

**PATRÓN PARA LA ESPECIFICACIÓN DE REQUISITOS NO FUNCIONALES  
DE FIABILIDAD Y MANTENIBILIDAD DURANTE LA ETAPA  
DE CAPTURA DE REQUISITOS**



Universidad  
del Cauca

**Miller Santiago Castillo Muñoz  
Jhonatan David Zúñiga Jiménez**

**Trabajo de grado para la obtención del título de:  
Ingeniero de Sistemas**

**Directora:  
PhD. Sandra Lorena Buitrón.  
Codirector:  
PhD. Francisco José Pino Correa.**

**Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Programa de Ingeniería de Sistemas  
Línea de investigación Ingeniería de Requisitos**

**Popayán, 2022**

## Tabla de contenido

<b>Capítulo 1. Introducción</b> .....	1
1.1. Planteamiento del problema y justificación .....	1
1.2. Objetivos .....	3
1.2.1. Objetivo general .....	3
1.2.2. Objetivos específicos .....	4
1.3. Estrategia de investigación .....	4
1.3.1. Ciclo conceptual: Análisis conceptual.....	4
1.3.2. Ciclo metodológico: Definición de la estrategia .....	5
1.3.3. Ciclo de Evaluación: Evaluación de la estrategia.....	5
1.3.4. Ciclo de documentación y socialización .....	5
1.4. Organización del documento.....	6
<b>Capítulo 2. Marco teórico y estado actual de la literatura</b> .....	7
2.1. Definición de conceptos relevantes.....	7
2.1.1. Patrón.....	7
2.1.1.1. Patrones de análisis .....	9
2.1.1.2. Patrones de diseño.....	10
2.1.1.3. Patrones de requisitos .....	11
2.1.2. Requisitos .....	11
2.1.2.1. Requisitos funcionales .....	11
2.1.2.2. Requisitos no funcionales (RNF).....	12
Fiabilidad .....	13
Mantenibilidad.....	13
2.1.3. Prototipo de interfaz .....	14
2.1.3.1. Sketch.....	14
2.1.3.2. Wire-frame .....	15
2.1.3.3. Mockups.....	15
2.2. Profundización de conceptos de interés .....	16
2.2.1.1. Elementos o partes de los requisitos .....	16
2.2.1.2. Elementos o partes de los Requisitos No Funcionales.....	17
2.2.1.3. Elementos o partes de los patrones de análisis, diseño y requisitos .....	19
2.2.1.3.1. Patrones de análisis (Fowler, 1997).....	19
2.2.1.3.2. Patrones de diseño propuestas .....	20

2.2.1.3.3. Patrones de requisitos .....	29
2.3. Características de los RNF de fiabilidad y mantenibilidad .....	31
2.3.1. Elementos o atributos de fiabilidad .....	32
2.3.2. Elementos o atributos de mantenibilidad.....	35
2.4. Trabajos relacionados.....	43
2.5. Discusión.....	47
<b>Capítulo 3. Caracterización y definición de los componentes del patrón de especificación de RNF .....</b>	<b>49</b>
3.1. Caracterización de un patrón.....	50
3.2. Caracterización de los Requisitos No Funcionales .....	53
3.3. Definición del patrón de especificación de RNF .....	55
3.3.1. Definición de los elementos a tener en cuenta para la construcción del patrón .....	55
3.3.1.1. Elementos generales de un patrón.....	55
3.3.1.2. Elementos específicos de un patrón .....	55
3.3.2. Estructura del patrón de especificación de RNF .....	59
3.3.3. Definición de los elementos de Requisitos No Funcionales a tener en cuenta para la construcción del patrón .....	62
3.4. Mejora del patrón para la especificación de RNF con aspectos logrados a través de una encuesta sobre patrón MVC .....	66
3.5. Patrón de especificación de RNF en versión final .....	69
3.5.1. Particularidades de las variantes y datos para considerar los elementos de fiabilidad y mantenibilidad en la versión final del patrón. ....	77
3.6 Guía para aplicar los RNF de fiabilidad y mantenibilidad sobre el patrón para la especificación de RNF. ....	78
<b>Capítulo 4. Evaluación del patrón para la especificación de RNF .....</b>	<b>108</b>
4.1. Preguntas de investigación .....	108
4.2. Diseño del estudio de caso .....	109
4.3. Constructos y definiciones operacionales .....	111
4.4. Proposiciones teóricas .....	111
4.5. Unidad de análisis .....	112
4.5.1 Muestra.....	112
4.6. Fuentes e instrumentos de recolección de datos.....	113
4.7. Ejecución del estudio de caso.....	115
4.8. Recolección: .....	120

4.9. Análisis de los resultados .....	131
4.10. Plan de validez .....	139
4.11. Limitaciones .....	140
4.12. Lecciones aprendidas .....	140
<b>Capítulo 5. Conclusiones, lecciones aprendidas y trabajo futuro. ....</b>	<b>142</b>
5.1. Conclusiones .....	142
5.2. Lecciones aprendidas .....	143
5.3. Trabajos futuros.....	144
<b>Referencias Bibliográficas.....</b>	<b>145</b>

## Índice de tablas

<b>Tabla 1.</b> Atributos descriptivos para el análisis de requisitos.....	16
<b>Tabla 2.</b> Atributos de calidad vs subcaracterísticas de calidad.....	42
<b>Tabla 3.</b> Trabajo relacionado # 1. ....	43
<b>Tabla 4.</b> Trabajo relacionado # 2.....	44
<b>Tabla 5.</b> Trabajo relacionado # 3.....	44
<b>Tabla 6.</b> Trabajo relacionado # 4.....	45
<b>Tabla 7.</b> Trabajo relacionado #5.....	46
<b>Tabla 8.</b> Número de ocurrencias de cada elemento que contiene un patrón.....	50
<b>Tabla 9.</b> Número de ocurrencia de cada elemento que contiene un requisito no funcional.....	53
<b>Tabla 10.</b> Resultados de la encuesta del patrón MVC en la pregunta número 1 .....	66
<b>Tabla 11.</b> Resultados de la encuesta del patrón MVC en la pregunta número 2 .....	67
<b>Tabla 12.</b> Resultados de la encuesta del patrón MVC en la pregunta número 4 .....	68
<b>Tabla 13.</b> Fuerzas y alternativas de solución .....	81
<b>Tabla 14.</b> Elementos o participantes que intervienen en el proceso de negocio.....	82
<b>Tabla 15.</b> Estereotipos generales de BPMN.....	83
<b>Tabla 16.</b> Estereotipos propuestos como extensión de BPMN .....	84
<b>Tabla 17.</b> Sección 1 del Instrumento de registro de patrón.....	85
<b>Tabla 18.</b> Sección 2 del Instrumento de registro del patrón - RNF .....	88
<b>Tabla 19.</b> Sección 3 del Instrumento de registro del patrón – Prioridad del RNF.....	89
<b>Tabla 20.</b> Sección 4 del Instrumento de registro del patrón - Otros elementos del RNF .....	91
<b>Tabla 21.</b> Sección 5 del Instrumento de registro del patrón - Datos del RNF .....	92
<b>Tabla 22.</b> Propuesta de estereotipos de representación para fiabilidad.....	97
<b>Tabla 23.</b> Propuesta de estereotipos de representación para mantenibilidad.....	97
<b>Tabla 24.</b> Instrumento de captura del elemento – Ejemplo.....	100
<b>Tabla 25.</b> Identificación de los elementos que intervienen en el proceso de negocio – Ejemplo .....	101
<b>Tabla 26.</b> Sección 1 Instrumento de captura del patrón – Ejemplo .....	101
<b>Tabla 27.</b> Sección 2 Instrumento de captura del patrón – Ejemplo .....	102

<b>Tabla 28.</b> Sección 3 Instrumento de captura del patrón – Ejemplo .....	103
<b>Tabla 29.</b> Sección 4 Instrumento de captura del patrón – Ejemplo .....	103
<b>Tabla 30.</b> Sección 5 Instrumento de captura del patrón – Ejemplo .....	103
<b>Tabla 31.</b> Sección RNF complejos de representa en los prototipos de interfaz.....	106
<b>Tabla 32.</b> Preguntas de investigación de estudio de caso .....	108
<b>Tabla 33.</b> Características de la organización involucrada en el estudio de caso.....	110
<b>Tabla 34.</b> Constructos y definiciones operacionales .....	111
<b>Tabla 35.</b> Descripción de los estudiantes participantes.....	112
<b>Tabla 36.</b> Indicadores y métricas - Estudio de caso.....	113
<b>Tabla 37.</b> Recursos de apoyo .....	115
<b>Tabla 38.</b> Elementos de fuerza – capa de entendimiento.....	122
<b>Tabla 39.</b> Elementos identificados en el proceso de negocio – Estudio de caso .....	123
<b>Tabla 40.</b> Sección 1 Instrumento de registro - Estudio de caso .....	124
<b>Tabla 41.</b> Sección 2” RNF” del Instrumento de registro - sección 2 RNF – estudio de caso....	126
<b>Tabla 42.</b> Instrumento de registro - sección 3 Prioridad del RNF – estudio de caso.....	127
<b>Tabla 43.</b> Instrumento de registro - sección 4 Otros elementos del RNF – estudio de caso.....	128
<b>Tabla 44.</b> Instrumento de registro - sección 5 Datos del RNF – estudio de caso .....	129
<b>Tabla 45.</b> Total, de RNF especificados por característica de calidad .....	132
<b>Tabla 46.</b> Campos diligenciados correctamente en la capa de datos .....	134
<b>Tabla 47.</b> Porcentaje de correctitud de los RNF en la capa de vista .....	135
<b>Tabla 48.</b> Métricas obtenidas a través del uso del patrón - Estudio de caso.....	136
<b>Tabla 49.</b> Total de RNF por proyecto y su promedio de tiempo de especificación por prototipo .....	137
<b>Tabla 50.</b> Promedio de evaluación de los aspectos en la encuesta .....	138
<b>Tabla 51.</b> Nivel de conocimiento adquirido por los participantes .....	138
<b>Tabla 52.</b> Resultados de las preguntas realizadas en la encuesta a los participantes .....	138

## Índice de ilustraciones

<b>Ilustración 1.</b> Convecciones o elementos para los grafos de atributos de calidad.....	32
<b>Ilustración 2.</b> Grafo detallado de fiabilidad.....	33
<b>Ilustración 3.</b> Grafo detallado de mantenibilidad .....	36
<b>Ilustración 4.</b> Caracterización de patrones y de requisitos no funcionales.....	49
<b>Ilustración 5.</b> Primera versión de la propuesta del patrón para la especificación de RNF .....	62
<b>Ilustración 6.</b> Segunda versión de la propuesta del patrón para la especificación de RNF.....	65
<b>Ilustración 7.</b> Resultados de la encuesta del patrón MVC en la pregunta número 3.....	68
<b>Ilustración 8.</b> Versión final de la propuesta de patrón para la especificación de RNF .....	70
<b>Ilustración 9.</b> Particularidades de variantes y datos - Elementos de fiabilidad y mantenibilidad	78
<b>Ilustración 10.</b> Grafo de eficiencia de desempeño .....	86
<b>Ilustración 11.</b> Grafo de portabilidad .....	86
<b>Ilustración 12.</b> Grafo de seguridad .....	87
<b>Ilustración 13.</b> Matriz de Stephen Covey .....	90
<b>Ilustración 14.</b> Colores para representación de RNF para proceso del diagrama de estructura ..	94
<b>Ilustración 15.</b> Plantilla descriptiva para la representación de los RNF en los prototipos .....	95
<b>Ilustración 16.</b> Sección asociada a la plantilla de representación de los RNF en los prototipos.	96
<b>Ilustración 17.</b> Técnica espina de pescado - Ejemplo .....	99
<b>Ilustración 18.</b> Modelado 1 – Ejemplo de seguridad.....	102
<b>Ilustración 19.</b> Modelado con los RNF numerado para realizar el prototipo de interfaz – Ejemplo.....	104
<b>Ilustración 20.</b> Vista general de RNF representado en prototipo de interfaz – Ejemplo.....	104
<b>Ilustración 21.</b> Vista general de RNF de seguridad en prototipo de interfaz – Ejemplo.....	105
<b>Ilustración 22.</b> Vista de opción ver prioridad de la plantilla descriptiva – Ejemplo .....	105
<b>Ilustración 23.</b> Plantilla descriptiva del prototipo .....	106
<b>Ilustración 24.</b> Representación en prototipo de interfaz para RNF de Eficiencia de desempeño – Ejemplo .....	107
<b>Ilustración 25.</b> Modelado inicial de los procesos – capa de estructura del estudio de caso .....	123
<b>Ilustración 26.</b> Modelado final de los procesos en la capa de estructura - Estudio de caso .....	125

<b>Ilustración 27.</b> Prototipo 1 de usabilidad - estudio de caso .....	130
<b>Ilustración 28.</b> Plantilla descriptiva del prototipo 1 de usabilidad - estudio de caso .....	130
<b>Ilustración 29.</b> Plantilla descriptiva del prototipo 1 de fiabilidad - estudio de caso.....	131
<b>Ilustración 30.</b> Porcentaje de RNF especificados por característica de calidad .....	133



## Capítulo 1. Introducción

### 1.1. Planteamiento del problema y justificación

En el desarrollo de *software* es importante considerar el proceso de Elicitación de Requisitos (ER) como base para las etapas siguientes del desarrollo del mismo, pues garantiza que los requisitos del sistema respondan a las necesidades de la organización que lo utilizará y con las futuras necesidades de los usuarios (Villanueva, 2009). El proceso de ER se refiere al aprendizaje y entendimiento de las necesidades de los usuarios e interesados en un proyecto, con el objetivo principal de comunicar dichas necesidades a los desarrolladores del sistema para que estos las plasmen de manera adecuada y completa (Buitrón-Ruiz, 2018).

Dentro de este proceso se toman en cuenta dos elementos principales: los Requisitos Funcionales (RF) y los Requisitos No Funcionales (RNF) (Chun, 2009). Los RF son las acciones que debe realizar el software sin considerar las limitaciones físicas; mientras que los RNF son los que definirán las propiedades ambientales y las restricciones de implementación relacionadas con el desempeño del producto software (Pandey, 2010). Además, Franch (1998) señala que “la funcionalidad está relacionada con lo que el sistema hace y su no funcionalidad o calidad se refiere a cómo el sistema se comporta frente a atributos observables como el desempeño, reusabilidad y confiabilidad” (p. 60), la característica observable hace referencia a como se puede determinar el comportamiento interno de dichos atributos en un sistema determinado. Es por eso que durante la ER es importante especificar esas necesidades del cliente, razón por la cual se puede entender esta actividad como la documentación de los acuerdos entre el cliente y el grupo de desarrollo, en la que se prescriben los requisitos técnicos que debe cumplir un producto, un proceso, un servicio o un sistema, y así llevar a cabo la totalidad de exigencias estipuladas (Arévalo, 2008). En este sentido, es importante reconocer que el éxito de un sistema depende en gran parte de una buena especificación, de un mutuo acuerdo entre clientes y proveedores frente a los RNF que estarán presentes en el producto y sobre el adecuado desarrollo de estos (Yin, 2012).

En ingeniería de software, los RNF en lugar de ser tenidos en cuenta desde el inicio del proceso de elicitación de requisitos en los proyectos de desarrollo, se pasan por alto hasta las últimas etapas de desarrollo de software (Maiti, 2015). Así mismo, en reportes de *Standish Groups's International* se identifican dificultades o problemas asociados con la no consideración de RNF durante el proceso de desarrollo de software, entre los cuales se encuentra, según Clancy (2014): (i) retrabajo por la no definición clara de los requisitos (en un 13%), (ii) deterioro y cancelación de los proyectos de desarrollo de software debido a requisitos incompletos (en un 13,1%), (iii) cambios en los proyectos debido a la falta de entradas de información del stakeholder (en un 12,8%), (iv) cambios debido a requisitos y

especificaciones incompletas (en un 12,3%), y (v) cambios debido a modificaciones en los requisitos y especificaciones (en un 11,8%).

Autores como Garzas (2013), confirman que:

Frente al objetivo de lograr la calidad en las organizaciones, el 5,88% de las no conformidades encontradas se deben a errores en la definicion de los requisitos por parte de los *stakeholders*, y el 12,94% de ellas al analisis indebido de los requisitos del sistema.

Por lo anterior, se puede evidenciar que la falta de mecanismos para la ER no funcionales bien definidos y estructurados puede generar problemas en el desarrollo (Buitron-Ruiz, 2018).

Bajo esta misma perspectiva, frente a la omision de los RNF en los proyectos de desarrollo software, Maiti (2015) confirma que incluso las metodologas agiles son negligentes respecto a la elicitacion de estos requisitos, lo que genera un impacto negativo en los productos de software: calidad deficiente, mayor costo para solucionar problemas en etapas posteriores de desarrollo, retrasos en la salida al mercado (Cysneiros 2004), errores en el desarrollo del producto (Breitman, 1999), y por tanto una tasa de fracaso del 60% o mas en los proyectos (Bajpai & Gorthi, 2012). Estas estadsticas soportan y evidencian la importancia de una correcta y completa ejecucion de la elicitacion de RNF.

Adicional a lo anterior, Rodriguez & Piattini (2014) muestran como, dentro del grupo de RNF que no son tenidos en cuenta, se encuentra la mantenibilidad y la fiabilidad, los cuales son cruciales en la etapa de desarrollo, dado que:

- En primer lugar, la mantenibilidad supone una de las fases del ciclo de desarrollo mas costosas, llegando al alcanzar el 60% de sobrecostos. Tambien, es una de las caractersticas mas demandadas en la actualidad por los clientes de software, ya que requieren que el producto desarrollado pueda ser, posteriormente, mantenido por ellos mismos o incluso por un tercero; y, por ultimo, se debe tener en cuenta que las tareas de mantenimiento sobre productos con poca mantenibilidad tienen mas probabilidad de introducir nuevos errores en el producto.
- En segundo lugar, Pecht (2009) presenta la fiabilidad como una caracterstica crtica, pues determina la efectividad del producto para evitar fallas. Ademas, el autor seala que una deficiencia en esta caracterstica, eventualmente, resultar en un rendimiento deteriorado o perdido, seguridad comprometida y la necesidad de acciones restaurativas como diagnstico, reparacion y mantenimiento. Los productos con alta fiabilidad funcionarn por mas tiempo, permitiendo que los recursos se centren en mejorar el rendimiento. Por las razones mencionadas anteriormente los RNF de fiabilidad y mantenibilidad fueron propuesto como estudio en el presente estudio.

La omisión de los RNF de fiabilidad y mantenibilidad es recurrente en el área de Ingeniería de Requisitos, por lo que se hace necesario generar o buscar una estrategia que le permita al analista especificarlos, involucrando al cliente en esta actividad. Una estrategia válida que se puede considerar para direccionar la especificación de estos RNF es el concepto de patrón, puesto que mediante este se puede describir una solución para dicho problema recurrente.

Un patrón ofrece un esquema, el cual establece una relación entre: un **contexto** dado, que describe las situaciones en las que ocurre el problema; y una **solución** adecuada para resolver el problema recurrente (Buschmann *et al.*,2007). Para este trabajo de grado, el contexto es la especificación de requisitos, el problema es la inadecuada e incompleta especificación de RNF debido a la ausencia de mecanismos que existe para esta, y por último la solución adecuada para resolver el problema recurrente será la que nos arroje el esquema brindado por el patrón.

Dentro de la solución del patrón, se incluirá una propuesta que permita visualizar la especificación obtenida, esta hará posible la visualización de los RNF usando algunos estereotipos planteados en este trabajo de grado, o plasmando dicha especificación sobre los prototipos de interfaz.

La selección del prototipo (*mockup*, *sketch* y *Story-Board*) como forma de especificación a ser utilizada para este estudio se debe a que es una forma que permite: (i) al usuario idear preliminarmente el producto software, y (ii) al ingeniero de requisitos obtener retroalimentación temprana que aclara los requisitos del sistema levantados (Preece, 1994). Estos aspectos podrían apoyar el propósito del trabajo de grado: crear un patrón que permita llegar a una solución estructurada que pueda usarse recurrentemente y de manera sistemática en la actividad de elicitación, de tal manera que los RNF especificados puedan ser visibles en el proyecto de desarrollo de software por medio de un prototipo.

Bajo este contexto, en este trabajo de grado se plantea la siguiente pregunta de investigación: **¿Cómo un patrón apoya el proceso de especificación de requisitos no funcionales plasmando estos requisitos sobre el prototipo de software?**

## **1.2. Objetivos**

### **1.2.1. Objetivo general**

Proponer un patrón para la especificación<sup>1</sup> de requisitos no funcionales, mediante el cual se logren capturar<sup>2</sup> dichos requisitos y queden reflejados<sup>3</sup> en un prototipo de interfaz.

