecto no tendría ninguna variación durante el desarrollo del ciclo de vida en cuanto a las restricciones y condiciones impuestas por el cliente así como el argumento de la aplicación misma [28], [29].

### 2.1.2. Técnicas de Recolección

La sección anterior está íntimamente relacionada con esta sección puesto que la primera describe lo qué se va a tener en cuenta a la hora del levantamiento de requerimientos y las técnicas de recolección responden al cómo se va a enfrentar este levantamiento de requerimientos. A nivel de roles de Alimnova® las personas que estarán más involucradas en esta etapa del ciclo de vida del hito serán Andrea Fajardo, Arquitecto, y Germán Morales, Director de desarrollo puesto que son los directamente implicados en los procesos posteriores de diseño e implementación aunque contaran con la debida colaboración de los demás miembros de Alimnova®.

* Entrevistas
* Antes de definir un tipo de entrevista formal para los potenciales usuarios de T-Monopoly® los miembros de Alimnova® nos propusimos una “autoentrevista” con el fin de definir el rumbo que queríamos darle a este mecanismo de recolección de información, el resultado fue la decisión de realizar entrevistas de manera informal y al interior del ambiente del juego.

Durante este hito se ha desarrollado un proceso de entrevistas informales a varios de los Stakeholders. Estas entrevistas se han realizado durante partidas de Monopoly (de mesa y el videojuego), para de esta manera buscar la máxima aproximación al punto de vista de un usuario [30].

* Reuniones entre Stakeholders
  + En las reuniones entre Stakeholders, principalmente entre los miembros de Alimnova®, se expondrán nuevos requerimientos a partir de la visión de cada miembro del grupo Alimnova® sobre las funcionalidades del sistema.
* Observación
  + Se realizará una partida de Monopoly®, tanto de versión para computador como para la de juego de mesa, en la cual un miembro de Alimnova® actuará como observador con el fin de extraer requerimientos que sean difíciles de identificar para un jugador como. Esta técnica facilita la adaptación de Monopoly a T-Monopoly®.
* Prototipos
* Desde un principio Alimnova® planeó la implementación de prototipos con el fin de presentar al cliente distintas aproximaciones al producto solicitado, además cumple uno de las características del ciclo de vida en espiral seleccionado por Alimnova®. La idea de los prototipos es la manifestación de las reacciones del cliente sobre el proceso de desarrollo del sistema esto con el fin de identificar nuevos requerimientos y de realizar los cambios necesarios a tiempo y sin mayores dificultades. El tipo de prototipos que Alimnova® se propone desarrollar es un prototipo de características seleccionadas el cual consiste en una implementación con funcionalidades esenciales para el producto final pero que no cuenta en completitud con dichas funcionalidades. Este prototipo también tendrá características de un prototipo corregido pues las funcionalidades a desarrollar no estarán sujetas a requerimientos de desempeño [30].



Ilustración **8**: Caricatura que expresa la desventaja de la implementación de prototipos. Tomada de [31].

La metodología a seguir para la implementación de este prototipo será el Paper Prototyping que consiste en una lluvia de ideas encaminada hacia el diseño, creación, pruebas y comunicación de interfaces de usuario [31].

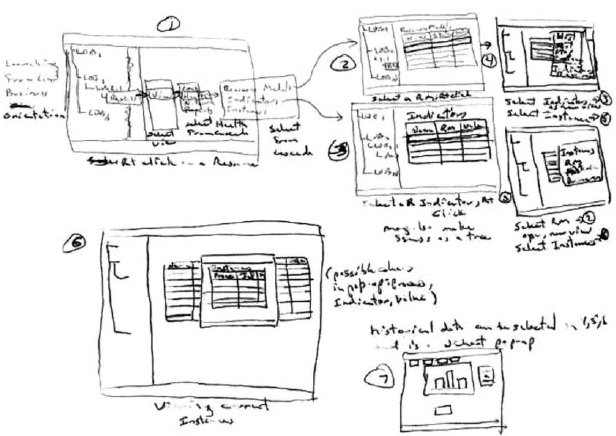


Ilustración **9**: Un guión gráfico dibujado a mano que utilizan los desarrolladores para elaborar una secuencia de pantallas. Las anotaciones explican las opciones que el usuario realiza o procesos efectuados por el sistema. Tomada de [31].