### 1.2.2. Objetivos específicos

- Identificar a partir de la literatura los elementos relevantes del concepto de patrón y de un prototipo de interfaz.
- Caracterizar la información y elementos de los requisitos de Fiabilidad y Mantenibilidad, según la literatura, para el posterior diseño del patrón.
- Diseñar el patrón para la especificación de los RNF de Fiabilidad y Mantenibilidad, considerando los elementos identificados previamente.
- Aplicar el patrón obtenido, a través de un estudio de caso para evaluar su idoneidad en la etapa de captura de requisitos.

### 1.3. Estrategia de investigación

Para llevar a cabo el trabajo de investigación se utilizó la metodología de investigación-acción multi-ciclo con bifurcación (Pino *et al.*, 2013), de la cual se tomaron los siguientes ciclos: conceptual, metodológico, de evaluación y de documentación.

#### 1.3.1. Ciclo conceptual: Análisis conceptual

- **Estudio de la literatura:** Se identificó desde la literatura los problemas que han surgido en las empresas de software a raíz de insuficientes mecanismos de especificación de RNF y las consecuencias que esto conlleva para los productos software.
- **Identificar componentes de la propuesta:** Se revisaron e identificaron, a partir de la literatura dos componentes: (i) los componentes del patrón, y (ii) los componentes de un prototipo de interfaz.

---

<sup>1</sup> Documentación en la que se prescriben los requisitos técnicos que debe cumplir un producto, proceso, servicio o sistema (DRAE)

<sup>2</sup> Captar o reflejar un aspecto de la realidad en una obra con soporte

<sup>3</sup> Manifestar, hacer patente o dejar ver algo.

- **Justificar los requisitos no funcionales seleccionados para el patrón:** Se realizó una investigación que permitió justificar la decisión sobre qué atributos de calidad son tenidos en cuenta para el diseño del patrón y su representación en el prototipo de interfaz.

### 1.3.2. Ciclo metodológico: Definición de la estrategia

- **Análisis de componentes:** Se revisan los componentes identificados en el análisis conceptual y se seleccionan los componentes que harán parte del patrón para la especificación.
- **Definición del patrón para la especificación:** Se establece el patrón para la especificación de los RNF para los atributos de calidad candidatos, incluyendo las formas de evidenciarlos sobre el prototipo de interfaz.

### 1.3.3. Ciclo de Evaluación: Evaluación de la estrategia

Se realiza la evaluación del patrón para la especificación propuesta, mediante un estudio de caso (Brereton, 2008).

- **Diseño de la investigación:** Se definen los objetivos, preguntas, temas de investigación, así como el rol del investigador y la realización del diseño para, finalmente, elaborar la estructura.
- **Recopilación de datos:** Se organiza el equipo de entrevistados y se consideran los distintos tipos de datos disponibles e iniciar la recopilación de datos en el documento establecido.
- **Análisis:** Se analizan los resultados obtenidos para, posteriormente, concluir los resultados.
- **Informe o documentación:** Se documentan los resultados obtenidos del estudio de caso en el formato establecido para tal fin.

### 1.3.4. Ciclo de documentación y socialización

Se realiza de manera transversal al desarrollo de trabajo de grado con el objetivo principal de organizar toda la documentación obtenida en los ciclos preliminares, dándoles una estructura

dentro de la monografía, de tal manera que se den las conclusiones del trabajo para, posteriormente, realizar un artículo con los resultados obtenidos.

- **Elaboración de la monografía:** para la realización de la documentación de la monografía se tuvo en cuenta en este trabajo de investigación, la necesidad de resolver un problema que es recurrente dentro del área de la especificación de requisitos no funcionales, mediante el concepto de patrón. Este trabajo recopila el patrón propuesto, la guía de aplicabilidad del patrón y la validación de este. Dentro de su escritura se tuvo presente las normas estipuladas y reglamentarias (Normas APA).
- **Divulgar y sustentar los resultados:** divulgar los resultados del trabajo de grado y sustentarlos.

#### **1.4. Organización del documento**

El presente trabajo de grado, incluyendo la sección de introducción, se organiza por capítulos. El capítulo dos presenta el estado del arte relacionado con el tema de investigación propuesto para este trabajo, describiendo los resultados obtenidos en el método de búsqueda de información. También, se evidencia el cumplimiento del ciclo conceptual de la estrategia de investigación propuesta. Luego, el capítulo tres expone la conceptualización realizada del tema de patrones para la especificación de requisitos, a partir de la cual se construye: una definición ampliada, una mejora de esta y una guía para considerar la aplicación de los RNF sobre el patrón propuesto. Seguido, se encuentra el cuarto capítulo que da a conocer la aplicación y evaluación del patrón de especificación, desarrollada a través de un estudio de caso que fue aplicado a un proyecto de desarrollo de la Universidad del Cauca. Este estudio de caso incluye el contexto de la investigación, resultados y análisis. Finalmente, el capítulo cinco presenta el resumen, aportes de investigación, conclusiones, lecciones aprendidas y trabajos futuros en el proceso de investigación llevado a cabo para obtener la estrategia propuesta.

## **Capítulo 2. Marco teórico y estado actual de la literatura**

En este capítulo, se presenta la definición de los conceptos más importantes que se involucran en la propuesta, como son: patrón, requisitos no funcionales y prototipos de interfaz, buscando identificar las partes relevantes de conceptos que serán tenidos en cuenta para la posterior definición y construcción del patrón en la especificación de RNF. Asimismo, se presenta un análisis de los trabajos relacionados con la propuesta en términos de: objetivo, propuesta, resultado y diferencias entre la propuesta analizada con nuestra propuesta.

### **2.1. Definición de conceptos relevantes**

#### **2.1.1. Patrón**

Un patrón es una idea que ha sido útil en un contexto y probablemente lo sea en otros. Es un modo de proveer información en forma de una declaración de problema, una presentación de una solución ampliamente aceptada al problema, y luego una discusión de las consecuencias de esa solución. El patrón involucra una descripción general de una solución recurrente a un problema recurrente con diversos objetivos y restricciones. Además, identifica algo más que simplemente una solución, también explica por qué la solución se necesita (Rosanigo, 2000).

Los patrones no son principios abstractos que requieran su redescubrimiento para obtener una aplicación satisfactoria, ni son específicos a una situación particular o cultural; son algo intermedio. Un patrón define una posible solución correcta para un problema de diseño dentro de un contexto dado, describiendo las cualidades invariantes de todas las soluciones (Alexander, 1964).

Los patrones, son más que simples soluciones, son buenas soluciones. Una solución solo tiene sentido si la aplica en el momento adecuado (Berczuk & Appleton, 2003). Un patrón ayuda con la creación de unidades similares en su amplia estructura, sin embargo, suelen ser muy diferentes en su apariencia detallada. Así mismo, los patrones ayudan a resolver problemas, mas no proporcionan soluciones completas.

Un patrón proporciona un esquema para una solución genérica a una familia de problemas, en lugar de un módulo prefabricado que se puede usar 'tal como está', el cual debe implementarse de acuerdo con las necesidades específicas del problema en cuestión. El esquema en su totalidad denota un tipo de regla que establece una relación entre un contexto dado, un

cierto problema que surge en ese contexto y una solución adecuada para el problema. Las tres partes de este esquema están estrechamente acopladas, se describen como sigue (Buschmann *et al.*, 2007):

1. **Contexto:** Describe situaciones en las que ocurre el problema.
2. **Problema:** Describe el problema que surge repetidamente en el contexto dado, y comienza con una especificación general del problema que permite capturar su propia esencia.
3. **Solución:** Muestra cómo resolver el problema recurrente, o mejor, cómo equilibrar las fuerzas asociadas con él.

Una mirada más cercana a los patrones existentes revela que estos cubren varios rangos de escala y abstracción. Algunos patrones ayudan a estructurar un sistema en subsistemas, mientras que otros, apoyan el refinamiento de los subsistemas y componentes, o las relaciones entre ellos; también van desde los independientes del dominio, como aquellos para desacoplar componentes interactivos, hasta patrones que abordan aspectos específicos del dominio (Buschmann *et al.*, 2007).

Para que un patrón cumpla de manera adecuada y brinde una buena solución, debe presentar algunas cualidades como:

- (i) Encapsulamiento y Abstracción: cada patrón encapsula un problema bien definido y su solución en un dominio particular, también abstrae conocimiento y experiencia del dominio.
- (ii) Equilibrio: tiene que comprender algún tipo de balance entre sus metas y sus restricciones para minimizar el conflicto dentro del espacio de la solución.
- (iii) Apertura y Variabilidad: cada patrón puede trabajar junto con otros para resolver un problema mayor y en una variedad de implementaciones.
- (iv) Capacidad de generación y de composición: un patrón, una vez aplicado, genera un contexto resultante que puede coincidir con el contexto inicial de otros patrones. Por aplicación de estos patrones posteriores se progresa hacia el objetivo final para generar un "todo" o solución global completa.



Aplicando un patrón se provee un contexto para la aplicación del siguiente patrón. En un nivel particular de abstracción y granularidad puede estar compuesto con otros patrones (Rosanigo, 2000).

Al revisar la literatura respecto al tema en cuestión, se hallaron diferentes tipos de patrones entre los cuales están: los patrones arquitectónicos, de codificación, de análisis, de diseño y de requisitos. Para efectos conceptuales y prácticos, el presente trabajo de grado tendrá en cuenta los patrones de análisis, diseño y de requisitos, dado que la naturaleza del problema que pretenden cubrir estos patrones es más cercana al contexto de elicitación de requisitos y, por tanto, más compatible con el problema de la no elicitación de requisitos no funcionales; es decir, el análisis y diseño son las etapas más próximas a la actividad de elicitación de requisitos en el contexto del ciclo de desarrollo software.

En torno a los patrones mencionados, se realizó una revisión literaria con el fin de llevar a cabo una caracterización de estos tipos de patrón. A continuación, se presentan las definiciones más relevantes:

#### **2.1.1.1. Patrones de análisis**

Es un tipo de patrón que, aunque no es tan común como los patrones de diseño o arquitectónicos, está orientado también a la fase de análisis y toma de requerimientos. Cuando el analista realiza una toma de requerimientos apoyándose en los casos de uso que el cliente comunica, todo parece, en un primer momento, sin complicación alguna; no obstante, con el paso del tiempo y la experiencia se puede observar que la mayoría de los negocios son mucho más complejos de lo que parecen. En la mayoría de los casos, las empresas de desarrollo no se dedican a una sola área, sino que hacen desarrollos para varios sectores. Teniendo en cuenta esto, los analistas realizan una toma de requerimientos de muchos tipos de proyectos, en su mayoría sin nada en común entre ellos, por lo que en algunas oportunidades el estado inicial de conocimiento sobre el modelo de negocio de alguno de los proyectos en cuestión puede ser muy básico y es el cliente el experto, o debería serlo, quien se le asigna toda la responsabilidad de transmitir de forma precisa y eficaz toda la información correspondiente al tema que permita una correcta toma de requerimientos. Lastimosamente en la mayoría de los casos, el cliente no transmite la totalidad de información, y sucede que la información obtenida puede estar

incompleta o prestarse para una interpretación errónea y llevar a una aplicación que no responde a los requerimientos y necesidades del cliente.

Para evitar este tipo de situaciones existen los patrones de análisis que son en realidad modelos conceptuales, hechos de manera general de los tipos de negocio más habituales, sirviendo de apoyo para que un analista pueda entender los distintos contextos donde se desenvuelve el software a desarrollar, bien sea salud, finanzas, entre otros. Teniendo este apoyo, el analista puede validar con el cliente, quien es el experto en el campo (Fowler, 1997).

Fouler (1997) sugiere que, para la elaboración de un patrón de análisis, existe primero una fase llamada **Objeto Focal**, en la cual se dice que, para la elaboración de un software, se debe realizar una descripción completa del problema que se presenta, de igual manera para sus requerimientos. El problema se define por medio de un análisis, el cual servirá para que con base a él se busque identificar los objetos que se pueden identificar en dicho problema.

Siguiente a la fase de objeto focal, se procede a realizar el **desarrollo**, donde se define como tal la estructura conceptual del análisis realizado (Fouler, 1997). Se sugiere que para desarrollar un buen patrón de análisis se tenga un formato coherente y uniforme.

#### **2.1.1.2. Patrones de diseño**

En Ingeniería de Software, un Patrón de Diseño nombra, motiva y explica de forma sistemática un diseño general que afronta un problema de diseño recurrente en los sistemas orientados a objetos. Esto, partiendo de que para Alexander, (1964): “cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, y luego describe el núcleo de la solución a ese problema, de tal manera que puede usar esta solución un millón de veces sin tener que hacer lo mismo dos veces”. Aunque el autor se refería a patrones aplicados en construcción de edificios y ciudades, termina guardando cierta similitud en nuestro contexto acerca de los patrones de diseño orientados a objetos, pues nuestras soluciones se expresan en términos de objetos e interfaces en lugar de paredes y puertas, pero en el centro de ambos tipos de patrones hay una solución a un problema en un contexto.

Un patrón de diseño ofrece pistas para la implementación y ejemplos. Los diseñadores expertos no resuelven los problemas desde sus principios, sino que reutilizan soluciones que han funcionado en el pasado. Estos patrones resuelven problemas de diseño específicos y hacen el diseño flexible y reusable (Gamma, 1995).

Para Gamma (1995), un patrón de diseño permite capturar la experiencia, y es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular. El uso de estos patrones ayuda a obtener software de calidad (reutilización y extensibilidad).

### **2.1.1.3. Patrones de requisitos (Withall, 2007)**

Los patrones de requisitos de software pueden ayudar a cualquier analista a escribir mejores requisitos, estos patrones proporcionan una forma de incorporar el conocimiento integral y estructurado sobre diferentes tipos de requisitos. Pueden ayudar a: (i) los analistas en la formulación de las preguntas correctas para comprender y especificar adecuadamente los requisitos de muchos tipos en un nivel apropiado de detalle, (ii) los desarrolladores y tester sobre cómo implementar y probar los requisitos en las siguientes etapas de desarrollo. Para este fin, los patrones de requisitos proporcionan tanto plantillas como ejemplos que permitan entender más fácilmente estos requisitos.

## **2.1.2. Requisitos**

Un requisito se define como la necesidad o expectativa establecida, generalmente implícita u obligatoria (Norma ISO/IEC/IEEE 9000). Este debe decir qué características del sistema servirán y satisfarán un contexto. La definición de requisitos no se limita solamente al análisis del problema que produce una especificación funcional, es mucho más que eso, pues debe abarcar “todo lo necesario” para sentar las bases de las etapas posteriores en el desarrollo del sistema. Así, esta definición debe tratar abordar tres temas: análisis del contexto, especificación funcional y restricciones de diseño (Ross *et al.*, 1997). A menudo, los requisitos de sistema software se clasifican en funcionales y no funcionales (Sommerville, 2005).

### **2.1.2.1. Requisitos funcionales (Sommerville, 2005)**

Los requisitos funcionales para un sistema se refieren a lo que el sistema debe hacer y dependen del tipo de software que se esté desarrollando, de los usuarios esperados del software y del enfoque general que adopta la organización cuando se escriben los requisitos. Detallan las funciones del sistema, sus entradas y salidas, sus excepciones, entre otros. Los requisitos

funcionales del sistema varían desde requisitos generales que cubren lo que tiene que hacer el sistema, hasta requisitos muy específicos que reflejan maneras locales de trabajar o los sistemas existentes de una organización.

### **2.1.2.2. Requisitos no funcionales (RNF)**

Los requisitos no funcionales o de calidad, describen el nivel de calidad de los servicios junto con sus capacidades, y especifican propiedades importantes del sistema, como la seguridad, el rendimiento o la facilidad de uso. Es decir, describen no lo que hará el software sino cómo éste lo hará (Adams, 2015).

Los RNF han sido referidos como "*-ilities*" o "*-ities*", p.e.: *usability* (usabilidad), *integrity* (integridad), es decir, palabras que terminan con la cadena "*-ility*" o "*-ity*". Hay muchos otros tipos de RNF que no terminan con "*-ility*" o "*-ity*", como *performance* (rendimiento), *user-friendliness* (facilidad de uso) y *cohesiveness* (cohesión). Además de otros nombres que se les designa como requisitos de calidad o requisitos de no comportamiento, en el área de arquitectura de software un término clave encontrado con frecuencia es: "atributos de calidad", que se entiende como un conjunto de inquietudes relacionadas con el concepto de calidad. Algunos autores dividen estos atributos de calidad en niveles: de calidad básica (funcionalidad, confiabilidad, facilidad de uso y seguridad) y de calidad adicional (flexibilidad, capacidad de reparación, adaptabilidad, comprensibilidad, documentación y capacidad de mejora) (Chung, 2009).

Los requisitos no funcionales son vitales para el éxito de los sistemas de software, por tanto, si no se tratan adecuadamente, se producen resultados indeseables, que derivan en usuarios, desarrolladores y clientes insatisfechos, o sobrecargas de planificación y de presupuesto para poder corregir el software que se desarrolló sin tener en cuenta estos requisitos no funcionales (Miller, 2009).

Una propuesta taxonómica de los RNF que deben ser tenidos en cuenta en la industria del software es la ISO 25010. En esta taxonomía, se listan ocho atributos de calidad software, entre los cuales siete están relacionados a atributos no funcionales del producto software, y en los que se encuentra la *fiabilidad* y la *mantenibilidad*, aspectos de mucho interés para este trabajo.

## ***Fiabilidad***

En el estándar ISO 25000, la fiabilidad se define como la capacidad de un sistema para desempeñar las funciones específicas bajo condiciones y tiempo determinado, la cual incluye unos sub atributos como:

- a) Madurez → capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
- b) Disponibilidad → capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.
- c) Tolerancia a fallos → capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.
- d) Capacidad de recuperación → capacidad del producto software para recuperar los datos directamente afectados y reestablecer el estado deseado del sistema en caso de interrupción o fallo.

## ***Mantenibilidad***

De otro lado, la mantenibilidad se define como la capacidad del sistema de ser modificado efectiva y eficientemente debido a necesidades evolutivas, correctivas o perfectivas. Esta mantenibilidad a su vez incluye los siguientes atributos:

- a) Modularidad → permite que un cambio en un componente tenga un impacto mínimo en los demás.
- b) Reusabilidad → capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.
- c) Analizabilidad → facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software.
- d) Capacidad para ser modificado → capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
- e) Capacidad para ser probado → facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios (ISO 25000).

### 2.1.3. Prototipo de interfaz

Los prototipos de interfaz permiten principalmente validar el cumplimiento de requisitos de usabilidad (Houde & Hill, 1997) y pueden ser: *sketch*, *wireframe*, y *mockup digital* (Rosson, 2002). A continuación, se describen cada uno de ellos:

#### 2.1.3.1. Sketch

Es una forma de comunicación eficaz para ilustrar ideas o compartir pensamientos que pueden ser plasmados en papel. Según Leblanc (2015), se suelen utilizar símbolos e íconos propios para los elementos que se quiere dibujar y algunas herramientas que se emplean generalmente para la creación de un *sketch* son: papel, lápiz y tablero.

Dentro de los aspectos más importantes que poseen estas clases de prototipo, se encuentran los siguientes:

- ✓ La primera versión de la interfaz de desarrollo.
- ✓ Contenidos generales que tendrá el software.
- ✓ Libertad imaginativa, pues no hay límites a la hora de hacer un boceto.
- ✓ Forma rápida y económica de ayudar a organizar sus pensamientos
- ✓ Proporciona una buena forma de convertir ideas abstractas en algo concreto y crear una estructura básica.
- ✓ Se debe tener mucha paciencia, pues no todas las personas tienen en todo momento lo mejor de su creatividad, puede que transcurra un tiempo antes de tener otra buena idea o hasta que pase algún tiempo antes de poder empezar a llenar un papel en blanco.

De igual manera, es importante tener en cuenta ciertas recomendaciones al momento de crear un *sketch*. En primer lugar, se debe tener clara la intención del desarrollo, pues según la intención que este tenga, las ideas presentadas serán más idóneas y acordes a lo requerido. Asimismo, las funcionalidades que tendrá el desarrollo deben ser previa y minuciosamente analizadas para finalmente tener una lista de ideas que se tengan en mente y quedarse con las que mejor se ajusten a las necesidades del usuario (Arnold, 2007).

### 2.1.3.2. Wire-frame

Es una guía visual que representa el esqueleto de las interfaces del sistema interactivo que se está diseñando. Carece de características de detalle como el estilo tipográfico y el color, pues se centra en la funcionalidad, el comportamiento, y la disposición de los contenidos (prioridades de la información) (Cao, 2015). Para la construcción de un *wireframe* es necesario herramientas básicas, como papel, lápiz y en algunos casos colores.

Entre las características propias de un *wireframe* se encuentran:

- ✓ Son típicamente un plano básico de baja fidelidad, generalmente representado con cuadros grises y marcadores de posición para contenido detallado (Saltiveri, 2011).
- ✓ Son el esqueleto de las interfaces del sistema interactivo que se está diseñando.
- ✓ Carece de características de detalle como estilo tipográfico y color.
- ✓ Se centra en el tipo de información que se va a mostrar, gama de funciones disponibles, prioridades de la información y funcionalidades, formas de visualizar ciertos tipos de información (Brown, 2010).

Para la creación de un *wireframe*, es importante tener en cuenta que este suele iniciarse a partir de un prototipo de baja fidelidad, en este caso un *sketch*. Posteriormente, se organizan las ideas y se decide cuáles dejar para luego ser plasmadas en el *wireframe*. Para concretar, se hace una representación de la información que está ya priorizada y la forma en que esta se va a presentar (Brown, 2010).

### 2.1.3.3. Mockups

**Mockup digital** (Saltiveri, 2011):

Son representaciones de calidad en formato digital que normalmente llenan el espacio entre el prototipo de papel y la versión definitiva de una interfaz o parte de ella. Para realizar un mockup digital no son suficientes los materiales básicos, sino que es necesario utilizar herramientas más sofisticadas (editores gráficos) que implican mayor tiempo de desarrollo y mejor preparación de las personas que los realizan.

Algunos de los aspectos más importantes que definen y caracterizan a los *mockups* son:

- ✓ Muestran la sensación y la textura del diseño que tendría un producto final.

- ✓ Están a un paso por encima de los *wireframes*, generalmente son de fidelidad media-alta, pero sigue siendo un diseño estático (Cao, 2015)
- ✓ Pueden ser muy simples, sirven para ayudar a la presentación de las interacciones entre el usuario y el sistema, o más detalladas con gráficos ricos, siempre que se resalten restricciones específicas en la interfaz gráfica del usuario. Una de las herramientas de *software* más importantes para el desarrollo de *mockups* que nos presenta la guía es la maqueta de pantalla (Gliffy, 2007).

## 2.2. Profundización de conceptos de interés

En este apartado se presenta la profundización de los conceptos más importantes para el trabajo de grado, junto con los trabajos relacionados que surgen luego de ejecutar una revisión documental.

### 2.2.1.1. Elementos o partes de los requisitos

En la norma ISO/IEC/IEEE 29148 (2011), se establece que, para respaldar el análisis de requisitos, los requisitos bien formados deben tener atributos descriptivos definidos para ayudar a comprenderlos y administrarlos. Dentro de los establecidos por la norma se encuentran:

**Tabla 1.** *Atributos descriptivos para el análisis de requisitos.*

<b>Identificación</b>	Los identificadores únicos ayudan en el rastreo del requisito, pueden indicar vínculos o relaciones con otros requisitos. Una vez asignada, esta nunca cambia, así el requisito presente algún cambio o se elimine, y tampoco se reutiliza en otro.
<b>Prioridad</b>	Se debe establecer una prioridad a cada requisito, esto puede realizarse a través de un consenso entre los <i>stakeholders</i> , puede ser, p.e., una escala del 1-5, o un esquema simple alto, medio o bajo. Esto no implica que algunos requisitos no sean necesarios, pero es útil para establecer requisitos candidatos en el momento de tomar decisiones según la necesidad de las partes interesadas.
<b>Dependencia</b>	Esta solo se define cuando existe una dependencia entre requisitos, algunos de estos pueden tener una prioridad baja desde la perspectiva de una de las partes interesadas, sin embargo, son esenciales para el éxito del sistema. P.e., un requisito para medir la temperatura ambiente externa podría ser esencial para brindar soporte a otros requisitos como el mantenimiento de la



	temperatura interna de la cabina. Esta relación debe identificarse de modo que, si se elimina el requisito principal, también se pueda eliminar el requisito de soporte.
<b>Riesgo</b>	Pueden usarse técnicas de análisis de riesgo para calificar o clasificar los requisitos, en términos de su consecuencia o grado en que estos evitan o promueven riesgos. Los riesgos para considerar están relacionados con pérdidas financieras, potenciales oportunidades de pérdidas de negocio, pérdida de confianza de los <i>stakeholders</i> , impacto ambiental, problemas de seguridad, salud y normas o leyes nacionales.
<b>Fuente</b>	Cada requisito debe incluir un atributo que indique su origen. Múltiples fuentes pueden considerarse creadores de un requisito. La identificación de las fuentes para cada requisito permite identificar a qué organizaciones consultar para gestionar, aclarar, modificar o eliminar a estos. El concepto de propiedad está relacionado con la fuente, por lo tanto, se aplica al origen de un requisito; y para los requisitos de las partes interesadas, la parte interesada que los emite es quien tiene la propiedad. Como los requisitos se delegan aún más a través de la asignación y derivación, la responsabilidad de cumplir con estos se transfiere al equipo de producto apropiado, pero la fuente será la misma.
<b>Justificación</b>	La justificación indica la razón por la que se necesita el requisito y puede ser respaldada por cualquier análisis de apoyo, estudio comercial, modelado, simulación u otra evidencia objetiva que permita soportar esta justificación.
<b>Dificultad:</b>	Se debe tener en cuenta la dificultad asumida para cada requisito, p.e.: Fácil / Normal / Difícil. Esto proporciona un contexto adicional en términos de amplitud de requisitos y asequibilidad. También ayuda con el modelado de costos.
<b>Tipo:</b>	Los requisitos varían en intención y en los tipos de propiedades que representan, esto ayuda a ordenarlos en grupos para su análisis y asignación.

Fuente: ISO/IEC/IEEE 29148 (2011).

### 2.2.1.2. Elementos o partes de los Requisitos No Funcionales

Debido a que el trabajo está enfocado en RNF, se tienen en consideración las siguientes definiciones para caracterizar este tipo de requisitos:

Según (Franch, 1998), los RNF podrían caracterizarse de las siguientes maneras: atributo no funcional, comportamiento no funcional de la implementación de un componente y requisito no funcional.

### Características:

- **Dominio:** pertenecen a un conjunto de valores y operaciones válidas.
- **Tipos:** se clasifican en básicos o derivados. Los derivados dependen de uno o varios atributos No Funcionales (NF) básicos, quien o quienes definen si este RNF derivado se cumple, es decir, en el caso de atributos no funcionales básicos, se les asignará valores de cumplimiento o no y en el caso de los derivados los valores se calcularán automáticamente según el cumplimiento de los básicos.
- **Alcance:** determina los componentes en los que este se está usando.
- **Relaciones:** pueden vincularse a componentes completos o a operaciones individuales.
- **Múltiples definiciones:** Se pueden encontrar varias definiciones en distintos proyectos, por lo que se establece una sola definición particular que se utilice en cada ámbito donde parezca el atributo.

La noción típica que se tiene de los Requisitos No Funcionales es defectuosa, por lo cual se presenta una nueva clasificación de requisitos basada en cuatro facetas: (i) de **tipo** que puede ser de función, rendimiento o restricción, (ii) de **representación** ya sea operativa, cuantitativa o cualitativa, (iii) de **satisfacción**, dura o blanda, y (iv) de **función prescriptiva o supuesta**.

### Características: representación

- **Tipo:** Pueden ser de rendimiento, calidad específica o restricción.
- **Representación:** operacional, cuantitativa, cualitativa y declarativa. Para todos los funcionales se representan de forma operativa, esto es, por medio de la operación en sí. Los requisitos de rendimiento se representan de forma cuantitativa, es decir, pueden ser medibles. Cualitativas, por medio de cumplimiento de objetivos (comerciales, usabilidad.) y, por último, declarativa que se hace en restricciones, pues solo describe una situación requerida (se ejecutará en plataforma *linux*)
- **Satisfacción:** Se mide en el momento de verificar un requisito, es decir, al determinar si el requisito se cumple o no se cumple. Además, cuando se examinan los criterios que ayudan a saber si realmente se cumple con lo que se requería, se presentan dos casos: el primero es el de requisitos que se cumplen completamente o no (caso de los Requisitos

Duros); el segundo es el de los que se pueden cumplir de manera gradual (Requisitos Blandos).

- **Rol:** Un requisito puede desempeñar tres funciones en una especificación de requisitos: especificar propiedades del futuro sistema, indicar hechos o reglas en el entorno del sistema que influyen en el diseño y la implementación del sistema, y especificar cómo debe comportarse un actor en el entorno del sistema cuando interactúa con el sistema.

### 2.2.1.3. Elementos o partes de los patrones de análisis, diseño y requisitos

Para profundizar acerca del concepto de patrones, se definen los elementos más relevantes de cada tipo de patrón: análisis, diseño y requisitos. Por cada tipo se hizo una revisión documental y se encontraron diferentes propuestas; de cada propuesta consultada se extrajo los componentes o partes que presenta el autor con el fin de poder realizar un análisis que permita llegar a una caracterización de patrón como resultado.

Seguidamente, se describen de manera sintetizada cada una de estas propuestas por cada tipo de patrón, en términos de **contexto, problema, solución, partes y área de aplicación**. Donde **contexto, problema, solución** serían los tres elementos generales o esenciales que describen un patrón, **partes** que se refiere a los elementos que describen los aspectos específicos de la estructura de un patrón y **área de aplicación** que describe el entorno donde se utiliza o se desenvuelve el patrón.

#### 2.2.1.3.1. Patrones de análisis (Fowler, 1997)

**Contexto:** Se centra en sistemas comunes, es decir sistemas que solucionan problemas en los tipos de negocio habituales, bien sea salud, finanzas, entre otros.

**Problema:** Un analista comúnmente se enfrenta a situaciones en las cuales especificar requerimientos suele ser en algunos casos complejo, debido a que el estado inicial de conocimiento sobre el modelo de negocio de los proyectos en cuestión puede ser muy básico y es el cliente el experto. En la mayoría de los casos el cliente no transmite la totalidad de información, así este piense que sí, la información obtenida puede ser incompleta o prestarse para interpretar erróneamente y llevar a una aplicación que en realidad no responde a los requerimientos del cliente.

**Solución:** Como solución a este inconveniente se tiene una serie de modelos conceptuales, hechos de la manera más general sobre los tipos de negocio más habituales. Dichos modelos sirven de apoyo para que un analista pueda entender los distintos contextos donde se desenvuelve el *software* a desarrollar. Teniendo este apoyo, el analista puede validar con el cliente quien es el experto en el campo.

**Área de aplicación:** Software.

**Partes del patrón de análisis:**

- **Nombre del patrón:** Este refleja el significado de lo que se está considerando y a su vez debe ser simple para una fácil referencia en el análisis.
- **Intención:** Describe el objetivo del patrón y describe el problema que trata de resolverse.
- **Motivación:** Ilustra el problema y menciona como el patrón ayuda a resolverlo.
- **Fortalezas y contexto:** Discusión de las fortalezas y tensiones posibles que deben ser resueltas por el patrón.
- **Solución:** Describe la solución del problema por medio del patrón de análisis.
- **Consecuencias:** Habla sobre las limitaciones que puede tener el patrón de análisis al resolver el problema.
- **Diseño:** Parte en la que se selecciona las implementaciones de diseño en el patrón.

**Usos:** Apartado que menciona ejemplos del mundo real en el cual se aplica el patrón.

### 2.2.1.3.2. Patrones de diseño propuestas

Al realizar la búsqueda sobre patrones de diseño en la literatura, se tomaron como referencia ocho fuentes, cada una de estas divide en secciones o partes a cada patrón, de acuerdo con una plantilla, la cual da una estructura uniforme a la información, haciendo que los patrones de diseño sean más fáciles de aprender, comparar y usar. Las propuestas encontradas se describen de la siguiente manera, teniendo en cuenta los elementos definidos anteriormente en la sección de patrón:

#### 1. Título: “Pattern-Oriented Software Architecture” (Buschmann, 2007)

**Contexto:** se explica en profundidad cómo se usan los patrones para la arquitectura de software y cómo ellos ayudan a construir software, centrándose en componentes de subsistemas.

**Problema:** los diseños de software son poco flexibles ante cambios y solicitudes de los clientes, además es complejo entender algunos principios de diseño para aplicar a problemas con una descripción complicada y en variados vocabularios.

**Solución:** se enfoca en describir una estructura general entre componentes, clases u objetos, y detalla sus responsabilidades y relaciones, así como su cooperación. Todos los componentes juntos resuelven el problema que aborda el patrón de manera más efectiva y elegante que en un solo componente. Además, proporciona un común vocabulario y comprensión de los principios de diseño.

**Partes:**

- **Nombre:** el nombre y un breve resumen del patrón.
- **También conocido como:** otros nombres para el patrón, si se conocen.
- **Ejemplo:** un ejemplo del mundo real que demuestra la existencia del problema y la necesidad del patrón. A lo largo de la descripción, se refiere el ejemplo para ilustrar los aspectos de solución e implementación, donde esto es necesario o útil.
- **Contexto:** las situaciones en las que puede aplicarse el patrón.
- **Problema:** el problema que aborda el patrón, incluida una discusión sobre sus fuerzas asociadas.
- **Solución:** El principio de solución fundamental subyacente al patrón.
- **Estructura:** una especificación detallada de los aspectos estructurales del patrón, incluidas las tarjetas CRC para cada componente participante y un diagrama de clase OMT.
- **Dinámica:** escenarios típicos que describen el comportamiento en tiempo de ejecución del patrón. Ilustramos más a fondo los escenarios con gráficos de secuencia de mensajes de objeto.
- **Implementación:** Pautas para implementar el patrón. Estas son solo una suposición, no una regla inmutable. Se considera adaptar la implementación para satisfacer sus necesidades, agregando diferentes adicionales o pasos más detallados, o reordenando los pasos, a menudo describiendo detalles del problema de ejemplo.

- **Ejemplo resuelto:** Discusión de cualquier aspecto importante para resolver el ejemplo que aún no está cubierto en las secciones de Solución, Estructura, Dinámica e Implementación.
- **Variantes:** una breve descripción de variantes o especializaciones de un patrón.
- **Usos conocidos:** ejemplos del uso del patrón, tomado de los sistemas existentes.
- **Consecuencias:** los beneficios que proporciona el patrón y cualquier posible responsabilidad.
- **Visto también:** referencias a patrones que resuelven problemas similares, y a patrones que nos ayudan a refinar el patrón que estamos describiendo.

## 2. Título: “Modelos y Patrones” (Rosanigo, 2000)

**Contexto:** Se centra en los subsistemas software, más específicamente en las fases de análisis y diseño.

**Problema:** Selección ineficiente de arquitecturas en fases tempranas del desarrollo, donde se presentan malas relaciones entre componentes.

**Solución:** Proveer un esquema que sirva para la guía en la selección de una arquitectura idónea para el sistema que se está construyendo y que se vea reflejado en una buena comunicación entre componentes.

### Partes:

- **Nombre:** Debe tener un nombre significativo. Sería muy incontrolable tener que describir el patrón cada vez lo utilizamos en una discusión.
- **Intención:** Descripción sucinta de cuál es el problema que resuelve.
- **Descripción del problema:** Describe cuándo aplicarlo, explica el problema y su contexto y la lista de precondiciones que deben encontrarse, si las hubiera.
- **Solución:** Describe los elementos que lo componen: clases, objetos, relaciones, responsabilidades y colaboraciones.
- **Consecuencias:** Describe los costos y beneficios de aplicarlo. Incluye el impacto sobre la flexibilidad, extensibilidad y portabilidad del sistema.

### 3. **Título: Patrones de diseño: elementos de software orientado a objetos reutilizables** (Gamma, 1995)

**Contexto:** En el software orientado a objetos, se encuentran muchos problemas específicos de diseño, en los cuales se necesitan implementar buenas soluciones.

**Problema:** Software poco flexible con lenguajes poco conocidos y descripciones complejas.

**Solución:** Crear software orientado a objetos, con diseños más flexibles, con un lenguaje estándar, elegantes y reutilizables, basado en experiencias previas y haciendo más fácil reutilizar buenos diseños y arquitecturas.

#### **Partes:**

- **Nombre del patrón:** El nombre del patrón transmite la esencia del patrón sucintamente. Un buen nombre es fundamental porque se convertirá en parte de su vocabulario de diseño.
- **Propósito:** Una breve declaración que responde a las siguientes preguntas: ¿Qué hace el patrón de diseño?, ¿Cuál es su justificación e intención?, ¿Qué problema(s) de diseño en particular aborda?
- **También conocido como:** Otros nombres conocidos para el patrón, si los hay.
- **Motivación:** Un escenario que ilustra un problema de diseño y cómo las estructuras de clase y objeto en el patrón resuelven el problema. El escenario lo ayudará a comprender la descripción más abstracta del patrón que sigue.
- **Aplicabilidad:** ¿En qué situaciones se puede aplicar el patrón de diseño? ¿Cuáles son ejemplos de diseños pobres que el patrón puede abordar? ¿Cómo puedes reconocer estas situaciones?
- **Estructura:** Una representación gráfica de las clases en el patrón utilizando una notación basada en la Técnica de Modelado de Objetos (OMT) [RBP + 91]. También se utiliza diagramas de interacción [JCJO92, Boo94] para ilustrar secuencias de solicitudes y colaboraciones entre objetos. El Apéndice B describe estas anotaciones en detalle.
- **Participantes:** Las clases y / u objetos que participan en el patrón de diseño y sus responsabilidades.
- **Colaboraciones:** Cómo colaboran los participantes para llevar a cabo sus responsabilidades.

- **Consecuencias:** ¿Cómo apoya el patrón sus objetivos?, ¿Cuáles son las compensaciones y resultados de usar el patrón?, ¿Qué aspecto de la estructura del sistema te permite variar de forma independiente?
- **Implementación:** responde a las preguntas: ¿Qué trampas, sugerencias o técnicas se deben tener en cuenta cuando se está implementando el patrón?, ¿Hay problemas específicos del lenguaje?
- **Usos conocidos:** Ejemplos del patrón encontrado en sistemas reales. Incluimos al menos dos ejemplos de diferentes dominios.
- **Patrones relacionados:** ¿Qué patrones de diseño están estrechamente relacionados con éste?, ¿Cuáles son las diferencias más importantes?, ¿Con qué otros patrones deberían usarse estos?

#### 4. Título: “Strategies, Patterns and Applications” (Coad, 1995)

**Contexto:** Sistemas orientados a objetos donde se necesitan que estos sean, de alguna manera, genéricos y flexibles; ilustrando desde la perspectiva de otras áreas que aplican patrones, para emplear conocimiento y enseñanzas provenientes de obras, situaciones o libros pasados.

**Problema:** Software acoplado y específico solo para la solución del producto en desarrollo, en el cual dicha esta solución no puede ser utilizada en un futuro y donde el software creado es difícil de modificar y mantener.

**Solución:** Propone una abstracción formada por un pequeño grupo de clases que resulta ser útil una y otra vez en el desarrollo orientado a objetos, es decir, los define como una plantilla de objetos que interactúan.

##### Partes:

- **Nombre del patrón y clasificación:** El nombre del patrón transmite la esencia del patrón sucintamente. Un buen nombre es fundamental, porque se convertirá en parte de su vocabulario de diseño.
- **Intención:** Una breve declaración que responde a las siguientes preguntas: ¿Qué hace el patrón de diseño?, ¿Cuál es su justificación e intención?, ¿Qué problema o problema de diseño en particular aborda?
- **También conocido como:** Otros nombres conocidos para el patrón, si los hay.



- **Motivación:** Un escenario que ilustra un problema de diseño y cómo las estructuras de clase y objeto en el patrón resuelven el problema. El escenario lo ayudará a comprender la descripción más abstracta del patrón que sigue.
- **Aplicabilidad:** ¿En qué situaciones se puede aplicar el patrón de diseño?, ¿Cuáles son ejemplos de diseños pobres que el patrón puede abordar?, ¿Cómo puedes reconocer estas situaciones?
- **Estructura:** Una representación gráfica de las clases en el patrón utilizando una notación basada en la Técnica de Modelado de Objetos (OMT) [RBP + 91]. También, utilizamos diagramas de interacción [JCJO92, Boo94] para ilustrar secuencias de solicitudes y colaboraciones entre objetos. El Apéndice B describe estas anotaciones en detalle.
- **Participantes:** Las clases y / u objetos que participan en el patrón de diseño y sus responsabilidades.
- **Colaboraciones:** Cómo colaboran los participantes para llevar a cabo sus responsabilidades
- **Consecuencias:** ¿Cómo apoya el patrón sus objetivos?, ¿Cuáles son las compensaciones y resultados de usar el patrón?, ¿Qué aspecto de la estructura del sistema te permite variar de forma independiente?
- **Implementación:** Son sugerencias o técnicas que se deben tener en cuenta cuando se está implementando el patrón. Incluye la descripción de los problemas específicos del lenguaje.
- **Usos conocidos:** Ejemplos del patrón encontrado en sistemas reales. Se incluyen al menos dos ejemplos de diferentes dominios.
- **Patrones relacionados:** ¿Qué patrones de diseño están estrechamente relacionados con éste?, ¿Cuáles son las diferencias importantes?, ¿Con qué otros patrones deberían usarse éstos?

**Área de aplicación:** Software

## 5. Título: “Smalltalk Best Practice Patterns” (Beck, 1997)

**Contexto:** Patrones de diseño aplicados a Sistemas de Información en el modelado de objetos y arquitectura de software.