## 2.2 Análisis de Requerimientos

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

Con el fin de establecer un marco de evaluación y análisis para los requerimientos se ha diseñado un conjunto de encuestas con el fin de recolectar información para la especificación de los requerimientos [30]. Las encuestas serán diligenciadas 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.:

***Ilustración 10: Descripción de encuestas realizadas***

### 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[23]. | 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[23]. | ¿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 [24]. | 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[25]. | 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[24]. | 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[24]. | 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[23]. | El cliente Miguel Torres, definió únicamente 3 restricciones, estas están aterrizadas. |
| RRN2 | No involucrar al cliente en el proceso de construcción de requerimientos[23]. | 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 Requerimientos

Las primeras experiencias en cuanto al levantamiento de requerimientos por parte de los miembros de Alimnova® conducían a una clasificación trivial de estos: Funcionales y no funcionales. Por esta razón se acogió, en primera, instancia un modelo piramidal que provee características valiosas para una asimilación rápida de los posibles niveles de requerimientos a desarrollar.

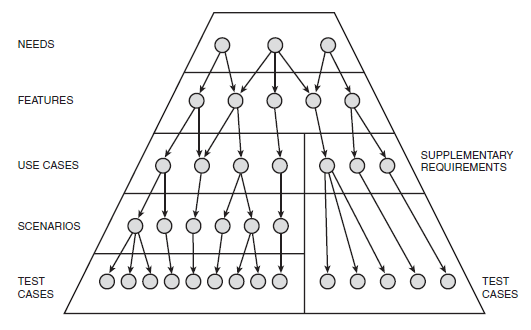


Ilustración ***11***: Pirámide de requerimientos . Tomada de [32].

Esta representación nos brinda un primer bosquejo de una categorización de requerimientos según las características inherentes de los mismos. Las relaciones de la pirámide entre los distintos nodos representan derivación, es decir que por ejemplo las características que el sistema debe tener generan los casos de uso.

A continuación se presenta la descripción de cada nivel.

Ilustración ***12***: Descripción de los niveles de la pirámide de requerimientos. Adaptadas de [32].

A partir del análisis de esta perspectiva el resultado de la clasificación preliminar fue el siguiente:

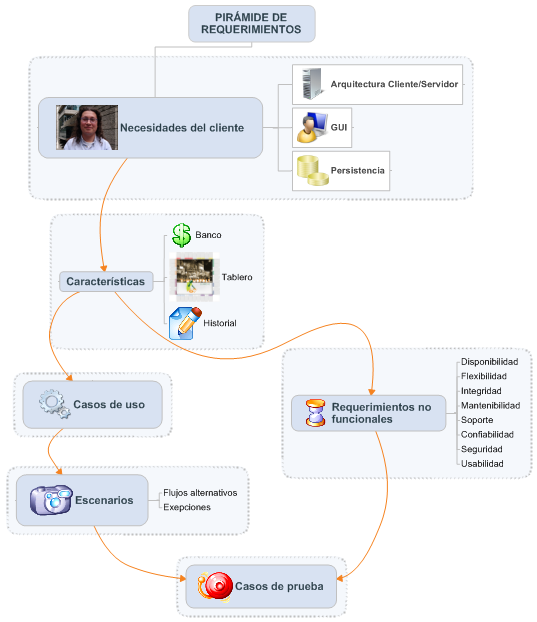


Ilustración ***13***: Clasificación preliminar de requerimientos según la pirámide de requerimientos.

El principal aporte que nos brinda esta clasificación es una impresión global de la trazabilidad [[**sección 4.2**](#_4.2._Análisis_de_1)] de requerimientos. A pesar de esto esta estructura jerárquica es limitada puesto que omite posibles relaciones entre requerimientos de distintos niveles, por ejemplo no presenta una relación entre los requerimientos funcionales derivados de los casos de uso y los no funcionales. Por las razones anteriormente expuestas Alimnova® y en particular su departamento de desarrollo liderado por el Arquitecto, Andrea Fajardo, llego a la conclusión de la necesidad de una clasificación que aportara al diseño y finalmente a la implementación del sistema lo cual también abriría nuevos horizontes en el ámbito de la trazabilidad de requerimientos. De esta manera se procedió a realizar una clasificación más flexible basada en componentes, globales y específicos del juego, que permita identificar relaciones entre los mismos.