**Problema:** Software poco flexible y reutilizable. Siempre habrá nuevos dominios para modelar, pero siguen siendo las mismas estructuras de objetos. Siguen apareciendo una y otra vez, independientemente de la aplicación, lo que se puede aprovechar al usar patrones.

**Solución:** Se propone una base más flexible para producir variaciones sistemáticas en los temas comunes de la ingeniería de software, incluyendo cómo reconocer la presencia del problema y cómo generar la solución para que se ajuste al contexto.

**Partes:**

- **Título:** es un texto creado con el fin de identificar el resultado de ejecutar el patrón, diseñado para ser utilizado de manera conversacional para proporcionar una identidad y referirse a este.
- **Patrones anteriores:** describe brevemente qué patrón precede a este y por qué. Algunos de estos patrones son el resultado de la experiencia sobre qué problemas son los más importantes y cuáles pueden diferirse de manera segura.
- **Problema:** se declara como una pregunta. Leer el problema le dirá rápidamente si está interesado en el resto del patrón. Aparece en negrita.
- **Fuerzas:** describe las restricciones sobre cualquier solución al problema y argumenta a favor de la solución requerida en este patrón. Después de leer la sección de fuerzas, debe estar convencido de que la solución es válida para su situación, o debe saber por qué la solución no es válida. A veces toma la forma de una descripción de alternativas y por qué no funcionan, a veces solo una discusión de las influencias en la decisión de diseño.
- **Solución:** le da una receta inequívoca y concreta para crear el objeto nombrado en el título del patrón. Por lo general se trata de acciones del entorno de desarrollo: subclase esto, nombre de esta variable tal y cual, entre otros etc. Aparece en negrita.
- **Discusión:** explica cómo hacer un uso práctico del patrón. Puede contener un ejemplo de uso o problemas a tener en cuenta.
- **Siguientes patrones:** indica qué patrones se deben considerar en continuación.

**6. Título: “A Confederation of Patterns for Resource Management” (Braga, 1998)**

**Contexto:** Sistemas software para pequeñas y medianas empresas, donde se establecen los recursos que se van a tener en cuenta, como clientes, vendedores, entre otros. Estos sistemas

tienen mucho en común, pues, aunque la lógica de negocio sea distinta, presentan ciertos componentes junto con las relaciones que hay entre ellos.

**Problema:** La naturaleza del sistema, la identificación de los recursos y las relaciones que existen entre estos es un poco compleja de especificar, pues se están desaprovechando soluciones a problemas de la misma naturaleza presentes en sistemas que se han construido con anterioridad.

**Solución:** Maximizar la reutilización del análisis, el diseño y el código de estos sistemas simples construidos con anterioridad, por medio de un conjunto de estrategias que permitan utilizar un análisis y conocimiento adquirido en etapas anteriores.

**Partes:**

- **Problema:** describe una situación que se presenta, para la cual el patrón pueda resultar idóneo y aportar a su solución. Este problema se presenta de manera clara y concisa.
- **Contexto:** se resalta el medio donde se presenta el problema, para ello se realiza una breve descripción de dónde se está presentando el problema y bajo qué situación.
- **Fuerzas:** define los factores que influyen en el problema, describiendo estos detalladamente para tenerlos en cuenta en una futura solución; así mismo se detalla los puntos límites a los que puede aplicarse este patrón.
- **Solución:** presenta una aplicación del patrón al problema bajo el contexto especificado, es decir, cómo este patrón sería de ayuda para encontrar o apoyar a una solución.
- **Ejemplos:** se describe una situación en la cual pueda ser utilizado el patrón, como también se ilustra con un diagrama de clases los elementos que están implicados en el contexto y que de alguna manera influyen en la forma que se da la solución. Se dan varios ejemplos de situaciones y se detallan cada una de estas.
- **Usos conocidos:** se describen más enfoques en los que este patrón puede apoyar a una solución.
- **Patrones relacionados:** muestra patrones que pueden ser utilizados en situaciones de esta misma naturaleza y de qué forma lo pueden hacer.

**Área de aplicación:** Software.

**7. Título: “Introducción a los patrones de Diseño” (Blancarte, 2016)**

**Contexto:** Sistemas de información donde las buenas prácticas de programación pueden aumentar en gran medida la calidad del desarrollo aprovechando soluciones planteadas por expertos anteriormente.

**Problema:** Reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.

**Solución:** Catálogo de soluciones documentadas y probadas de diseño para problemas comunes conocidos.

**Partes:**

- **Introducción:** Se explica de forma teórica en qué consiste el patrón de diseño, su estructura, componentes y cuándo deberíamos utilizarlo.
- **Escenario:** Se expone un escenario de la vida real en el cual podríamos utilizar el patrón de diseño para después solucionarlo.
- **Solución:** Se intenta resolver el escenario planteado utilizando patrones de diseño.
- **Implementación:** En esta sección se utilizará el código para programar la solución propuesta.
- **Ejecución:** Se ejecuta el código desarrollado para comprobar los resultados tras implementar el patrón de diseño.
- **Siguientes pasos:** Breve resumen de lo aprendido y se plantean una serie de ejercicios para resolver con la finalidad de afianzar los conocimientos adquiridos.

**8. Título: “Patrones de diseño: Construyendo aplicaciones flexibles y reutilizables”**  
(Yepes y Hurtado, 2015)

**Contexto:** Sistemas orientados a objetos cuyo diseño sea reutilizable.

**Problema:** Aplicaciones informáticas mal construidas que presentan síntomas como la rigidez (difícil de cambiar), fragilidad (realizar un cambio en alguna parte ocasionan cambios en otra) y la inmovilidad (inhabilidad para reutilizar).

**Solución:** Se buscan conseguir diseños generales y flexibles donde se tenga la capacidad de reutilizar soluciones que han funcionado en el pasado, aprovechando la experiencia, describiendo los elementos que componen el diseño sus relaciones, responsables y colaboraciones.

**Partes:** Nombre del patrón, Problema, Solución, Consecuencias.

**Área de aplicación:** Software.

### 2.2.1.3.3. Patrones de requisitos

Al realizar la búsqueda sobre patrones de requisitos en la literatura se tomaron como referencia dos fuentes, las cuales se describen a continuación:

#### 1. Título: “Software requirement patterns” (Withall, 2007)

**Contexto:** requisitos similares en naturaleza, por ejemplo: varias funciones de consulta, varias funciones de visualización.

**Problema:** complejidad en el momento de especificar requisitos y requisitos poco entendibles en diferentes formatos.

**Solución:** se introduce el concepto de patrón, con el fin de especificar todos los requisitos de un tipo de manera coherente, evitando esfuerzo en requisitos cuya naturaleza sea similar.

#### Partes:

- **Detalles básicos:** la manifestación del patrón, dominio propio, patrones relacionados (si los hay), frecuencia de uso anticipada, clasificaciones de patrones y autor del patrón.
- **Aplicabilidad:** describe las situaciones en las que se puede aplicar el patrón y en cuáles no.
- **Discusión:** detalla cómo escribir un requisito de este tipo y que debe tener en cuenta un requisito de este tipo.
- **Contenido:** especifica que debe un requisito de este tipo, incluyendo detalles adicionales. Esta es la sustancia principal del patrón.
- **Plantilla (s):** un punto de partida para escribir un requisito de este tipo, o más de uno si hay formas alternativas distintas.
- **Ejemplo (s):** uno o más requisitos representativos escritos usando este patrón.
- **Requisitos adicionales:** describe que tipo de requisitos se derivan de un requisito de este tipo.
- **Consideraciones para el desarrollo:** consejos para diseñadores e ingenieros de software sobre cómo implementar un requisito de este tipo.

- **Consideraciones para la prueba:** especifica qué se debe tener en cuenta al probar este tipo de requisito.

## 2. Título: “Expressing Customer Requirements Using Natural Language Requirements Templates and Patterns” (Duran, 1999)

**Contexto:** especificación de requisitos que tienen cierta similitud.

**Problema:** problemas de alcance, esto es, decidir el límite del sistema y evitar información innecesaria, problemas de comprensión entre las comunidades involucradas en el proceso y problemas de volatilidad, pues los requisitos evolucionan con el tiempo. De igual manera brindar conocimiento sobre requerimientos de la misma naturaleza ya especificados anteriormente.

**Solución:** se usa el concepto de plantilla y patrón para evitar el sobre esfuerzo en la especificación de requerimientos anteriormente especificados, que poseen cierta familiaridad en cuanto a su naturaleza.

### Partes:

- **Identificador:** cada requisito debe identificarse de manera única mediante un número.
- **Nombre descriptivo:** ayuda a la identificación rápida del requisito (Comienza con RI)
- **Versión:** siguiendo las recomendaciones de IEEE, se deben gestionar diferentes versiones de los requisitos. Este campo contiene el número de versión actual y la fecha del requisito.
- **Autor, Fuente:** en estos campos deben ir el nombre y la organización del autor y la fuente de la versión actual del requisito.
- **Propósito:** debe indicar por qué el requisito es necesario para lograr los objetivos comerciales.
- **Descripción:** para los requisitos de almacenamiento de información, este ítem debe completarse con el concepto relevante sobre la información que debe almacenarse.
- **Datos específicos:** lista de datos específicos asociados con el concepto relevante.
- **Intervalo de tiempo:** indica cuánto tiempo la información sobre el concepto es relevante para el sistema. Puede tomar dos valores: pasado-presente y solo presente. P.e.: si el concepto es empleado, un intervalo de tiempo pasado y presente significa que los

empleados son relevantes para el sistema; un intervalo de tiempo presente solo significa que los empleados no están bajo consideración.

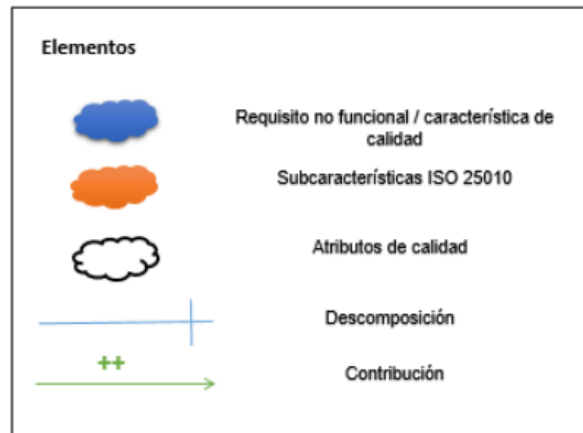
- **Importancia, urgencia:** este campo da cuenta de la importancia / urgencia del requisito para clientes y usuarios. Se le puede asignar un valor numérico o algunas expresiones enumeradas como de vital importancia, de inmediato, bajo presión o puede esperar la urgencia.
- **Comentarios:** aquí se puede registrar otra información que no se puede ajustar en campos anteriores.

### 2.3. Características de los RNF de fiabilidad y mantenibilidad

La caracterización se entiende como el hecho que determina los rasgos de una entidad, de manera que esta se logre distinguir de los demás. La caracterización de los requisitos de fiabilidad y mantenibilidad se hará desde una perspectiva de la literatura existente, con el fin de poder establecer los elementos constitutivos básicos, tanto de fiabilidad como de mantenibilidad, junto con los aspectos más importantes que los distinguen. Dichos RNF se caracterizan a la luz de dos trabajos previos fundamentales, se estructura la información de estos RNF a manera de grafos, buscando describir de manera clara las características, sub características y atributos de calidad. Además, para la mantenibilidad se ha considerado también el ejercicio hecho por (Rodríguez & Piattini, 2014), donde además de tener en cuenta los atributos de calidad, se incluye a su vez métricas para medir tales atributos. Seguidamente, se hace un cruce entre estos y las características, con el objetivo de aportar una mayor visibilidad o claridad a la mantenibilidad.

Para comprender las ilustraciones que se van a presentar en los grafos es necesario tener en cuenta las convenciones o elementos para los grafos de atributos de calidad (ver ilustración 1).

### Ilustración 1. Convecciones o elementos para los grafos de atributos de calidad



Fuente: Arias & Rosero, 2021

En seguida se muestra la información obtenida para cada RNF, basándose en los grafos de atributos de calidad.

#### 2.3.1. Elementos o atributos de fiabilidad

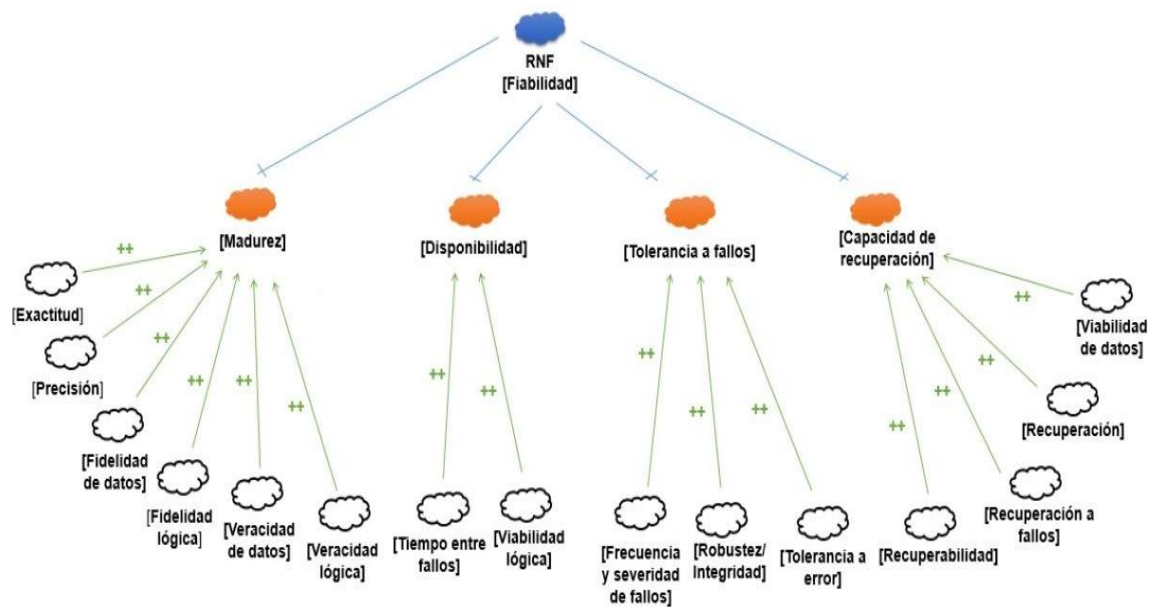
Para McCall (2010), la fiabilidad es el grado de cumplimiento de las funciones de un sistema y la divide en las sub características de tolerancia al error, consistencia, exactitud, simplicidad, madurez y recuperabilidad. De igual manera, Bohem's (1978), la describe como la medida en la que el código realiza sus funciones dividiéndola en las sub características de autocontención, exactitud, completitud y robustez / integridad. También para Gilb (1987), la fiabilidad es el grado en el que el sistema hace lo que debería hacer y divide esta característica en fidelidad lógica, veracidad lógica, viabilidad lógica, fidelidad de datos, veracidad de datos. De esta manera, muchos otros autores describen la fiabilidad en términos semejantes, los cuales son abordados en su totalidad por la ISO 25010 (2013), quien define la fiabilidad como la capacidad de un sistema para desempeñar funciones específicas cuando se usa bajo unas condiciones y periodo de tiempo determinados y divide esta característica en:

- **Madurez:** Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
- **Disponibilidad:** Capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.



- **Tolerancia a fallos:** Capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.
- **Capacidad de recuperación:** Capacidad del producto software para recuperar los datos directamente afectados y restablecer el estado deseado del sistema en caso de interrupción o fallo.

**Ilustración 2.** Grafo detallado de fiabilidad



Fuente: Arias & Rosero, 2021

El anterior grafo, muestra una organización estructural de la información obtenida para la fiabilidad. En el último nivel (el más bajo) se refleja los atributos de calidad pertenecientes a cada sub característica. La fiabilidad en este caso definida a continuación, en términos de subcaracterística y los atributos de calidad que las componen a estas.

**Sub característica (Nivel 2):**

**Madurez:** Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.

### **Atributos de calidad nivel 3 de madurez:**

- **Exactitud:** es un atributo del software, que hace referencia al hecho de proporcionar la precisión requerida en los cálculos y resultados.
- **Precisión:** se refiere a si el valor de una variable o constante es suficiente para alcanzar la exactitud requerida en los cálculos en los que interviene.
- **Fiabilidad de datos:** mide la precisión con la que se representa un concepto en un entorno hardware y software determinado.
- **Fidelidad lógica:** mide la precisión con la cual se ha implementado un algoritmo dado para un entorno hardware y software específico.
- **Veracidad lógica:** mide la adecuación con la que la implementación de un algoritmo se relaciona con el mundo real, con el que debe interactuar.
- **Veracidad de datos:** mide la adecuación con la que los datos representan al mundo real.

### **Sub característica (Nivel 2):**

**Disponibilidad:** capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.

### **Atributos de calidad nivel 3 de disponibilidad:**

- **Tiempo entre fallos:** describe el lapso entre un fallo y su siguiente acontecimiento.
- **Viabilidad lógica:** mide lo bien que el sistema cumple con las restricciones de diseño en áreas como la velocidad de ejecución, requisitos de espacio o seguridad.

### **Sub característica (Nivel 2):**

**Tolerancia a fallos:** capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.

### **Atributos de calidad nivel 3 de Tolerancia a fallos:**

- **Frecuencia y severidad de fallos:** describe con qué tasa de frecuencia ocurren fallos en el sistema y la severidad de cada fallo desde el punto de vista del producto.
- **Tolerancia a error:** atributos de software que proporcionan continuidad de operación en condiciones normales.

- **Robustez / Integridad:** capacidad del producto software de reaccionar apropiadamente ante condiciones excepcionales.

### Sub característica (Nivel 2):

**Capacidad de recuperación:** capacidad del producto software para recuperar los datos directamente afectados y restablecer el estado deseado del sistema en caso de interrupción o fallo.

#### Atributos de calidad nivel 3 de capacidad de recuperación:

- **Recuperabilidad:** el software que considera las acciones necesarias para recuperarse en caso de fallas de manera fácil y rápida.
- **Recuperación:** capacidad de restablecer un nivel específico de rendimiento y recuperar los datos directamente afectados en caso de falla.
- **Viabilidad de datos:** mide lo bien que los datos cumplen las restricciones de diseño en áreas como tiempos de recuperación de la información, espacio de almacenamiento o seguridad.

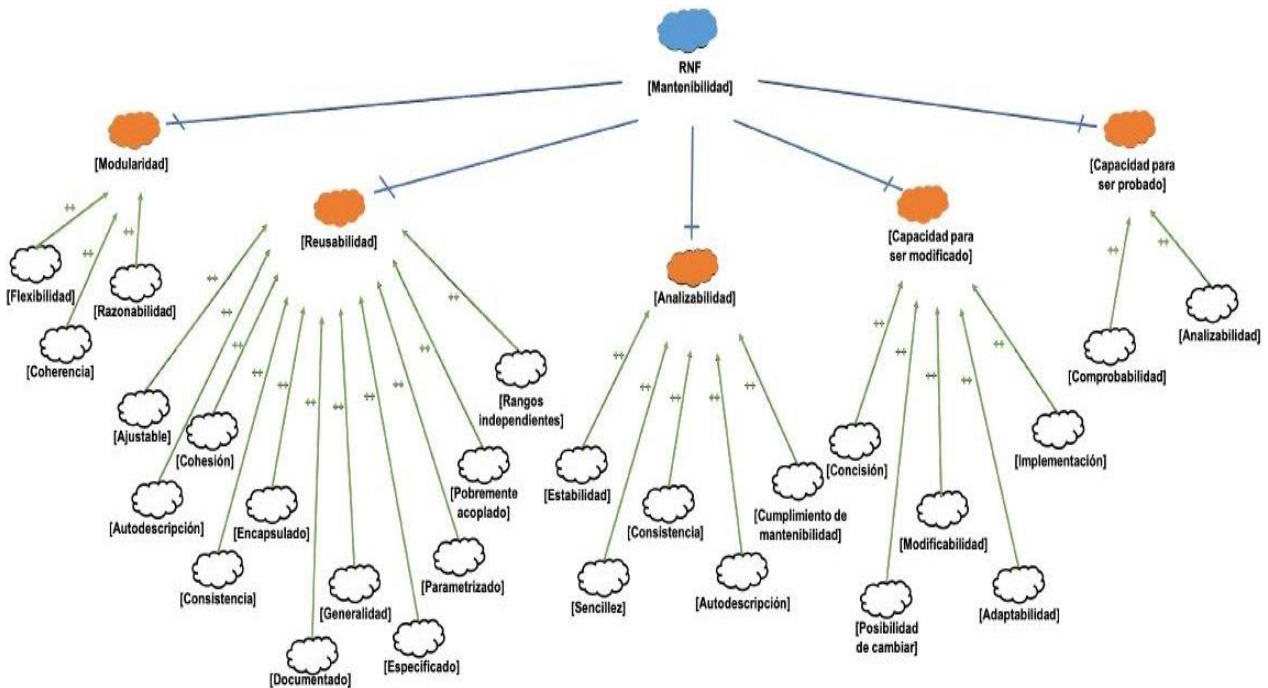
### 2.3.2. Elementos o atributos de mantenibilidad

La mantenibilidad se define como la capacidad del sistema de ser modificado efectiva y eficientemente debido a necesidades evolutivas, correctivas o perfectivas. Esta mantenibilidad a su vez incluye los siguientes atributos (ISO 25010):

- **Analizabilidad:** facilidad para identificar las partes de un sistema que se deben modificar debido a deficiencias o fallos, o la capacidad de evaluar el impacto que puede provocar un cambio en el sistema.
- **Modularidad:** grado en el que un sistema se encuentra dividido en módulos de forma que el impacto que causa una modificación en un módulo sea mínimo para el resto.
- **Capacidad de ser Modificado:** grado en el que se pueden realizar cambios en un producto software de forma efectiva y eficiente, sin introducir defectos ni degradar su rendimiento.
- **Capacidad de ser Reutilizado:** grado en que un activo (módulo, paquete, clase, etc.) puede ser usado en más de un sistema o en la construcción de otros activos.