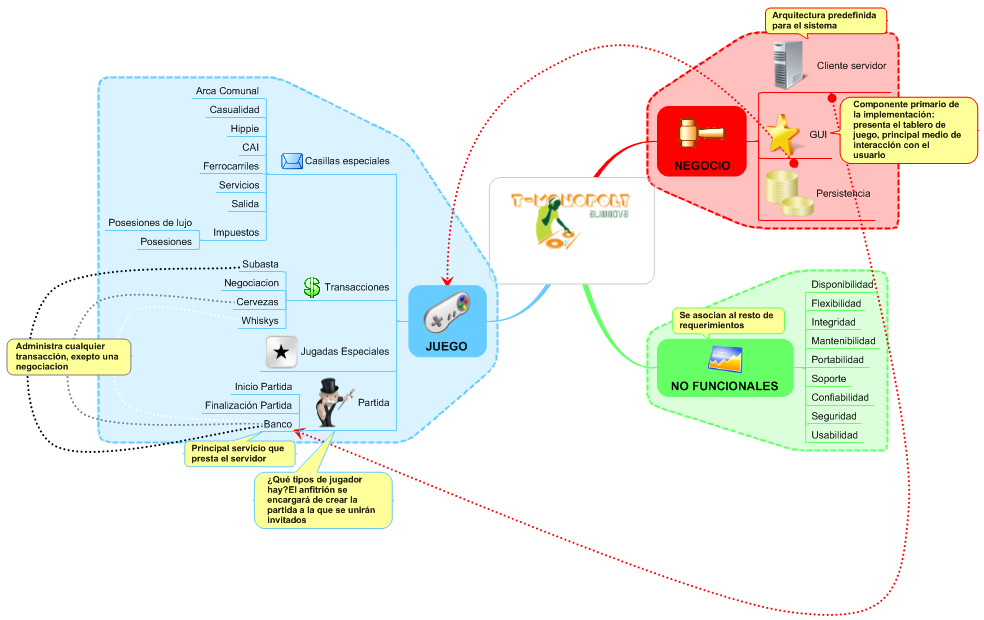


Ilustración ***14***: Clasificación de requerimientos y relación de categorías

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 ***15***: Explicación categorías de requerimientos definidas por Alimnova®.

La esencia de la nueva clasificación radica en los componentes del juego que consideramos pertinentes y necesarios para el entendimiento y desarrollo del juego acorde a las solicitudes del cliente.

**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.

**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.

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

Esta última clasificación abre el panorama para los posteriores hitos de diseño e implementación ya que se realizó pensando ya en un bosquejo de las componentes, clases y paquetes a utilizar en las mencionadas etapas así como una primera versión de sus nombres y relaciones [26].

### 2.2.3 Modelo Conceptual

Un buen entendimiento del dominio de la aplicación es necesaria para el desarrollo de requerimientos del sistema. Este entendimiento puede ser facilitado con el uso de un modelo conceptual. El modelado conceptual es la “actividad de descripción formal de algunos aspectos del mundo físico y contextual del problema a entender” [33]. Es posible que el término sea confuso pues existen muchos tipos de modelos conceptuales (modelos del dominio, modelo entidad relación, diagrama de clases, diagrama de flujo de datos entre los más comunes). Por esta razón el modelo conceptual que a esta sección corresponde se denomina modelo conceptual genérico el cual será el punto de referencia para los demás modelos que competan para el desarrollo de T-Monopoly®.

El modelo conceptual genérico se encarga de expresar una perspectiva global y preliminar del entendimiento del problema por parte de quienes se encargaran de solucionarlo, en este caso Alimnova®. El modelo conceptual identifica los aspectos más relevantes del dominio del problema sin tener que filtrar una porción de dicho dominio, cosa que hacen los demás modelos. Precisamente lo que el modelo conceptual busca es lo contrario, específicamente con este modelo pretende ser un punto de partida para la elaboración del modelo del dominio.

La herramienta utilizada para el modelo conceptual genérico será un mapa conceptual. La razón radica en la minimización de la formalidad (que será añadida en los otros modelos) y la carencia de una semántica específica.