- **Capacidad de ser Probado:** facilidad para establecer criterios de prueba para un sistema y realizar las pruebas que permitan comprobar que se cumplen esos criterios.

**Ilustración 3. Grafo detallado de mantenibilidad**



Fuente: Arias & Rosero, 2021

La ilustración 3 muestra estructuralmente la información obtenida para la mantenibilidad. En este se observa de arriba a abajo: característica (Nivel 1), sub características (Nivel 2), y atributos de calidad (Nivel 3).

### Sub característica (Nivel 2):

**Modularidad:** capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.

#### Atributos de calidad nivel 3 de Modularidad:

- **Flexibilidad** cambios que satisfagan nuevos requerimientos.
- **Reparabilidad:** corrección de defectos con una cantidad de trabajo limitada y razonable.

### Sub característica (Nivel 2):

**Reusabilidad:** capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.

#### Atributos de calidad nivel 3 de Reusabilidad:

- **Ajustable:** cuando se utiliza el mínimo número de variables con un único objetivo en la solución del problema. Deben tenerse en cuenta la no existencia de constantes no declaradas.
- **Auto descripción:** cuando el propósito, estrategia, intención y propiedades de cada elemento del software son claramente evidentes a partir de los nombres de los módulos y además los diferentes identificadores tienen sentido y son congruentes dentro del contexto de la aplicación.
- **Consistencia:** atributos del software que proporcionan diseño uniforme y técnicas de implementación y notación en todo el desarrollo del producto.
- **Documentado:** cuando se ha definido de forma explícita y precisa el propósito, estrategia, intención y propiedades de cada elemento del software.
- **Encapsulado:** si un nombre solo se utiliza en el ámbito en el que ha sido definido.
- **Especificado:** cuando la funcionalidad se describe mediante precondiciones y postcondiciones.
- **Generalidad:** un módulo es genérico si sus cálculos son independientes de los tipos de los datos.
- **Parametrizado:** se considera que un módulo está bien parametrizado cuando sus parámetros son todas las entradas y salidas necesarias y suficientes para determinar un módulo bien definido.
- **Pobremente acoplado:** cuando los datos de las llamadas a los módulos están acoplados únicamente al módulo llamado.
- **Rangos independientes:** cuando los límites inferiores y superiores de una estructura no son constantes numéricas o caracteres.
- **Cohesión:** cuando todos los elementos están vinculados a algún otro, contribuyendo todos a cumplir un objetivo del programa.

### **Sub característica (Nivel 2):**

**Analizabilidad:** facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar.

#### **Atributos de calidad nivel 3 de Analizabilidad:**

- **Estabilidad:** capacidad del producto software para permitir la implementación de una modificación específica.
- **Cumplimiento de mantenibilidad:** capacidad del producto software de adherirse a los estándares y convenciones relacionados con la mantenibilidad.
- **Sencillez:** atributos del software que proporcionan la implementación de funciones de la manera más comprensible (Por lo general evitar prácticas que aumenten la complejidad).
- **Consistencia:** atributos del software que proporcionan diseño uniforme y técnicas de implementación y notación, permitiendo un alto grado de comprensión en todo el software.
- **Auto descripción:** atributos del software que proporcionan una explicación de la implementación de una función, permitiendo una facilidad de lectura y comprensión.

### **Sub característica (Nivel 2):**

**Capacidad para ser modificado:** capacidad del producto que permite modificación de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.

#### **Atributos de calidad nivel 3 de Capacidad para ser modificado:**

- **Concisión** atributos del software que proporcionan la implementación de una función con una cantidad mínima de código.
- **Posibilidad de cambiar** capacidad del producto software para permitir la implementación de una modificación específica.
- **Modificabilidad** facilidad para incorporar cambios una vez que se ha determinado la naturaleza de los cambios deseados.

- **Adaptabilidad:** Capacidad del software para ser adaptado a diferentes entornos especificados, sin aplicar acciones que el propio software proporcione.

### **Sub característica (Nivel 2):**

**Capacidad para ser probado:** facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

#### **Atributos de calidad nivel 3 de Capacidad para ser probado:**

- **Comprobabilidad:** Capacidad del producto software para permitir que el software modificado sea válido.
- **Analizabilidad:** Capacidad del producto software para ser diagnosticado por deficiencias o causas de fallas en el software, o para que las partes sean modificadas para ser identificadas.

### **Aspectos importantes de la mantenibilidad**

Uno de los aspectos que cobran importancia de este RNF es que el mantenimiento supone una de las fases del ciclo de vida de desarrollo más costosa, sino la más, llegando a alcanzar el 60% de sobre costo. Por otra parte, la mantenibilidad es una de las características más demandadas hoy en día por los clientes de software, que piden que el producto desarrollado pueda ser después mantenido por ellos mismos o incluso por un tercero; y por último se dice que las tareas de mantenimiento sobre productos con poca mantenibilidad tienen más probabilidad de introducir nuevos errores en el producto.

Sin embargo, la familia de normas ISO/IEC 25000 todavía no ha definido el conjunto de métricas e indicadores que afectan a cada una de estas sub características, los umbrales para las mismas, ni las funciones de medición a aplicar para poder calcular el valor de calidad de cada una de ellas. Por ello, se han identificado un conjunto de propiedades de calidad, que obtienen su valor a partir de métricas del código fuente, y se ha establecido la relación que existe con las sub características anteriormente indicadas. El objetivo al identificar estas propiedades de calidad y métricas es lograr una caracterización de este RNF lo más detallada posible, permitiendo así agregar valor al diseño de este patrón propuesto. Estas propiedades de calidad son (Rodríguez & Piattini, 2014):

***Incumplimiento de Reglas.*** Esta propiedad se basa en el incumplimiento de reglas de programación y optimización de código establecidas por los estándares de programación de cada lenguaje y que se obtienen por varias herramientas de comprobación de reglas. Para cada sub característica de calidad se comprueba un conjunto de reglas relacionadas.

***Documentación del Código.*** La documentación de código hace referencia a los comentarios que existen en el código, los cuales se utilizan para explicar la funcionalidad del código. Los comentarios influyen en la Analizabilidad, ya que estos, bien expresados, ayudan a entender qué es lo que hace el código. Por esta misma razón, los comentarios también influyen en la capacidad para reutilizar el software: conocer qué es lo que hace un módulo, ayuda a reutilizarlo en otros sistemas.

***Complejidad.*** La propiedad Complejidad se relaciona con la dificultad para implementar, probar, entender, modificar o mantener un programa. Como esta definición muestra, la Complejidad está relacionada con las sub características: Analizabilidad, Capacidad de ser Modificado y Capacidad de ser Probado del modelo de calidad. En aplicaciones con complejidad elevada, las tareas de mantenimiento requieren más esfuerzo y, por tanto, son más costosas.

***Estructuración.*** La estructuración de un sistema es la división que se hace del mismo en componentes más pequeños. La calidad en el diseño de un sistema está estrechamente relacionada con esta propiedad: la estructuración correcta de un sistema en cualquiera de sus niveles (subsistemas, paquetes, clases, etc.) facilita su desarrollo y mantenimiento. Una correcta estructuración del sistema: facilita su comprensión y la identificación de elementos que se deben modificar (Analizabilidad), minimiza el impacto de los cambios que haya que realizar en el sistema (Modularidad), facilita la reutilización de únicamente aquellos elementos que sean necesarios, pudiendo utilizar toda la funcionalidad que se espera de ellos y no más o menos (Capacidad de ser Reutilizado) y facilita la elaboración de pruebas permitiendo que éstas se centren en comprobar el funcionamiento correcto de porciones concretas del sistema (Capacidad de ser Probado).

***Tamaño de Métodos.*** El tamaño de un sistema afecta directamente a la mantenibilidad del mismo, ya que, intuitivamente, un sistema más grande requiere más esfuerzo de mantenimiento. Se considera más adecuado evaluar el tamaño de sus elementos a un



nivel más bajo, como por ejemplo a nivel de métodos, que evaluar directamente el tamaño global del sistema, ya que una modificación normalmente no afecta a todo el sistema, sino a un conjunto de sus elementos.

**Código Duplicado.** Esta propiedad hace referencia a fragmentos de código que se encuentran repetidos en diferentes lugares del sistema. El código duplicado hace más difícil la realización de modificaciones en una aplicación software, ya que solucionar un defecto o introducir una mejora en dicho código requiere realizar las modificaciones en todas las partes del sistema en las que aparezca dicho código duplicado. El Código Duplicado afecta a las subcaracterísticas Analizabilidad y Capacidad de Ser Modificado, puesto que el código duplicado dificulta la identificación del código que se debe modificar y dificulta la realización de modificaciones que sean efectivas. También afecta a la Subcaracterística Capacidad de Ser Probado, ya que se deben definir varios casos de prueba diferentes (aunque similares en forma) para comprobar el correcto funcionamiento del código duplicado en sus distintas instancias.

**Acoplamiento.** El acoplamiento indica el grado de interdependencia entre las unidades de software (módulos, funciones, clases, bibliotecas, etc.). De forma general, cuanto más bajo sea el acoplamiento en una aplicación software, mejor se considera su diseño, debido a que el bajo acoplamiento permite mejorar la mantenibilidad (los cambios en una unidad no afectan al resto de unidades si no hay acoplamiento) y aumentar la reutilización de las unidades de software. El acoplamiento también dificulta la realización de pruebas, siendo más fácil probar elementos que dependen de pocos elementos externos (Spinellis, 2006).

**Balance Inestabilidad - Abstracción.** La inestabilidad indica la relación entre el acoplamiento de salida y el acoplamiento total de un paquete y da una idea de la resistencia al cambio del mismo. La abstracción indica la relación entre clases abstractas (e interfaces) y el número de clases de un paquete. Debe existir un balance entre ambas, de forma que los paquetes deben ser abstractos y estables o concretos e inestables.

**Ciclos.** La propiedad Ciclos hace referencia a la existencia de ciclos de dependencia entre los paquetes del sistema. El Principio de Dependencias Acíclicas (Martin, 1996) indica que no deben existir ciclos en la estructura de dependencias de un sistema. La existencia de Ciclos en un sistema afecta negativamente a su mantenibilidad dificultando la

realización de modificaciones, ya que un cambio en un paquete repercute a todos los paquetes en el ciclo de dependencias (Spinellis, 2006). Además, dificulta la reutilización del código y la realización de pruebas (Martin, 1996), pues si se quiere reutilizar un paquete o realizar pruebas sobre alguna de sus clases es necesario tener presentes todos los paquetes involucrados en el ciclo.

**Cohesión.** El concepto de cohesión (DeMarco, 1979) indica el grado de relación que existe entre los elementos internos de un módulo. Una clase tiene cohesión baja cuando realiza varias funciones no relacionadas. Así, las funcionalidades que proporciona mediante sus métodos tienen poco en común y para realizar sus actividades utiliza conjuntos de datos que no están relacionados. La Analizabilidad, la Modularidad y, en general, la Mantenibilidad en sistemas cuyas clases tienen una cohesión baja se ven perjudicadas, ya que estos sistemas son más difíciles de comprender, proporcionan funcionalidades que no son necesarias y su mala modularidad hace que los cambios en los requisitos afecten a varios módulos.

Teniendo en cuenta la organización anteriormente descrita por los grafos, el nivel 2 de este, es decir las sub características, fueron cruzadas con los aspectos o propiedades importantes de la mantenibilidad. Con esto, se muestra a continuación (ver tabla 2), cada propiedad y a cuál de las sub características afecta directa o indirectamente.

**Tabla 2.** Atributos de calidad vs subcaracterísticas de calidad

Propiedades / Subcaracterísticas	Analizabilidad	Modularidad	Capacidad de ser Modificado	Capacidad de ser Reutilizado	Capacidad de ser Probado
Incumplimiento de Reglas	X	X	X	X	X
Documentación del Código	X			X	
Complejidad	X		X		X
Balance Inestabilidad-Abstracción / Acoplamiento		X	X	X	X
Ciclos		X	X	X	X
Cohesión	X	X			
Estructuración de Paquetes	X	X		X	X
Estructuración de Clases	X	X			X
Tamaño de Métodos	X	X			
Código Duplicado	X		X		X

Fuente: elaboración propia

## 2.4. Trabajos relacionados

A continuación, se describen los estudios encontrados en la literatura académica y que tienen una relación con el propósito de este trabajo de grado, utilizando las cadenas de búsqueda (i) *non-functional AND requirements*, (ii) *patterns AND non-functional*, (iii) *patterns AND non-functional AND prototype*, las cuales fueron ejecutadas inicialmente sobre las bases de datos *Scopus*, *Google Académico* y posteriormente en la *IEEE*. Los documentos fueron analizados y se presentan de forma resumida en términos de: objetivos, propuesta, resultados y diferencias.

**Tabla 3.** Trabajo relacionado # 1.

<b>Pattern based SOA deployment. (Arnold, 2007)</b>	
<b>Objetivo(s):</b>	Mejorar la forma en que son capturados los requisitos funcionales y no funcionales, mejorando las técnicas con que se realiza esta captura, pues en la actualidad dichos requisitos se capturan en documentos informales, haciendo que la implementación del servicio sea lenta, costosa y propensa a errores.
<b>Propuesta:</b>	Se presenta un enfoque novedoso de capturar formalmente las mejores prácticas de implementación de servicios como patrones basados en modelos. Tales patrones de implementación capturan la estructura de una solución, sin enlaces a instancias de recursos específicos. También, se presenta un algoritmo para la instanciación automática de dichos patrones en múltiples entornos de servicios distribuidos.
<b>Resultados:</b>	este enfoque de captura ha sido verificado en un gran prototipo que se ha utilizado para capturar una variedad de restricciones de implementación funcionales y no funcionales, demostrando su mantenimiento y realización de extremo a extremo.
<b>Diferencias:</b>	<ul style="list-style-type: none"><li>○ Es un patrón orientado exclusivamente a servicios SOA.</li><li>○ Se describe por medio de patrones basados en modelos.</li></ul>

Fuente: Arnold, 2007

**Tabla 4. Trabajo relacionado # 2.**

<b>Using software architectures and design patterns for developing distributed applications.</b> (Dabous, 2004)	
<b>Objetivo(s):</b>	Abordar la actividad de diseño desde una perspectiva empresarial en lugar de una tecnológica e identificar patrones de diseño nuevos o existentes para capturar funcionalidades de procesos de negocios comunes y cumplir con los requisitos no funcionales.
<b>Propuesta:</b>	La definición de una arquitectura de software orientada a servicios que satisfaga los requisitos funcionales en un dominio particular. Se investiga un enfoque que aborde las limitaciones de las metodologías existentes, especialmente en el caso de aplicaciones distribuidas, en una arquitectura orientada a servicios que captura funcionalidades de sistemas heredados existentes en forma de servicios básicos y procesos de negocios en forma de servicios compuestos.
<b>Resultados:</b>	Se obtuvo información útil para mejorar la arquitectura propuesta, identificando así un conjunto de patrones de diseño basados en esta arquitectura que proporcionan un marco para la reutilización de soluciones.
<b>Diferencias:</b>	<ul style="list-style-type: none"> <li>○ No se trabaja el tema de patrones de requisitos, en su lugar se plantea el mejoramiento de patrones de diseño (en la parte de implementación) para abstracciones de alto nivel que garanticen unos requisitos de calidad mínimos.</li> <li>○ No se aborda el tema de prototipos de interfaz.</li> </ul>

Fuente: Dabous, 2004

**Tabla 5. Trabajo relacionado # 3.**

<b>Applying Security Patterns For Authorization Of Users In Iot Based Applications</b> (Ali & Asif, 2018)	
<b>Objetivo(s):</b>	Brindar seguridad a dispositivos que trabajan en conjunto con aplicaciones basadas en IoT, por medio de patrones que garanticen esta seguridad a sus usuarios.

<b>Propuesta:</b>	El artículo propone ocho patrones de seguridad para brindar un alto nivel de Confidencialidad: (i) documentación del patrón de objetivo de seguridad, (ii) elegir el material correcto, (iii) patrón de inscripción de terceros, (iv) patrón de monitor de referencia, (v) reglas de autorización de matriz de acceso, (vi) control de acceso basado en roles, (vii) autenticador / autorizador remoto, (viii) autenticación de los usuarios y eventos.
<b>Resultados:</b>	Se discuten los elementos propuestos para finalmente postular a tres de ellos como los mejores en términos de confidencialidad (Documentations of security Goals, Choose the Right Stuff, Enrollment from third-party Pattern), pues se plantean en un contexto de aplicaciones médicas donde la autorización de los usuarios es una preocupación principal.
<b>Diferencias:</b>	<ul style="list-style-type: none"> <li>○ Se trabaja explícitamente para aplicaciones IoT, excluyendo los proyectos que no pertenecen a este tipo de sistema.</li> <li>○ Los patrones usados, no son de requisitos genéricos, sino específicos para el campo de la seguridad informática.</li> <li>○ No se aplica el tema de prototipos de interfaz.</li> </ul>

Fuente: Ali & Asif, 2018

Dado que se identificaron referencias importantes frente al tema de patrones de requisitos a través de los artículos encontrados inicialmente, se decidió buscar estas propuestas con el fin de complementar el tema.

**Tabla 6. Trabajo relacionado # 4.**

<b>Visualizing Non-Functional Requirements Patterns</b> (Visualización de patrones de requisitos no funcionales) (Supakkul y Chung, 2010).	
<b>Objetivo(s):</b>	<ul style="list-style-type: none"> <li>○ Capturar conocimiento sobre RNF como patrones para su reutilización.</li> <li>○ Visualizar los patrones de RNF utilizando un enfoque que permite la captura y reutilización de estos, puesto que es difícil hacerlo cuando se presentan de forma textual.</li> </ul>

<b>Propuesta:</b>	Un marco de soporte de visualización y una herramienta basada en modelos que captura, organiza, aplica y hace cumplir las restricciones de integridad basadas en reglas de refinamiento del modelo. De igual manera se proponen patrones como el patrón objetivo para capturar el conocimiento de los RNF como objetivos a alcanzar; el patrón de problemas para identificar problemas u obstáculos que se deben evitar en el cumplimiento de los objetivos; patrones de soluciones alternativas para la mitigación de problemas; y el patrón de selección para elegir entre las alternativas, teniendo en cuenta sus efectos secundarios y visualizar las relaciones entre ellos, incluida la especialización, la composición y la creación de instancias.
<b>Resultados:</b>	se necesitarán extensiones al marco actual y reglas de refinamiento adicionales, para capturar patrones de selección menos subjetivos y para capturar el conocimiento de los RNF en el contexto de los agentes y los requisitos funcionales. Dado que la herramienta implementada es solamente un prototipo, se necesitan mejoras adicionales para refinar la utilidad de las operaciones de patrón y para verificar la integridad de las reglas de refinamiento definidas.
<b>Diferencias:</b>	Esta propuesta utiliza esquemas/modelos de visualización bajo la técnica de árbol de objetivos, donde los RNF son objetivos, y de esta manera se pueden representar incluso las relaciones entre ellos. La propuesta será tomada en cuenta para enriquecer algunos aspectos del planteamiento del patrón para la especificación de RNF sobre prototipos de interfaz.