De por si el mapa conceptual es ambiguo. La razón radica en las distintas posibilidades de interpretación que presenta a los analistas en el momento de concretar los demás modelos. La ambigüedad del modelo conceptual permite la categorización al final del análisis del problema, momento en el que toda la información relevante está disponible [34].

A continuación se presenta el modelo conceptual genérico resultante.

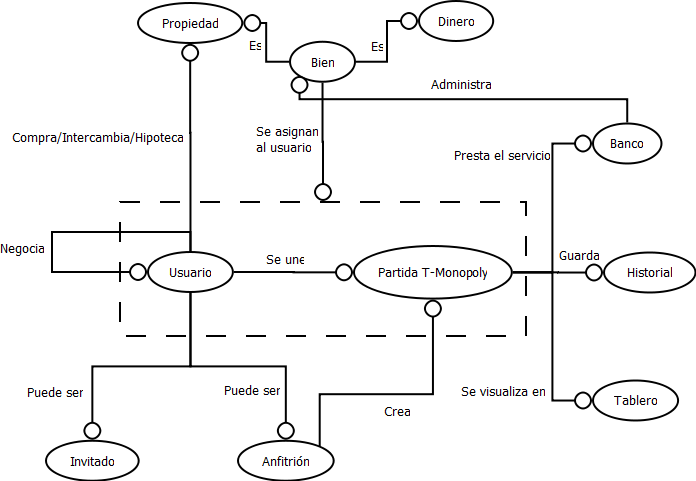


Ilustración ***16***: Modelo conceptual T-Monopoly®.

### 2.2.4 Modelo Del Dominio

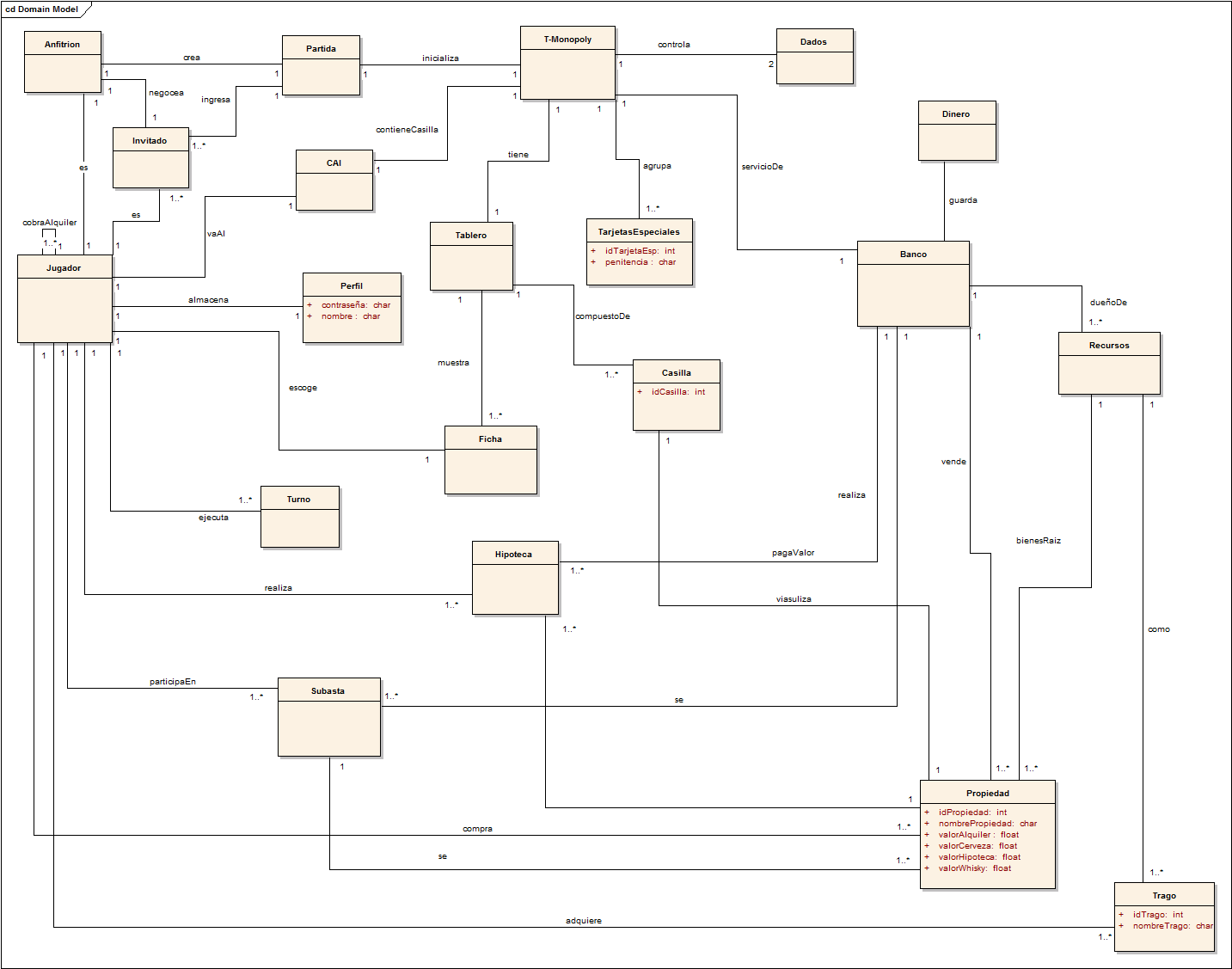
****

Ilustración **17**: 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.

# Especificación De Requerimientos

## 3.1. Especificación de Requerimientos

Con el fin de tener una idea más clara acerca de cada uno de los requerimientos identificados, y del origen e cada uno de estos, Alimnova® ha desarrollado un formato en el que muestra la relación entre los requerimientos dado el caso que existan y las relaciones de estos con los casos de uso, con el fin de mostrar el origen de cada uno de ellos, el cuál es el Documento de Especificación de Requerimientos que se encuentra anexo: [[**Documento de Especificación de Requerimientos**](#_5.2_Especificación_De)].

## 3.2. Especificación de las Categorías Funcionales de requerimientos

De acuerdo a la clasificación de requerimientos descrita anteriormente [[**ver sección 2.2.2**](#_2.2.2._Clasificación_De_1)] 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 que corresponde tanto a las casillas de Guayabo (Arca comunal) como a las de Prendido (Casualidad), Hippie, CAI, Salida, Impuestos. La razón principal de la definición de esta categoría corresponde al tratamiento especial que se le debe dar a las situaciones causadas por estas casillas, puesto que dichas situaciones estarán apegadas a transacciones o movimientos específicos.

#### 3.2.1.1 Guayabo

Corresponde a las tarjetas que muestran leyendas que se deben aplicar al jugador en el momento en que caen en una casilla de este tipo. El número de cartas de Guayabo es 12 de las cuales 6 benefician al jugador.

|  |  |
| --- | --- |
| 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 | Ve directo al CAI sin pasar por la casilla de SALIDA |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |

Tabla X: Descripción de las leyendas de Arca comunal

#### 3.2.1.2Prendido

Corresponde a las tarjetas que muestran leyendas que se deben aplicar al jugador cuando caen en una casilla de este tipo. El número de cartas de casualidad es 12. Es similar al de Guayabo, pero difiere en que las 9 leyendas de este tipo de tarjeta benefician al jugador, mientras que en Arca Comunal este número de leyendas 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 del Hippie el cual corresponde s $10.

#### 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 Guayabo.
   3. Que aparezca en una leyenda de la tarjeta de Prendido.

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 3.2.2](#_3.2.2_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 se tendrá una oportunidad.
   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 la cual se encuentran inicialmente las fichas de todos los jugadores y posteriormente es donde los jugadores reclaman al banco una cantidad de dinero correspondiente a $200, cada vez que pasen por dicha casilla, excepto si en alguna leyenda de Prendido o Guayabo dice “VAS DIRECTO AL CAI SIN PASAR POR LA SALIDA”.

#### 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

El jugador que caiga en esta casilla deberá cancelar la suma de $75

3.2.1.6.2 Posesiones

El jugador que caiga en esta casilla seleccionara una de las 2 opciones: cancelar $200 o la suma de todas sus posesiones (propiedades, tragos). El sistema no revelara la suma de la segunda opción hasta que el jugador seleccione dicha opción.

### 3.2.2 Jugadas Especiales

Corresponde a la situación en la que el jugador lanza los dados y obtiene pares en tres lanzamientos consecutivos. Esta situación 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, cancelando dicho valor.