Fuente: Supakkul y Chung, 2010

**Tabla 7.** Trabajo relacionado #5.

<b>A Norm-Based Approach towards Requirements Patterns</b> (Un enfoque basado en la norma hacia los patrones de requisitos) (Ketabchi, 2011)	
<b>Objetivo(s):</b>	Incluir el concepto de patrones en la ingeniería de requisitos, y mostrar cómo estos se crean, se almacenan, se reutilizan a partir de la relación entre un problema y un requerimiento
<b>Propuesta:</b>	Se crea un patrón de problemas, el cual da una solución a un problema en específico, este patrón se almacena en un repositorio de patrones de problemas. Del mismo modo, se crea un patrón de requisitos el cual se almacena en un

	repositorio de patrones de requisitos. Luego los patrones de problemas se asignan a los patrones de requisitos, todo esto basado en la teoría de la semiótica (establece una conexión desde un signo a un objeto a través de un intérprete) de manera que los patrones de problemas en el espacio problemático se consideran signos, mientras que los patrones de requisitos en el espacio de requisitos se consideran objetos. Un intérprete (analista) encontrará la conexión entre estos dos utilizando patrones almacenados en el repositorio de patrones de requisitos. Si no se encuentra ningún patrón relacionado, el analista debe crear el patrón y almacenarlo para su uso futuro.
<b>Resultados:</b>	Después de analizar los requisitos, mediante la tabla de reglas, se procede a analizar el triángulo semiótico, para mirar las relaciones presentes entre un signo o palabra hacia un objeto a través de un intérprete, obteniendo ciertos patrones que dan como resultado un repositorio de patrones.
<b>Diferencias:</b>	Los patrones almacenados finalmente, se utilizan en fases de desarrollo, pero los RNF a los que se aplicó el patrón, no se visualizan sobre ningún artefacto técnico.

Fuente: Ketabchi, 2011

## 2.5. Discusión

La revisión de la literatura actual evidencia que los trabajos se enfocan en: (1) patrones de implementación que capturan requisitos funcionales y no funcionales, orientados netamente a prestar servicios SOA (Arnold, 2007); (2) mejoramiento de patrones de diseño en la parte de implementación, para abstracciones de alto nivel que garanticen requisitos de calidad mínimos (Dabous, 2004); (3) patrones de seguridad informática enfocado a aplicaciones IoT (Ali & Asif, 2018); (4) visualizar en modelos basados en árbol de objetivos los RNF capturados con patrones (Supakkul & Chung, 2010); y (5) encontrar la relación entre un problema y un requerimiento para su reutilización en posibles problemas futuros (Ketabchi, 2011). Sin embargo, de todos los enfoques revisados, ninguno aborda un patrón que permita especificar un RNF a través de un prototipo de interfaz como artefacto de especificación que permita la participación activa de los usuarios.

Estos aspectos evidencian que la captura de RNF utilizando prototipos de interfaz como forma de especificación no han sido abordados hasta el momento, por lo tanto, el aporte de esta investigación va dirigido hacia:

- Definir un patrón, que sirva de estrategia al ingeniero de requisitos para la captura de los RNF de manera sistemática y recurrente en un proceso de captura.
- Ampliar el conocimiento en el uso de patrones en el contexto de la especificación de RNF.
- Brindar un mecanismo que permita plasmar los RNF en un Prototipo de Interfaz.
- Aportar en la validación de RNF a través de los prototipos de interfaz.

De toda la literatura (conceptos, artículos y libros) previamente revisada, se encuentra una brecha de investigación, pues de los trabajos relacionados que se revisaron, se presentan propuestas que, aunque utilizan patrones como método de reutilización de información, tienen otros enfoques o áreas de aplicación, e incluso algunos son muy específicos enfocados en RNF como seguridad, pero aún no se cuenta con propuestas que busquen aportar a la especificación de requisitos.

Por otra parte, se evidencia que un patrón que logre aportar a la especificación de RNF y que, además, estos se puedan visualizar en un prototipo de interfaz, sigue siendo una necesidad en la industria software. La realización de este patrón para la especificación servirá para que Pymes (Pequeñas y Medianas Empresas) tengan una alternativa de herramienta fácil de implementar, para especificar los RNF, de manera que no se pasen por alto y puedan ser tenidos en cuenta en el proceso de desarrollo de software. De igual manera la implementación de esta herramienta permitiría la reutilización de información obtenida en los desarrollos de software, ahorrando tiempo y costo en nuevos desarrollos, además de que los RNF que se están especificando pueden ser plasmados por el analista en prototipos de interfaz y de este modo puedan ser percibidos por el resto del equipo, facilitando su comprensión e importancia.



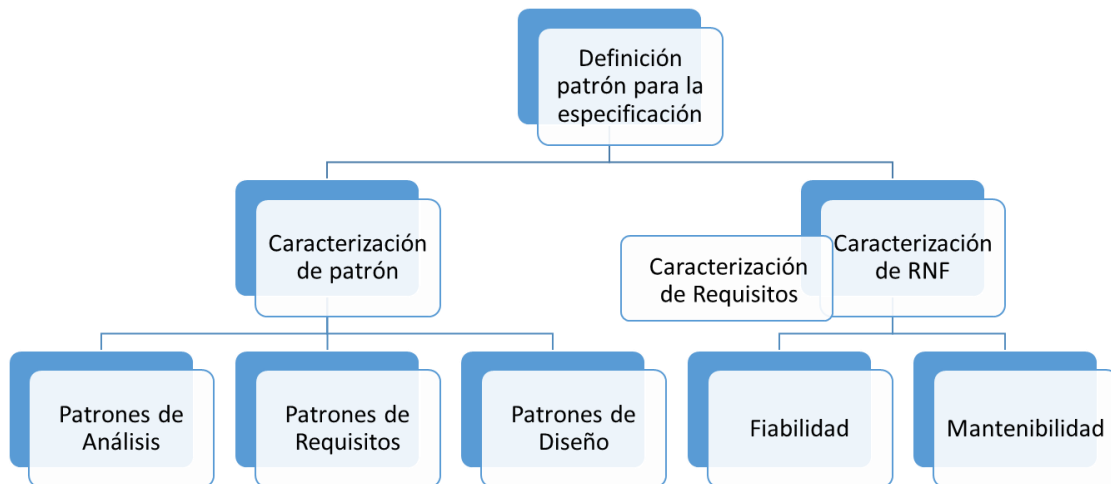
### Capítulo 3. Caracterización y definición de los componentes del patrón de especificación de RNF

Para poder definir un patrón de especificación de requisitos no funcionales es necesario primero abordar dos **dimensiones de caracterización**, como se muestra la ilustración 4. La primera dimensión presenta la Caracterización de Patrones, de la cual se desprenden los tipos de patrones que se tendrán en cuenta para esta caracterización, de ellos se extraerá un listado de las partes o elementos de un patrón propuestos por los autores analizados previamente.

Como segunda dimensión se encuentra la caracterización de RNF, esta a su vez contiene un pequeño aporte de la caracterización de requisitos en general, pues de estos se toman aspectos propios en cuanto a su estructura, que podrán servir como una base sólida para lograr una completa caracterización de los RNF.

Aunque los requisitos se dividen en Funcionales y no Funcionales, solo se realizará la caracterización de los Requisitos No Funcionales, esto debido a que este trabajo va dirigido a este tipo de requisitos. Finalmente se hace una caracterización de los dos RNF seleccionados (Fiabilidad y Mantenibilidad), en la cual se consideran los problemas que estos RNF presentan en el contexto del desarrollo de software, basados en la literatura, bien sean problemas propios de estos RNF o problemas causados al no considerarlos.

**Ilustración 4.** Caracterización de patrones y de requisitos no funcionales



Fuente: Elaboración propia

A continuación, se detalla este proceso de análisis de la literatura de acuerdo con las dimensiones mencionadas.

### 3.1. Caracterización de un patrón

Para llevar a cabo la caracterización del patrón para la especificación se utiliza la información analizada acerca de los patrones de análisis, diseño y requisitos presentada en la sección 2.2, con el propósito de consolidar los elementos que se van a tener en cuenta para la construcción del patrón de especificación de RNF que se propone en este trabajo.

Una de las primeras actividades que se llevó a cabo para esta caracterización, fue comprobar que no existieran duplicados de los elementos de un patrón en cada propuesta, por tanto, no se tendrán en cuenta los elementos generales de **contexto, problema y solución**, ya que estos están contenidos en los elementos internos de las **partes** que conforman la estructura de un patrón. Luego, se listaron todos los elementos encontrados (ver anexo 1), para posteriormente, unificar los que hacen referencia a lo mismo (tienen un mismo significado), pero que se nombran de manera diferente en cada propuesta. Por cada elemento listado se realiza un conteo del total de ocurrencias, y se ordenó de mayor a menor (ver tabla 8).

**Tabla 8.** *Número de ocurrencias de cada elemento que contiene un patrón*

Elemento	Cantidad
Nombre del Patrón o título	8
Solución	7
Consecuencias	6
Contexto	6
Problema	5
Usos conocidos	5
Intención o propósito	5
Implementación	4
Patrones relacionados/visto también	4
Ejemplo resuelto o discusión	3
Estructura	3
Aplicabilidad	3
Fuerzas, fortalezas	3
Ejemplo	3
También conocido	2
Descripción del patrón o Detalles básicos	2

Participantes	2
Colaboraciones	2
Patrones siguientes	1
Dinámica	1
Variantes	1
Contenido	1
Plantilla	1
Consideraciones para el desarrollo	1
Identificador	1
Versión	1
Autor	1
Datos específicos	1
Intervalo de tiempo	1
Importancia – Urgencia	1
Introducción	1
Consideraciones para la prueba	1
Patrones anteriores	1
Requisitos Adicionales	1
siguientes pasos	1

Fuente: Elaboración propia

### **Análisis y toma de decisiones**

Posterior a la recolección de los elementos constituyentes del patrón, se hizo un análisis, el cual concluyó en una toma de decisiones. La primera de ellas es la unificación de algunos términos en un solo concepto, ya que tienen similitud en su significado. Esto se realiza después de revisar la definición de cada concepto que propone un autor y este a su vez se vincula con otro y de un autor diferente. Aunque puede llevar distinto nombre se refieren a lo mismo. En segundo lugar, se excluyeron ciertos conceptos de la tabla, debido a que el concepto no pertenece a la actividad de especificación, siendo el caso de **diseño, ejecución**, o en conceptos como **comentarios**, donde su significado es muy ambiguo y se presta para que el analista ubique en esta sección información incorrecta. Finalmente, los conceptos que aparecen en la tabla se seleccionaron porque aportan de alguna manera a la descripción y caracterización del patrón, que son fundamentales para su construcción.

Los términos unificados fueron:

- ✓ **Título / Nombre:** Ambos conceptos se refieren a una denominación que se le da al patrón, esta denominación se utiliza de manera convencional para referirse al mismo. Debe transmitir la esencia del patrón, es decir brindar una idea de lo que él pretende resolver.
- ✓ **Ejemplo resuelto / Discusión:** Para ambos autores estos términos representan una discusión que se da sobre cualquier aspecto importante a la hora de resolver un ejemplo.
- ✓ **Intención / Propósito:** En varias propuestas aparecen estos términos, donde el objetivo es el mismo, definir qué hace en sí el patrón y qué problemas puede abordar.
- ✓ **Detalles básicos / Descripción:** Este concepto se presenta en varias propuestas, en todas refiriéndose a una descripción general que se hace del patrón.
- ✓ **Descripción del problema /Problema:** Aparece en casi todas las propuestas, aunque con distinto nombre, se refiere al problema como tal que intenta resolver el patrón.
- ✓ **Motivación / Contexto / Escenario:** En varias propuestas se maneja este concepto, en ellas se refiere a un escenario donde se desenvuelve el problema y donde se va a aplicar el patrón.
- ✓ **Fuerzas / Fortalezas:** Hace alusión a las distintas fuerzas presentes a la hora de aplicar el patrón, se analiza el contexto y como conclusión se tiene un análisis claro de pros y contras.
- ✓ **Patrones relacionados / Visto también:** para los autores estos conceptos se refieren a patrones similares o que están estrechamente relacionados, es decir que atacan problemas de la misma naturaleza.
- ✓ **Usos/ Usos conocidos:** En varias propuestas, este concepto se refiere a ejemplos de aplicación del patrón en sistemas del mundo real.

Términos que se sacaron de la lista:

- × **Diseño:** se refiere a la selección de las implementaciones de diseño, que es exclusivo para patrones de Análisis. No permite ser genérico, ni realiza aportes al patrón de especificación de este trabajo de grado.
- × **Comentarios:** sección en la que se puede ubicar cualquier aspecto que no se clasifique dentro del listado de conceptos (este concepto se saca de la lista), prestándose para

ambigüedad y además para que el analista registre aspectos que podrían clasificarse en alguno de los otros conceptos.

- × **Implementación:** se refiere a una implementación de la solución en código, en un lenguaje específico de programación, con lo cual difiere del enfoque de esta investigación, puesto que en la solución planteada no se incluye implementaciones en ningún lenguaje de programación.
- × **Ejecución:** El autor se refiere a la ejecución del código realizado para la solución propuesta en un lenguaje de programación en específico por lo tanto no se tendrá en cuenta para nuestra propuesta.

### 3.2. Caracterización de los Requisitos No Funcionales

Del mismo modo, y considerando la información presentada en la sección 2.2.1.2 (en términos de: **resumen y características de los RNF**) y que se sintetiza en el Anexo 2, se realiza una caracterización de los RNF, en la cual se incluye la identificación de unas características generales y otras específicas, con el propósito consolidar las partes o elementos que van a ser tenidos en cuenta para la construcción del patrón de especificación de RNF desde la perspectiva de los RNF.

Una de las actividades que se llevó a cabo como parte de la caracterización fue la realización de la lista de todos los elementos o partes encontradas tanto para requisitos en general como para RNF específicamente (ver anexo 2). De la lista obtenida se busca equiparar elementos que tienen diferente nombre, pero en su definición se refiere a lo mismo. Por cada elemento se realiza un conteo del total de ocurrencias en la literatura revisada, y se ordena de mayor a menor, como se muestra en la tabla siguiente (ver tabla 9).

**Tabla 9.** *Número de ocurrencia de cada elemento que contiene un requisito no funcional*

Nombre	Cantidad
Tipo	4
Identificación	2
Dependencia	2
Prioridad	1

Riesgo	1
Fuente	1
Justificación	1
Dificultad	1
Dominio	1
Alcance	1
Relaciones	1
Múltiples definiciones	1
Conceptos	1
Preguntas	1
Plantilla	1
Ejemplo	1
Obligatorio	1
Representación	1
Satisfacción	1
Rol	1

Fuente: Elaboración propia

### **Análisis y toma de decisiones**

Posterior a la recolección y análisis de los elementos constituyentes de los RNF, en relación con la tabla anterior, se determinó unificar algunos los términos “**Tipo de requisito / Tipo o asunto**”, pues al analizar las definiciones propuestas por cada autor, se halló que se refieren a lo mismo: tipo de RNF específico a la que pertenece.

Por lo demás, se mantienen los otros conceptos con sus términos, puesto que aportan a la descripción y caracterización del requisito, y por consiguiente, se considera que aportarían para la construcción del patrón de especificación de requisitos.

### **3.3. Definición del patrón de especificación de RNF**

En esta sección se presentan las definiciones finales tomadas en consideración para la construcción del patrón de especificación de RNF. Para ello, se exponen primero las definiciones de los elementos para el patrón y, posteriormente las definiciones de los elementos para RNF.

#### **3.3.1. Definición de los elementos a tener en cuenta para la construcción del patrón**

Luego de unificar y seleccionar los conceptos, que posteriormente serán incluidos en la ilustración 5, se da inicio al proceso de construcción del patrón para la especificación de RNF, para lo cual se definen los 35 elementos que se incluyeron en esta tabla.

Los elementos encontrados han sido divididos en dos grupos: elementos generales y elementos específicos de un patrón. Los elementos generales fueron seleccionados en este grupo desde la perspectiva de la estructura general de un patrón, es decir, un patrón siempre está compuesto por un contexto, problema y solución. Por su parte, los elementos específicos son los 32 restantes que permiten un detalle y una aplicación específica para la solución del problema, en este caso para solucionar el problema de especificación de los RNF. Los elementos específicos seleccionados se describen para el contexto de los requisitos, y considerando las definiciones centrales propuestas por la literatura.

##### **3.3.1.1. Elementos generales de un patrón**

**Contexto:** es un conjunto de elementos o circunstancias, que sirven para explicar una situación donde ocurre una dificultad.

**Problema:** es una situación en la que se presenta una dificultad, la cual ya se ha presentado y resuelto con anterioridad. Esta experiencia obtenida, no se tiene en cuenta para resolver dificultades de la misma naturaleza.

**Solución:** es una propuesta que permite resolver de forma idónea un problema o dificultad, que se presenta bajo un contexto específico.

##### **3.3.1.2. Elementos específicos de un patrón**

- **Nombre del patrón:** es una denominación que se le da al patrón y debe transmitir la esencia de éste, es decir brindar una idea de lo que intenta resolver.
- **También conocido como:** Son otros nombres o alias, con los que se conoce el patrón, si se conocen.
- **Autor, Fuente:** estos campos deben contener el nombre y la organización del autor y la fuente de la versión actual del requisito.
- **Patrones/técnicas previas:** es una lista de patrones o técnicas que pueden preceder al uso del patrón de especificación de RNF. Los patrones y las técnicas se relacionan en este concepto dado que permiten deducir problemas de negocio, a partir de los cuales surge una necesidad de desarrollo de software, necesidad que inicia a cubrirse a través del proceso de especificación de requisitos. Una de las técnicas es la espina de pescado, porque permite analizar un problema en términos de sus causas y efectos.
- **Introducción:** Es una explicación acerca de lo que consiste el patrón, su estructura, componentes y cuándo se debería utilizar.
- **Ejemplo:** es un caso del mundo real que demuestra la existencia del problema y la necesidad del patrón. Se usa para explicar, ilustrar o aclarar los aspectos de solución e implementación donde éste es necesario o útil. Puede ilustrarse por medio de un diagrama, los elementos que están implicados en el contexto y que de alguna manera influyen en la forma que se da la solución.
- **Consecuencias:** es una descripción de los costos y beneficios de aplicar el patrón, junto con cualquier tipo de responsabilidad que esto implique. Incluye el impacto sobre la flexibilidad, extensibilidad, portabilidad del sistema y resultados de usar el patrón.
- **Implementación:** son una serie de pautas que sirven para aplicar el patrón, a menudo describiendo detalles del problema de ejemplo.
- **Usos conocidos:** son ejemplos de uso del patrón, tomados de los sistemas comunes existentes.
- **Intención o propósito:** es una descripción del objetivo del patrón, para lo que fue concebido y describe brevemente el problema que trata de resolverse.
- **Descripción o detalles básicos:** es un listado de la información que va a resultar almacenada. Para los requisitos que implican almacenamiento, este campo debe llenarse con información relevante sobre la información que se va a almacenar.



- **Estructura:** es una descripción, en la que se especifica detalladamente los aspectos de cómo el patrón está organizado, distribuido o conformado, para cada componente participante, acompañado de un diagrama que ilustra las colaboraciones y solicitudes que se hacen entre ellos, esto para facilitar su entendimiento.
- **Aplicabilidad:** son situaciones en las que el patrón se puede emplear e incluye algunos ejemplos básicos que este puede abordar. De igual manera algunas pautas que permitan reconocer las situaciones donde el patrón se puede aplicar, en pocas palabras se hace una especie de diagnóstico.
- **Patrones relacionados/visto también:** es un listado de los patrones que están estrechamente relacionados, ya que atacan problemas similares o de la misma naturaleza. Además, estos pueden ayudar a refinar el patrón que se está describiendo. Para cada patrón relacionado se describe las diferencias o si este patrón debe usarse en colaboración con otro.
- **Fuerzas:** es una descripción, en la cual están presentes todas las soluciones posibles y las restricciones que cada una tiene, en esta se argumenta a favor de la solución requerida por el patrón y se describen las alternativas junto con una justificación de por qué estas no funcionan. Esto permite realizar un análisis del pro y contra de cada una de las soluciones, ayudando a estar seguros de que la solución es válida para la situación presente o saber con toda certeza que no lo es.
- **Ejemplo resuelto o discusión:** son una serie de recomendaciones para hacer un uso práctico del patrón, o solo presentar problemas que se deben tener en cuenta. Puede contener un ejemplo de uso, en el cual se describe la manera correcta de resolver tal ejemplo.
- **Patrones Sigüientes:** son los patrones que deben considerar en un futuro.
- **Sigüientes pasos:** es un breve resumen de lo aprendido y se plantean una serie de ejercicios para resolver con la finalidad de afianzar los conocimientos adquiridos.
- **Dinámica:** es una representación de cómo se ejecuta el patrón, en esta se describe el comportamiento en tiempo de ejecución del patrón. Puede ilustrarse más a fondo por medio de diagramas de secuencia con mensajes entre miembros participantes.