### 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 [35].

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:

Ilustración ***18: Negociación T-Monopoly®.***

#### 3.2.3.2 Subasta

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

Si el jugador opta por no comprar, 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 1***9***: Subasta 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.

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 y las carreras 12 y 13 [36].

La lista de propiedades se encuentra en el [[**anexo 5.5**](#_5.5_Propiedades_T-Monopoly)].

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

Para el inicio de una partida de T-Monopoly® es necesaria la creación de esta y esta acción solo puede ser efectuada por un anfitrión. Además de esto es necesaria la unión de uno a 4 jugadores mas a la partida los cuales se denominaras invitadas, esto con el fin de satisfacer el argumento mismo del juego (hacerse el más rico de todos los jugadores) y poder finalizarlo en algún momento. Adicionalmente el inicio de una partida incluye la repartición de dinero correspondiente a $1500 y de 2 propiedades a los jugadores la cual es otra modificación de Monopoly adaptada a T-Monopoly® con el fin de agilizar el juego.

#### 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 activos, 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 si 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 de una partida, ya sea voluntario o por un problema en la conexión con el servidor, por parte de los demás jugadores.

#### 3.2.4.3 Banco

El banco guarda, además del dinero, 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, préstamos 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



Ilustración 20: 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



Ilustración 21: 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 Icon

Ilustración 22: Hardware de persistencia

En 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.

Ilustración 23: Software

#### 3.3.2.3 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 24: Interfaces con el usuario

#### 3.3.2.4. 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 [[**..\SRS ENTREGA\Requerimientos\_NoFuncionales (0.2)lineaBase.xlsx**](../SRS%20ENTREGA/Requerimientos_NoFuncionales%20(0.2)lineaBase.xlsx)], 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.

#### 3.3.2.5 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[[**..\SRS ENTREGA\Requerimientos\_NoFuncionales (0.2)lineaBase.xlsx**](../SRS%20ENTREGA/Requerimientos_NoFuncionales%20(0.2)lineaBase.xlsx)].

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 25: Seguridad

#### 3.3.2.7 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 26: Confiabilidad

#### 3.3.2.8 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 15: 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

Para identificar y nombrar las restricciones de diseño tuvimos en cuenta este documento, ya que es un proyecto universitario, de la Universidad Nacional, sede en Manizales, en el cual explican aspectos del diseño, lo cual nos ayudó mucho para desarrollar y llevar a acabo esta sección. [42]

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

## 3.4 Requerimientos De Negocio

Una de las preguntas más comunes que un arquitecto se hace es: ¿Cómo puedo manejar con clientes que no pueden expresarse, cambian su forma de pensar tan fácilmente e introducen cada vez más requerimientos?[43].

Sin embargo, para nuestro proyecto no tendremos ese tipo de problema [[**sección 2.2.1**](#_2.2.1._Riesgos_Generales)] , ya que Miguel Torres, Cliente impuso 3 restricciones desde inicio del proyecto. A continuación se describirán la manera como Alimnova® manejara estas 3 categorías de los requerimientos de negocio.

### 3.4.1 Persistencia

La persistencia en T-Monopoly® se manejará con archivos planos y no con bases de datos.

Entre Las razones principales se encuentran:

* Se almacenara información referente al perfil de un jugador y ranking. Solo si un jugador desea ver el ranking de otra persona abrirá el archivo generado.
* Se almacenara información del historial de la partida mas no se cargara una partida, es decir no se guardara la partida para luego ser reconstruida.

De esta manera la razón principal por la que se decidió manejar archivos planos es que las bases de datos son para aplicaciones más complejas que requieran almacenar bastante información y en el caso de T-Monopoly® únicamente esta la información especificada anteriormente.

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

Ilustración ***19***: Especificaciones de persistencia

El formato a manejar en los archivos será:

1. Archivo perfil de usuario



Ilustración ***20***: Formato archivo de perfiles.

Este formato se aplica para poder almacenar el perfil cuando sea creado uno nuevo, esto con el fin de facilitar ambas actividades (almacenamiento, consulta), como se menciono anteriormente. Los campos serán nombre de usuario y contraseña.

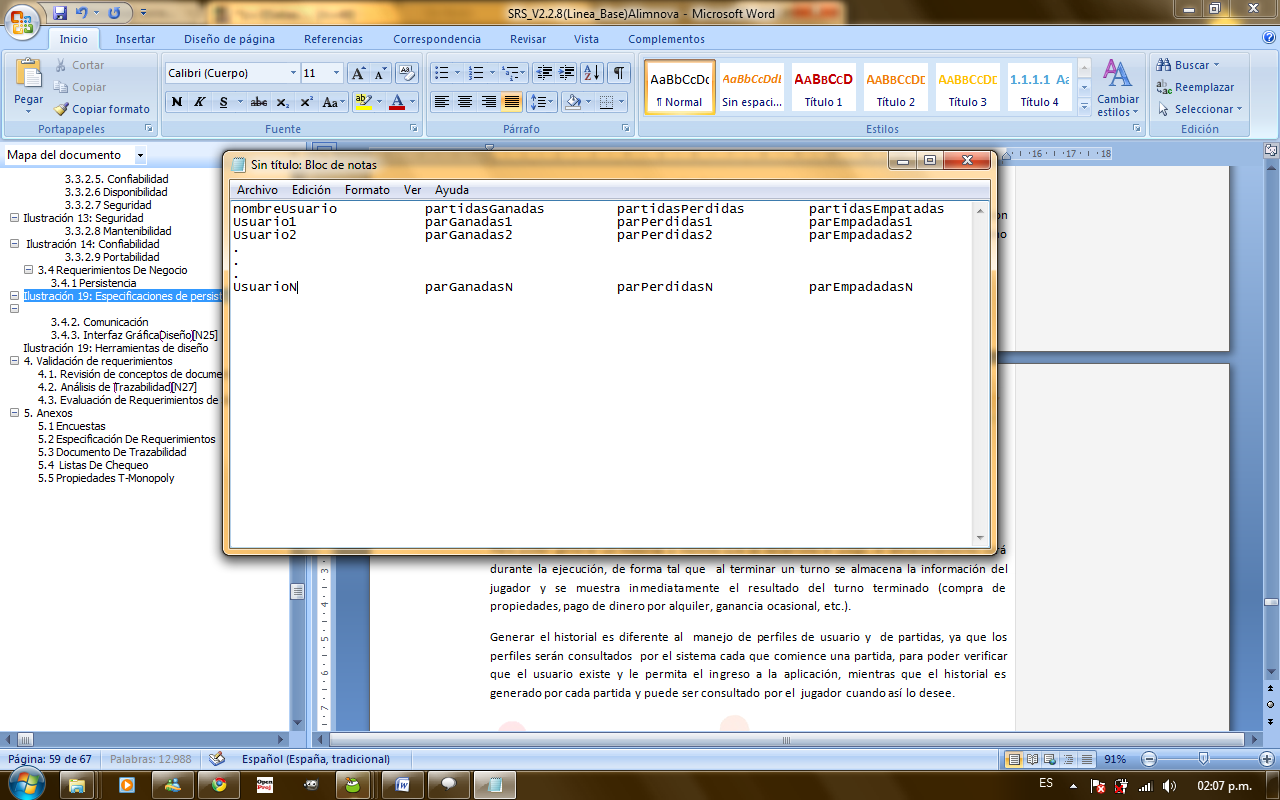
2. El manejo del ranking de partidas:

Ilustración ***21***: Formato archivo de ranking de jugadores.

Este archivo será visible para todos los jugadores no requiere de restringir la consulta. Los campos serán: nombre de usuario, partidas ganadas, partidas perdidas y partidas empatadas.