- **Variantes:** una breve descripción de especializaciones de un patrón (se describen algunas modificaciones que se pueden realizar al patrón, para solucionar problemas más específicos).
- **Participantes:** son todos los elementos que intervienen en el patrón, acompañados de sus respectivas responsabilidades.
- **Colaboraciones o dependencias:** son las interacciones entre miembros participantes del patrón, en las cuales se manifiesta como estos cooperan para llevar a cabo sus responsabilidades.
- **Contenido:** es la sustancia principal del patrón, donde se describe lo qué debe decir un requisito de este tipo y que cosas adicionales podría decir.
- **Plantilla (s):** es una guía o estructura básica, que sirve como punto de partida para escribir un requisito.
- **Consideraciones para el desarrollo:** son consejos, opiniones o advertencias, que se dan a quienes utilizan el patrón, esto con el fin de que se realice una buena implementación del patrón.
- **Identificador:** es un conjunto de caracteres, que ayuda a reconocer de manera única un requisito.
- **Versión:** siguiendo las recomendaciones de IEEE, se deben gestionar diferentes versiones de los requisitos. Este campo contiene el número de versión actual y la fecha del requisito.
- **Descripción:** este campo está conformado por los conceptos más relevantes asociados al requisito, los cuales están contenidos o almacenados de cierta forma.
- **Datos específicos:** es una lista con información específica y detallada asociada al requisito.
- **Importancia / urgencia:** es un valor, que puede ser de tipo numérico o que se le puede asignar expresiones numeradas, que indican el grado de relevancia que tiene el requisito para el producto software.
- **Intervalo de relevancia:** es un campo que indica el tiempo en que un requisito es o no relevante para el sistema.

- **Consideraciones para la prueba:** es información que se debe tener en cuenta en el momento de decidir cómo se va a probar que se consiguió el objetivo que buscaba el patrón.
- **Requisitos adicionales:** es un listado de los requisitos que podrían derivarse del tipo que se está especificando. De igual manera contiene requisitos generales del sistema que podrían ayudar a definir de alguna manera requisitos de este tipo.

### 3.3.2. Estructura del patrón de especificación de RNF

Considerando las anteriores definiciones, se propone una estructura del patrón de especificación para RNF, distribuyendo las definiciones presentadas en la sección anterior (3.3.1) en la estructura mencionada.

La estructura general del patrón de especificación se dividió en tres grupos: contexto, problema y solución, esto basado en los elementos generales del patrón descritos anteriormente. Los elementos específicos que se mencionan en tales secciones se incluyen dentro de estos tres grupos de la siguiente manera:

- **Contexto:** En este grupo se colocan elementos como patrones anteriores, ya que esta parte o elemento específico se encuentra en la etapa previa al análisis de los requisitos, es decir, en el momento donde se está concibiendo el problema, y va atado incluso a técnicas que preceden al patrón.
- **Problema:** Este grupo está conformado por elementos específicos como: introducción, ejemplo (Aplicabilidad, Usos conocidos, Plantilla, Consideraciones para el desarrollo), patrones relacionados, fuerzas, intención o propósito, datos específicos, Importancia/Urgencia (Intervalo de tiempo). Estos elementos se ubican en este grupo debido a que estos ayudan a comprender mejor el problema que se quiere solucionar, brindando una serie de elementos que facilitan su comprensión, y haciendo un preámbulo para crear la posible solución.
- **Solución:** Este grupo lo conforman elementos específicos como: estructura (Participantes, Colaboraciones), ejemplo (Aplicabilidad, Usos conocidos, Plantilla, consideraciones para el desarrollo), patrones relacionados, fuerzas, consecuencias, implementación, usos conocidos, descripción, usos futuros, dinámica, variantes (este a su

vez incluye otros elementos, los cuales darán al patrón una forma específica de aplicación), requisitos adicionales, consideraciones para prueba. Estos elementos se ubican en este grupo, puesto que algunas hacen parte del resultado obtenido a causa de la aplicación del patrón, y otras simplemente la caracterizan.

Adicionalmente, se establecieron las relaciones que existían entre algunos de los elemento o entidades que conforman cada grupo, para posteriormente clasificarlos en subgrupos. Como criterio para elegir los elementos que conforman el subgrupo y el representante de este, se analizó el concepto de cada uno y su finalidad, permitiendo establecer su relación y que conceptos estaban contenidos unos en otros, como se describe a continuación:

**Estructura:** se define como una representación gráfica de los conceptos relevantes en el patrón, utilizando una notación basada en la Técnica de Modelado. En este modelado se ilustra la secuencia de solicitudes y colaboraciones entre conceptos relevantes. A este elemento se incluyeron partes como:

- **Participantes:** hace referencia a todos los conceptos relevantes que participan en el patrón, junto con sus responsabilidades, siendo así una de las partes de la estructura del patrón.
- **Colaboraciones:** indica cómo colaboran los participantes para llevar a cabo sus responsabilidades, aportando a la completitud de la estructura.

**Ejemplo:** se define como una serie de recomendaciones para hacer un uso práctico del patrón, o solo presentar problemas que se deben tener en cuenta. Puede contener un ejemplo de uso, en el cual se describe la manera correcta de resolver tal ejemplo. Se incluyen los siguientes elementos:

- **Aplicabilidad:** son situaciones en las que el patrón se puede emplear, por lo tanto, conformaría una parte del ejemplo.
- **Usos conocidos:** son ejemplos de uso conocidos del patrón, por lo que sería una parte de este.
- **Plantilla:** se define como un punto de partida para escribir un requisito de este tipo, o más de uno si hay formas alternativas distintas, de esta manera sería un ejemplo básico

del cual se partirá, con esto agrega un grado de facilidad en la aplicación del patrón en un problema más grande o de mayor dificultad.

- **Consideraciones para el desarrollo:** se define como una serie de consejos, opiniones o advertencias que se dan a quienes utilizan el patrón, esto con el fin de que se realice una buena implementación. De esta manera constituye una pequeña parte del ejemplo, donde se muestra la manera correcta de aplicar el patrón y recomendaciones.

**Importancia/Urgencia:** indica cuán importante o urgente es un requisito, siguiendo una notación de nivel de importancia establecida previamente. Este elemento estaría conformado por:

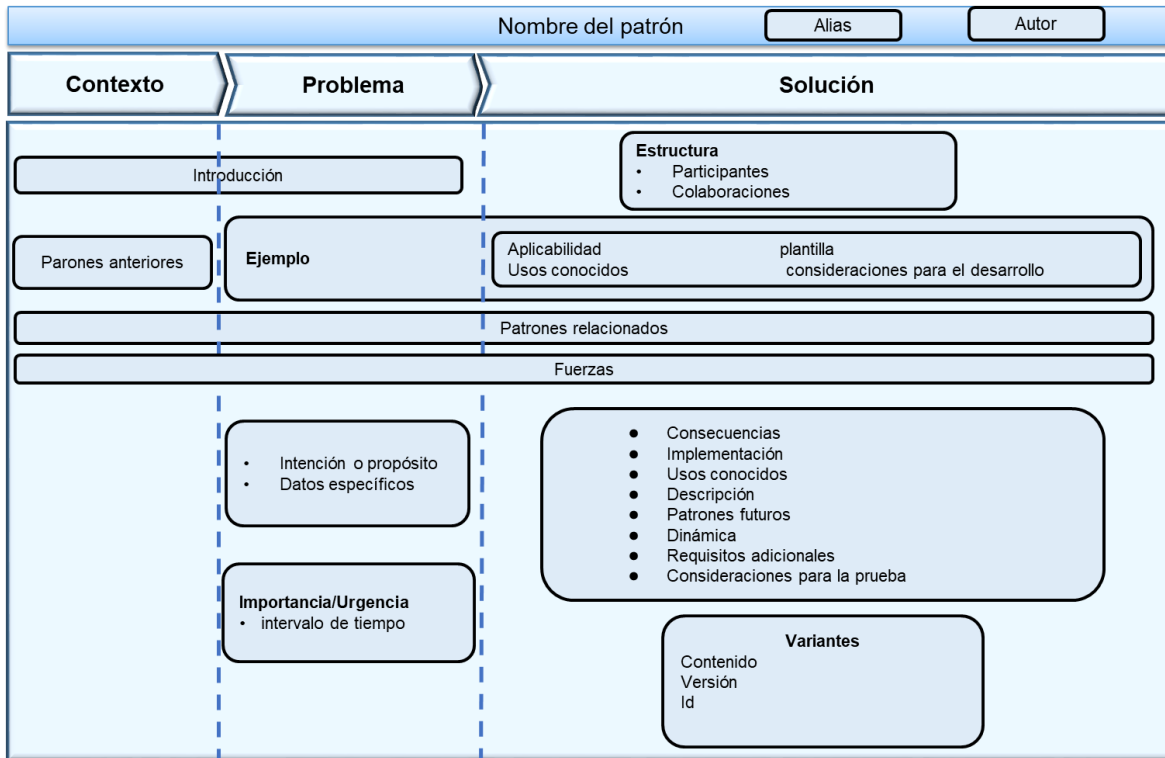
- **Intervalo de tiempo:** Indica por cuánto tiempo el concepto relevante es importante para el sistema, en este caso acompañaría a la importancia o urgencia, pues daría un tiempo para especificar si el requisito es importante o deja de serlo.

**Variantes:** Se define como una breve descripción de especializaciones, es decir, algunas modificaciones que se pueden realizar al patrón para solucionar problemas más específicos que pueden depender del requisito que se quiera especificar. Este elemento lo componen:

- **Contenido:** indica la esencia del requisito, es decir, lo que este contiene. Esto sería cambiante dependiendo del requisito.
- **Versión:** indica la versión del requisito, lo que sería específico y cambiante según el mismo.
- **Id:** ayudan a reconocer un requisito de manera única, haciendo cambiante el identificador, según el requisito que se quiera especificar.

A continuación, se muestra la estructura de la primera propuesta de patrón para la especificación de RNF (ver ilustración 5).

**Ilustración 5.** Primera versión de la propuesta del patrón para la especificación de RNF



Fuente: Elaboración propia

### 3.3.3. Definición de los elementos de Requisitos No Funcionales a tener en cuenta para la construcción del patrón

Luego del proceso de listar y unificar los conceptos frente a los RNF, se seleccionan diecinueve definiciones a ser tenidos en cuenta para el patrón:

- **Identificación:** Los identificadores únicos ayudan en el rastreo del requisito, pueden indicar vínculos o relaciones con otros requisitos. Una vez asignada esta, nunca cambia, así el requisito presente algún cambio, se reutilice en otro, o incluso si el requisito se elimina.
- **Prioridad:** Se debe establecer una prioridad a cada requisito, esto puede realizarse a través de un consenso entre los *stakeholders*, pudiendo ser por ejemplo una escala del 1-5, o un esquema simple Alto, Medio, Bajo. Esto no implica que algunos requisitos no sean necesarios, pero es útil para establecer requisitos candidatos en el momento de tomar decisiones según la necesidad de las partes interesadas.

- **Dependencia.** Esta solo se define cuando existe una dependencia entre requisitos. Algunos requisitos pueden tener una prioridad baja desde la perspectiva de una de las partes interesadas, sin embargo, son esenciales para el éxito del sistema. Por ejemplo, un requisito para medir la temperatura ambiente externa podría ser esencial para brindar soporte a otros requisitos, como el mantenimiento de la temperatura interna de la cabina. Esta relación debe identificarse de modo que, si se elimina el requisito principal, también se pueda eliminar el requisito de soporte.
- **Riesgo:** Pueden usarse técnicas de análisis de riesgo para calificar o clasificar los requisitos, en términos de su consecuencia o grado en que estos evitan o promueven riesgos. Los riesgos por considerar están relacionados con pérdidas financieras, potenciales oportunidades de pérdidas de negocio, pérdida de confianza de los *stakeholders*, impacto ambiental, problemas de seguridad y salud y normas o leyes nacionales.
- **Fuente.** Cada requisito debe incluir un atributo que indique su origen. Múltiples fuentes pueden considerarse creadores de cada uno de estos. La identificación de las fuentes para cada requisito permite identificar a qué organizaciones se debe consultar para su gestión (aclarar, solucionar conflictos, modificar o eliminar). El concepto de propiedad está relacionado con la fuente, pues dicha propiedad se aplica al origen de un requisito. La fuente del requisito indica de dónde vino, para los requisitos de las partes interesadas, la parte interesada que emite el requisito es quien tiene la propiedad. Como los requisitos se delegan aún más a través de la asignación y derivación, la responsabilidad de cumplir con estos se transfiere al equipo de producto apropiado, pero la fuente será la misma.
- **Justificación:** La justificación indica la razón por la que se necesita el requisito y puede ser respaldada por cualquier análisis de apoyo, estudio comercial, modelado, simulación u otra evidencia objetiva, que permita soportar esta justificación.
- **Dificultad:** Se debe tener en cuenta la dificultad asumida para cada requisito (por ejemplo, Fácil / Normal / Difícil). Esto proporciona un contexto adicional en términos de amplitud de requisitos y asequibilidad. También ayuda con el modelado de costos.
- **Tipo:** Los requisitos varían en intención y en los tipos de propiedades que representan. Esto ayuda a ordenar los requisitos en grupos, para su análisis y asignación.

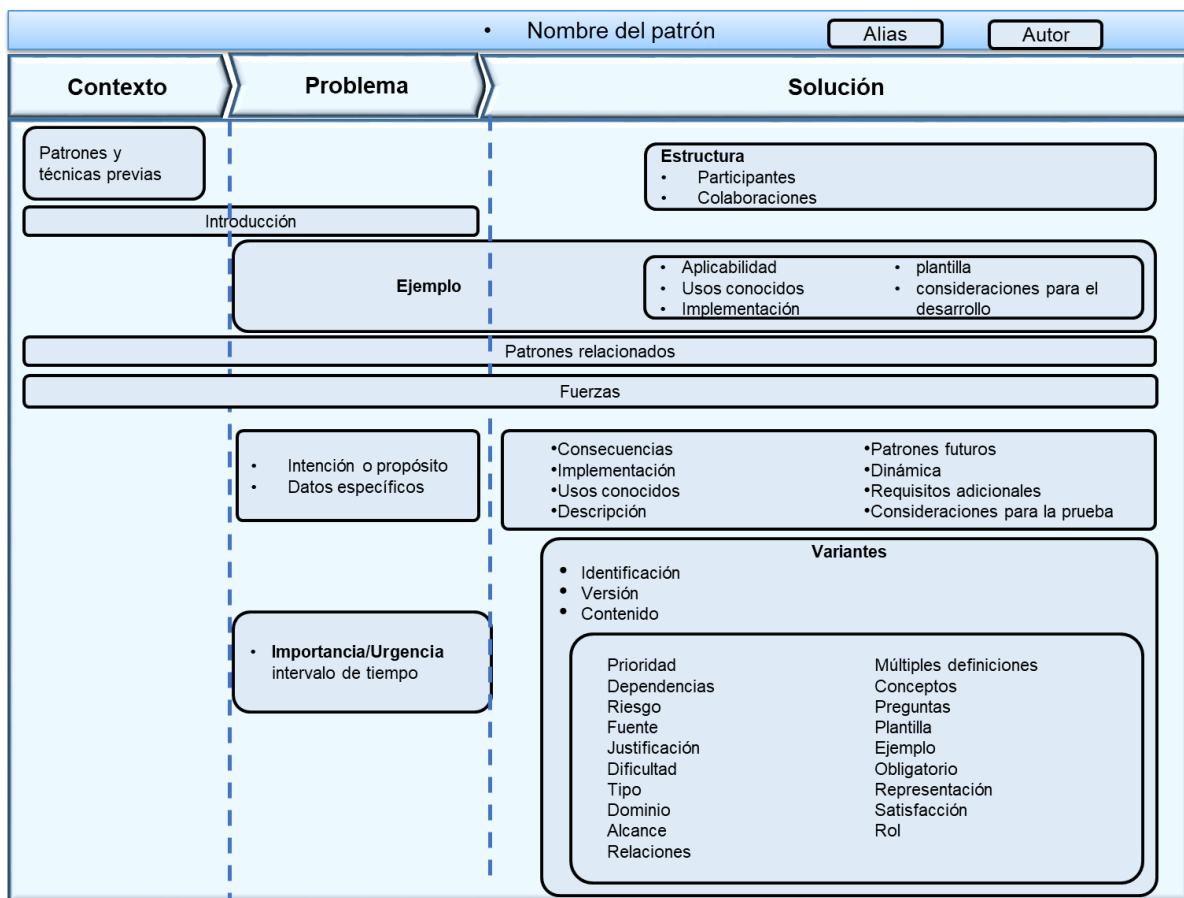
- **Dominio:** Es el conjunto de valores y operaciones válidas que se relacionan estrechamente al RNF.
- **Alcance:** Es el listado de componentes en los que se está usando el RNF.
- **Relaciones:** Son los vínculos que tiene el RNF con componentes completos o con operaciones individuales.
- **Múltiples definiciones:** Son definiciones que se le da al RNF en distintos proyectos. De estas definiciones se establece una particular que se va a utilizar en cada ámbito donde aparezca el RNF.
- **Conceptos:** Es la definición del tipo (consolidada) de requisito, descrita preferiblemente con base en normas, estándares, artículos técnicos, libros u otras referencias reconocidas.
- **Preguntas:** Pregunta hecha al cliente sobre el RNF correspondiente que permite identificar y capturar el RNF de forma correcta. Estas deben ser simples y directas para evitar la interpretación subjetiva.
- **Ejemplo:** Requisito basado en el modelo establecido que incluye valores reales para las variables. Se sugiere que los valores de ejemplo se definan en función de la capacidad de servicio de la organización, ya que, de esta forma, los analistas de negocios tienen la información que la organización puede cumplir al momento de la negociación con los clientes.
- **Obligatorio:** indica si el requisito es obligatorio o deseable.
- **Representación:** es la forma como se representa el RNF, puede ser de forma operacional, cuantitativa, cualitativa, declarativa. Los requisitos de rendimiento se representan de forma cuantitativa (medibles), cualitativas, por medio de cumplimiento de objetivos (comerciales, usabilidad.), y por último, declarativa, que se hace en restricciones, pues solo describe una situación requerida (se ejecutará en plataforma *Linux*).
- **Satisfacción:** Es el resultado de la verificación del cumplimiento de un requisito, basado en criterios que ayudan a saber si este se cumple o no. Pueden presentarse dos casos: el de los requisitos duros, donde estos se cumplen solo si se aprueban el total de los criterios establecidos, y el de los requisitos blandos, que son los que se pueden cumplir en un porcentaje, en otras palabras, se puede medir el cumplimiento en una escala de valores.
- **Rol:** Es el conjunto de funciones que puede desempeñar un requisito en una especificación, estas funciones pueden ser de 3 tipos: (1) puede especificar propiedades



del futuro sistema, (2) puede indicar hechos o reglas en el entorno del sistema que influyen en el diseño y la implementación del sistema, y (3) puede especificar cómo debe comportarse un actor en el entorno del sistema cuando interactúa con este.

Una vez definidos, analizados y revisados los conceptos de requisitos que formarían parte del patrón para la especificación de RNF, se procede a realizar una segunda versión de la propuesta del patrón (ver ilustración 6), reformando su estructura.

**Ilustración 6.** Segunda versión de la propuesta del patrón para la especificación de RNF



Fuente: Elaboración propia

Se puede observar que las partes concernientes a requisitos se ubicaron en la sección de variantes, esto debido a que en esta sección se puede detallar los diferentes RNF según su naturaleza y sus implicaciones específicas. Esta estructura es considerada la primera versión del patrón de especificación de RNF dentro de la fase metodológica del trabajo.

### 3.4. Mejora del patrón para la especificación de RNF con aspectos logrados a través de una encuesta sobre patrón MVC

En esta sección se presentan los resultados de la encuesta realizada sobre el patrón MVC (ver anexo 3), el cual fue seleccionado por ser uno de los más completos y usados, además por ser líder a nivel mundial en la industria de *software*.

La encuesta se realiza con el fin de obtener información relevante que permita complementar el diseño de la propuesta del patrón, abstrayendo sus principales características. Este ejercicio práctico fue necesario, puesto que los conceptos obtenidos en la literatura no fueron suficientes para completar la propuesta de diseño del patrón. La encuesta se dirigió a 20 diseñadores de soluciones y/o desarrolladores con alto grado de experiencia, dado que estos perfiles cuentan con la experticia y conocimiento idóneo para responder las preguntas planteadas. El instrumento se aplicó a través de la herramienta de encuestas de “Google Forms”, y se envió por medio de correo electrónico (personal ya mencionado). En el cuestionario se presentaron cuatro preguntas que fueron cuidadosamente elegidas y validadas, buscando obtener información que permitiera cumplir con el objetivo de la encuesta.

La información obtenida a través de la encuesta fue considerada para mejorar el patrón de especificación de RNF, lo que permitió obtener una segunda versión del patrón, este se presenta en la sección 3.5 del documento. A continuación, se contemplan los resultados obtenidos para cada una de las preguntas planteadas en la encuesta:

**Pregunta 1:** *¿Por qué considera que, actualmente, el patrón MVC (Modelo Vista Controlador) es uno de los patrones líder en los desarrollos de software?*

Como se observa en la tabla 10, nueve personas coincidieron con el concepto de organización y separación de capas, seguido del concepto “facilita la mantenibilidad” con siete coincidencias, lo que hace un aspecto fundamental para tener en cuenta, ya que se consigue al aplicar de buena manera el patrón MVC.