3. Historial del juego

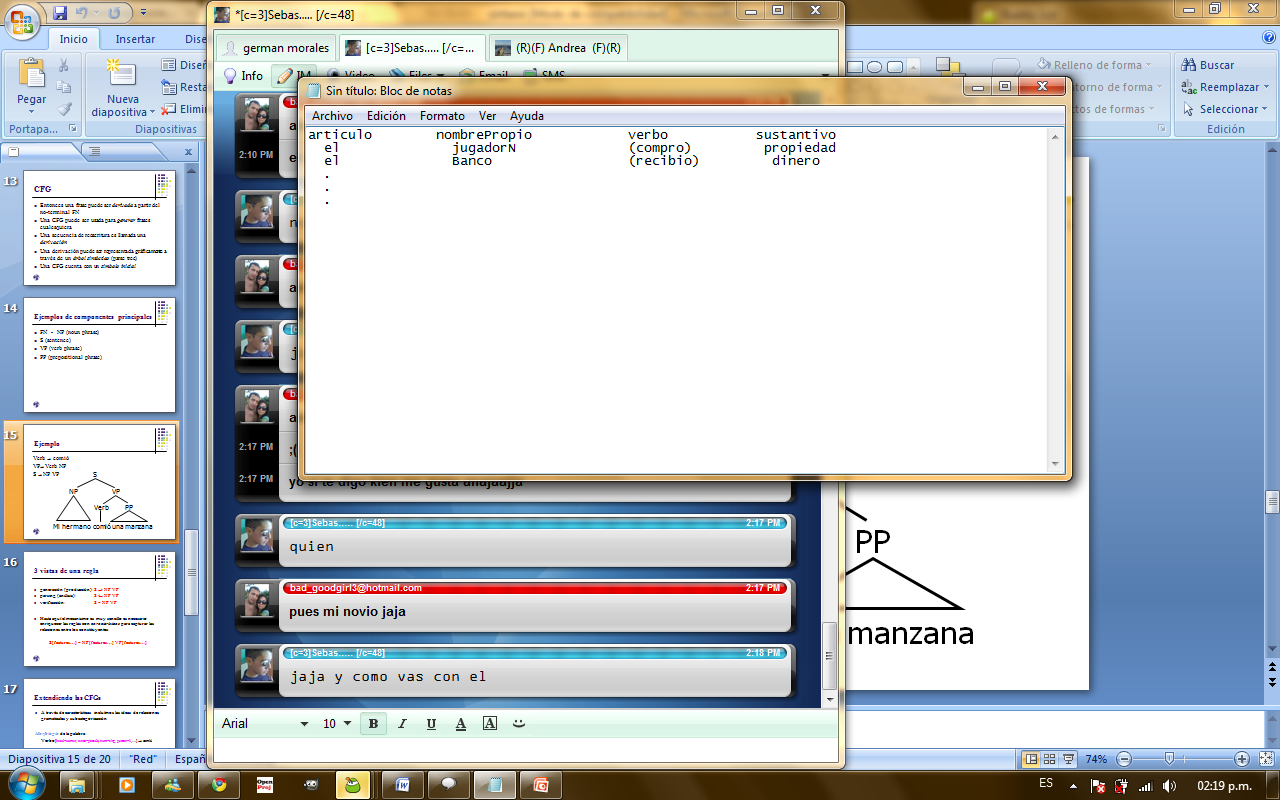


Ilustración ***22***: Formato archivo historial del juego.

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 almacenaran todos los movimientos ejecutados en el turno de del jugador. Los campos son: articulo (el) , nombre propio (referente a que participante ejecuto el movimiento), verbo( entre ellos: comprar, vender, lanzar dados, hipotecar, deshipotecar, intercambio), y sustantivo(referente al objeto: propiedad, tragos, dinero).

### 3.4.2. Comunicación Y Arquitectura Cliente-Servidor

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.

Finalmente respecto a la restricción que debe tener arquitectura Cliente-ServidorAlimnova® tendrá en cuenta que existe un proveedor del juego, para este caso será el servidor, el cual además de contener las funcionalidades del juego actuando como Banco, 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 podrá jugar contra el computador.

### 3.4.3. Interfaz Gráfica

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 estas ya definidas en el SPMP sección 6 [**SPMP**]. Sin embargo, a grandes rasgos las herramientas a utilizar por Alimnova® son flash, para realizar algunas animaciones, y la librería de java Look and Feel[44] el cual le dará un aspecto más amigable a la aplicación, el encargado de dirigir el proceso de desarrollo en flash será Néstor Diazgranados, Diseñador Gráfico, el cual es conocedor de la herramienta acompañado de Germán Morales, Director de desarrollo, quien está relacionado con la parte correspondiente a Java con la librería Look and Feel.

# 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].

Desde el punto de vista de la pirámide de requerimientos, la trazabilidad esta marcada en todas las fases de dicha pirámide, ver explicación [[**sección 2.2.2**](#_2.2.2._Clasificación_De_1)].

Ilustración **26**: Visualización de la trazabilidad en la pirámide de requerimientos. Tomado de [22].

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.

## 5.5 Propiedades T-Monopoly

//Incluirla