**Tabla 10.** Resultados de la encuesta del patrón MVC en la pregunta número 1

Elemento	Nro.
Organización y Separación de capas	9

Facilita mantenibilidad	7
Modular	3
Escalabilidad	3
Uso y desarrollo fácil	2
Fácil de entender	2
Integrado por defecto en <i>FrameWorks</i>	2
Soluciones interactivas que permiten retroalimentación de las acciones	1
Desarrollos rápidos	1
Seguridad	1

Fuente: Elaboración propia

**Pregunta 2:** ¿Qué características o aspectos diferencian al patrón de otros, o qué lo hace tan especial?

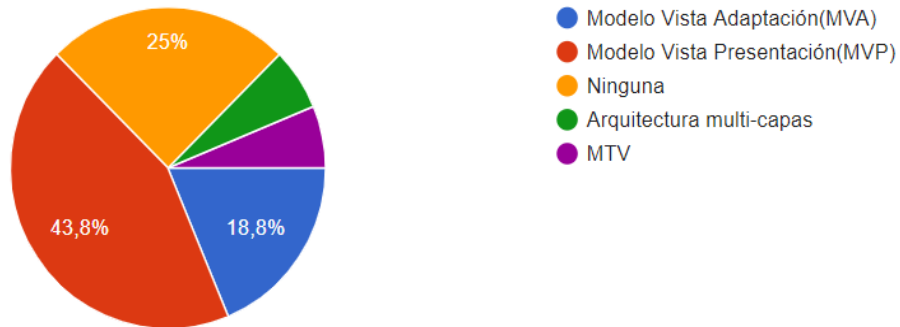
**Tabla 11.** Resultados de la encuesta del patrón MVC en la pregunta número 2

Elemento	Nro.
Microarquitectura, separada en piezas que interactúan. Permite definir la interacción de 3 aspectos fundamentales (Modelo, vista, controlador)	12
Fácil de entender	3
Fácil de usar	2
facilita la mantenibilidad	2
Fácil detección y solución de errores	1
Permite escalabilidad	1
Rapidez en el desarrollo	1
reconocido en el mundo del desarrollo	1

Fuente: Elaboración propia

**Pregunta 3:** ¿Qué variantes del MVC usa o ha usado?

**Ilustración 7.** Resultados de la encuesta del patrón MVC en la pregunta número 3



Fuente: Elaboración propia

**Pregunta 4:** ¿Utiliza el patrón MVC en colaboración con otros?, por favor describa cuál(es) (ver tabla 12).

**Tabla 12.** Resultados de la encuesta del patrón MVC en la pregunta número 4

Patrón	Nro.
Singleton	5
Ninguno	4
Factory	3
Layers (Capas)	2
Fachada	2
SOMOS (No se halló información)	1
Factory Method	1
Tiers (Niveles)	1
DAO	1
DTO	1
Proxy	1
Publicar/Subscribir	1
Front Controller	1
Page Controller	1

Patrones de bajo nivel de UI	1
Abstract Factory	1
Command	1
Mediator	1
API Gateway	1
Chain of Responsibility	1
Cliente/Servidor	1
Observer	1

Fuente: Elaboración propia

Posteriormente a la encuesta del Patrón Modelo Vista Controlador (MVC), se realizó un análisis de cada una de las respuestas que presentó cada entrevistado, destacando elementos claves de sus respuestas. De las respuestas analizadas, se realizó un conteo estadístico, con el fin de abstraer las características más relevantes que incluye este patrón, y de esta manera lograr aplicar tales características al patrón de especificación para lograr una mejora de la propuesta del que se tiene hasta el momento. Una de las características más importantes que se tienen en cuenta y que resaltan los entrevistados es la división en capas, la cual divide la arquitectura de la solución según las responsabilidades que tenga cada componente o parte de la solución.

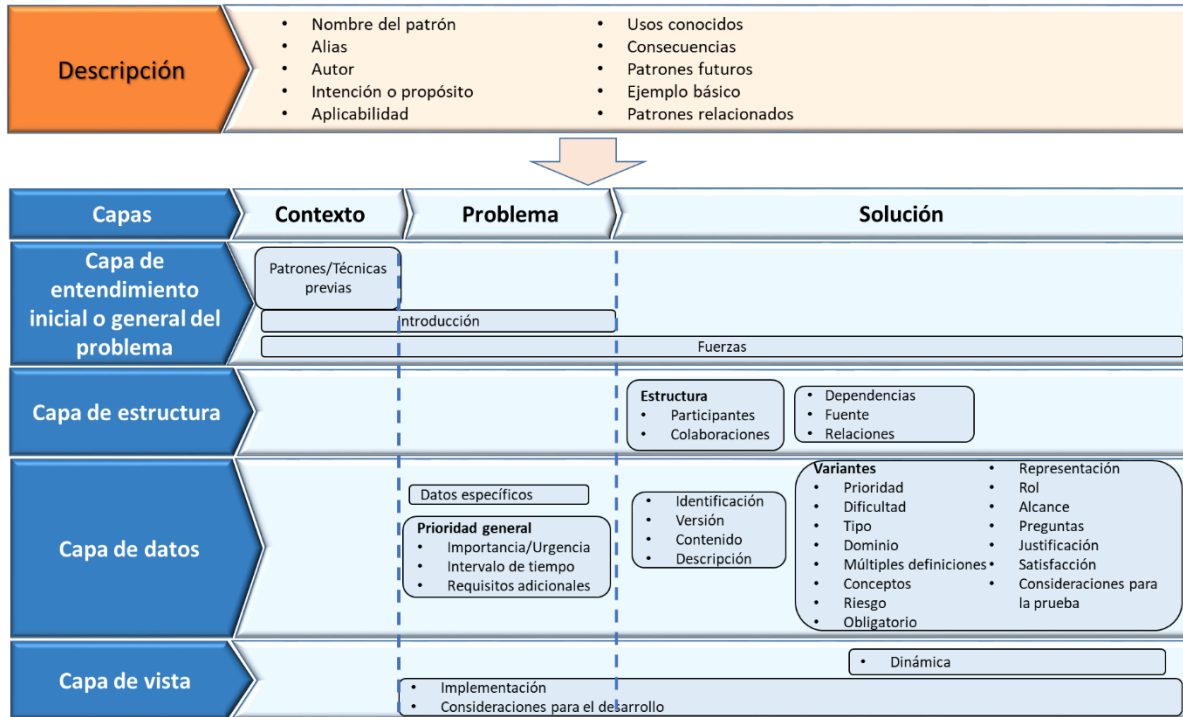
Las preguntas tres y cuatro permitieron realizar un análisis acerca de las variantes y de los patrones que colaboran con el patrón MVC. luego se concluyó que el resultado de dichas preguntas no colaboran con el objetivo del patrón, debido a que solo se usan para resolver problemas específicos de diseño, por lo anterior no se tuvieron en cuenta para la creación del presente patrón.

### 3.5. Patrón de especificación de RNF en versión final

Luego de obtener retroalimentación acerca de las características que pueden aportar al patrón, a partir de las experiencias con el patrón MVC, se genera la versión final del patrón en donde se incluyen los conceptos de capas, estos son: (i) capa de entendimiento inicial o general del problema, (ii) capa de estructura del requisito, (iii) capa de datos del requisito, (iv) capa de vista, y, adicionalmente, una sección introductoria llamada “**Descripción**” donde se incluyen

conceptos netamente informativos acerca del patrón. En la siguiente imagen (ver ilustración 8), se presenta la versión final del patrón mencionada antes, la cual será posteriormente evaluada a través de un estudio de caso.

**Ilustración 8.** Versión final de la propuesta de patrón para la especificación de RNF



Fuente: Elaboración propia

Los anteriores elementos fueron agrupados y ordenados en cada capa, de acuerdo con su definición y tomando como criterio que este aporte al cumplimiento del objetivo propuesto por dicha capa. A continuación, se lista los elementos que componen la sección y las capas anteriormente nombradas.

### Sección de descripción del patrón

En esta sección se busca que el usuario entienda algunos elementos esenciales que describen el patrón, los cuales permiten identificar en qué consiste este y en qué momentos debe aplicarse, tales conceptos estarán detallados o explicados en la guía de aplicabilidad del patrón.

Los elementos que componen esta sección son:

- Nombre del patrón
- Alias

- Autor
- Intención o propósito
- Aplicabilidad
- Usos conocidos
- Consecuencias
- Patrones futuros
- Ejemplo básico
- Patrones relacionados

### **Capa de entendimiento inicial o general del problema**

En esta capa se incluyen elementos que tengan que ver con la parte del entendimiento del problema, los cuales permitan tener una mayor claridad de este, con el fin de poder verlo de manera más entendible y así lograr plantear una solución que logre una buena especificación de los RNF. Los elementos pertenecientes a esta capa son:

- Patrones/técnicas previas
- Introducción
- Fuerzas

### **Capa de estructura**

Esta capa incluye elementos relacionados con la estructura de la solución a especificar, esto es, que permitan tener una idea de los elementos que hacen parte o se relacionan en esta solución. De igual manera, elementos que puedan brindar una idea de cuál es la interacción entre los elementos que comprenda el modelo. Los elementos incluidos en esta capa son:

- Estructura
- Participantes
- Colaboraciones
- Dependencias
- Fuente
- Relaciones

### **Capa de datos**

En esta sección se encuentran los elementos que proporcionan información detallada de cada RNF (así como un atributo de una clase). Esta información adicional se puede usar para entender mejor el RNF y cómo tratarlo; además, se considera la forma en que se almacenará esta información de tal RNF. Los siguientes elementos se consideran como datos del requisito:

- Datos específicos
- Prioridad general
- Importancia urgencia
- Intervalo de tiempo
- Requisitos adicionales
- Identificación

- Versión
- Contenido
- Descripción
- Variantes
- Prioridad específica
- Dificultad
- Tipo
- Dominio
- Múltiples definiciones
- Conceptos
- Riesgo
- Obligatorio
- Representación
- Rol
- Alcance
- Preguntas
- Justificación
- Satisfacción
- Consideraciones para la prueba

### **Capa vista**

En esta capa se representa visualmente la información de la especificación de cada RNF obtenido en las capas anteriores. Se incluyen todos los elementos que tenga que ver con interfaz, en este caso, los prototipos de interfaz se usan como ayuda visual y punto de partida para mostrar la propuesta de representación que más adelante servirá para el desarrollar el objetivo número 3 planteado en este trabajo de grado. Algunos elementos de la segunda versión de la propuesta del patrón de la estructura (ver ilustración 6) fueron reubicados en busca de un mejoramiento, de acuerdo con los resultados obtenidos por la encuesta de MVC. Entre ellos se encuentran:

- Dinámica
- Implementación
- Consideraciones para el desarrollo
- Plantilla

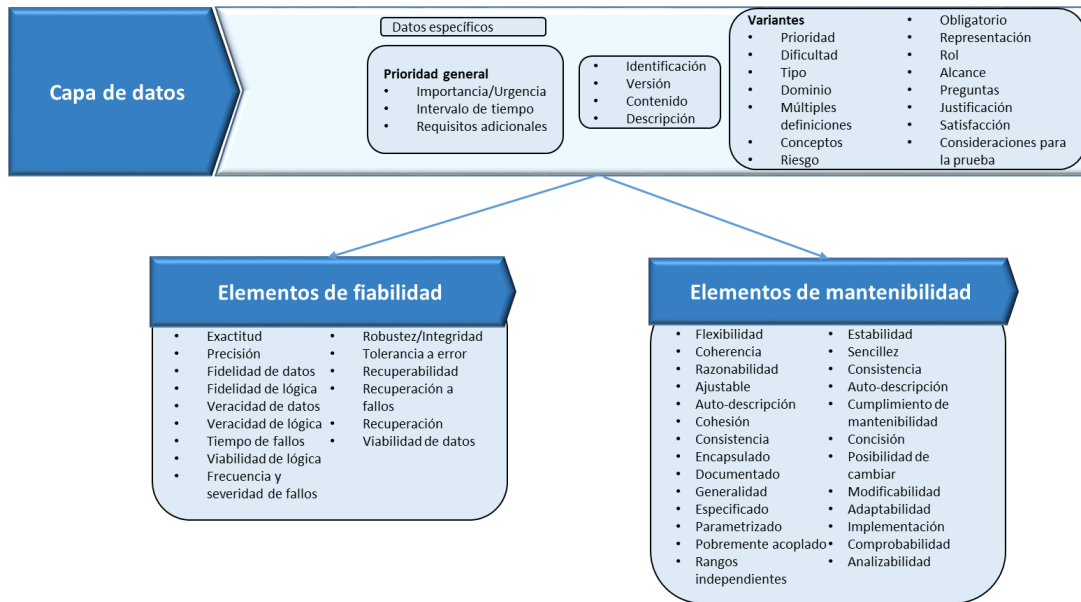
#### **3.5.1. Particularidades de las variantes y datos para considerar los elementos de fiabilidad y mantenibilidad en la versión final del patrón.**

Posterior a la obtención de la versión final del patrón, se pretende introducir los elementos o atributos de calidad de los RNF de fiabilidad y mantenibilidad definidos en la sección 2.3. del marco teórico, de tal manera que sean considerados dentro de la solución del patrón. Dado que el diseño del patrón mostrado en la última versión podría aplicarse en la



especificación de cualquier RNF, es decir, sería un patrón genérico. Se incluirán los elementos de fiabilidad y mantenibilidad manteniendo el diseño del patrón actual, por tanto, tales elementos solo se tendrán en cuenta en la instancia del patrón en el momento de redactar los elementos pertenecientes a la capa de datos. A continuación, se muestra los elementos a tener en cuenta:

**Ilustración 9.** Particularidades de variantes y datos - Elementos de fiabilidad y mantenibilidad



Fuente: Elaboración propia

Cuando el analista redacte estos RNF debe considerar todos los elementos anteriores para obtener una buena especificación de los RNF. En el siguiente apartado se explica de manera detallada cada uno de los pasos a seguir para aplicar el patrón, por medio de la guía de aplicabilidad del patrón.

### 3.6 Guía para aplicar los RNF de fiabilidad y mantenibilidad sobre el patrón para la especificación de RNF.

A partir del análisis de los conceptos y definiciones mostrados en el marco conceptual sobre requisitos funcionales y no funcionales, y más específicamente sobre los RNF de fiabilidad y mantenibilidad - que incluyen las mejoras que se realizaron a la propuesta final del patrón (Ver ilustración 8), y que fueron sustentadas a partir de la encuesta realizada sobre el patrón MVC -,

esta guía pretende brindar una serie de pasos a tener en cuenta para la aplicación del patrón como solución a un problema de especificación de RNF de un producto *software*.

Es de aclarar que la guía permite cubrir los elementos definidos y que constituyen la estructura del patrón para la especificación, el cual, puede aplicarse a cualquier RNF. Para esto se definen una serie de pasos muy generales a seguir, que serán los que permitirán al analista llegar a realizar una correcta especificación. No obstante, estos pasos, al tener un alto grado de generalidad, se debe tener en cuenta el contexto en el que se desenvuelve el desarrollo.

Por otra parte, esta guía permitirá reducir en gran manera la omisión de algunos aspectos que sean relevantes para los RNF a especificar, pues de lo contrario se generarían grandes problemas en la fase de validación con el cliente y en el *software* como tal. Cabe resaltar que los pasos definidos dentro de la guía se van a presentar o aplicar buscando la especificación de los RNF específicos de mantenibilidad y fiabilidad.

Previamente a la realización de los pasos o tareas propuestas en la guía para la aplicación del patrón, el analista debe leer, analizar y comprender los conceptos que se nombran a continuación, los cuales hacen parte de la sección “**Descripción**” de la versión final del patrón (ver ilustración 8), con el propósito de que tales conceptos sirvan como complemento desde la perspectiva técnica y de ayuda para lograr un mejor entendimiento de la finalidad del patrón. En seguida se describen cada uno de estos elementos:

**Autor:** Autores de este trabajo de grado en colaboración del director y codirector.

**Alias:** Se ha establecido nombrar a este patrón **MDV**, por sus siglas Model Data View, que hacen alusión a las tres principales capas que utiliza el patrón para cumplir su objetivo.

**Nombre del patrón:** Patrón para la especificación de requisitos no funcionales de fiabilidad y mantenibilidad.

**Intención o propósito:** La intención del patrón es brindar un análisis amplio, tanto del problema, la solución y la inclusión del proceso de negocio para lograr este análisis. De igual manera dar a conocer todos los requisitos que se encuentran inmersos dentro de un problema planteado y poder especificarlos, de manera tal que puedan ser plasmados la mayoría de estos en un prototipo de interfaz.

**Aplicabilidad:** Este patrón puede aplicarse en la etapa de análisis de un producto *software*, bien sea para sistemas legados o para sistemas que se encuentran en su fase inicial,

donde se necesite realizar un análisis exhaustivo que deje claro todo el problema y pueda llegar a obtenerse una solución óptima.

**Usos conocidos:** Especificación de requisitos no funcionales.

**Consecuencias:** El uso de este patrón representa un beneficio a largo plazo en cuestiones de costos y retrabajo, pues el análisis que se aplica tanto al problema como a la solución que se le va a dar a este, hacen que la especificación a la que se llegue tenga un alto grado de correctitud. Aunque, por otro lado, representaría un aumento en el costo, en cuanto a tiempo y recursos, designados a la fase de análisis.

**Patrones futuros:** El objetivo de este patrón es lograr una clara especificación de requisitos no funcionales. Sin embargo, una vez obtenida esta especificación se puede aplicar posteriormente otro tipo de patrones en etapas futuras a la especificación, para lograr un objetivo establecido previamente, por ejemplo, para lograr cumplir con la especificación de un RNF relacionado con el atributo de calidad “*pobrementemente acoplado*” de la característica de *mantenibilidad* se puede aplicar el patrón arquitectónico hexagonal el cual aporta al cumplimiento de lo especificado en tal RNF.

**Ejemplo básico:** Este ejemplo será presentado en la sección 3.6

Luego de leer, analizar y comprender los conceptos presentados anteriormente, se prosigue con el desarrollo de la primera capa de la guía, la cual permite entender de una forma muy general el contexto del problema, analizar sus posibles soluciones y seleccionar una de ellas de acuerdo con los criterios que cumplan con las expectativas de la organización.

### **Capa de entendimiento inicial o general del problema**

En la capa de entendimiento se van a analizar elementos importantes que ayudan al analista a entender el problema al cual se están enfrentando. Es importante que antes de todo se tenga el conocimiento previo sobre contexto del problema. Aquí se explican cada uno de los elementos que pertenecen a esta capa en la guía y que se sugiere sean considerados al momento de realizar el análisis:

- 1. Técnicas previas:** En este elemento se debe considerar y describir todas las técnicas y/o patrones previos que se usaron para lograr un análisis del problema en cuestión.

2. **Introducción:** Se debe realizar una introducción resumida acerca del problema a tratar de la organización, en la cual se explique parte del contexto.
3. **Fuerzas:** Para resolver este concepto debe tener en cuenta la siguiente tabla (ver tabla 13):

**Tabla 13.** *Fuerzas y alternativas de solución*

Nombre de la propuesta o alternativa de solución	Criterio	Pro/Ventajas	Contra/Desventajas
...	...	...	...

Fuente: Elaboración propia

Para el diligenciamiento de esta tabla realice los siguientes pasos:

- 3.1 Haga un listado de las posibles alternativas de solución para el o los requisitos.
- 3.2 Establezca unos criterios sobre los cuales se van a evaluar estas alternativas. Sobre cada uno de estos criterios se definirán los pros y contras, los cuales permitirán deliberar sobre las soluciones y poder concluir cuál de ellas es la solución más acorde con los intereses del cliente. Estos criterios pueden ser con respecto a costos, rendimiento, entre otros.
- 3.3 Pinte de color verde la fila de la alternativa de solución seleccionada, basado en los criterios del punto anterior.

A partir del análisis de las distintas soluciones planteadas anteriormente, se continúa con el proceso de especificación en la capa de estructura, en la que se trabajará a partir de la alternativa de solución seleccionada, y es esta solución la que se va a especificar a través de las diferentes capas propuestas por el patrón.

### **Capa de estructura**

En esta capa se pretende identificar cada uno de los elementos que intervienen en el proceso de negocio que se desea especificar (roles, *stakeholder*, actividades, productos de trabajo), así como también el papel que desempeñan estos elementos en un proceso. Luego de la identificación de estos elementos, se procede a representarlos a través de un proceso de modelado haciendo uso de BPMN, el cual fue escogido por ser un estándar a nivel mundial propuesto por la OMG en el modelado de procesos. A este modelado se le adicionan estereotipos que hacen parte de la propuesta de este trabajo de grado, los cuales permiten

representar elementos como: características de calidad, dependencia, etc., de manera que este modelo sirva como herramienta de apoyo al analista para la estructuración de la especificación.

Para la identificación de las características de calidad que se pueden plasmar a través de los estereotipos del modelo, el analista puede usar como herramienta de apoyo los grafos de características de calidad. Finalmente se ha definido un **instrumento** de registro que permite apoyar la identificación y la captura de los requisitos no funcionales.

Teniendo en cuenta la solución seleccionada en la capa de entendimiento, se modela el proceso que está implicado en esta. En seguida, se presentan cada uno de los pasos a seguir que permiten a su finalización la obtención del propósito de la capa:

1. Identificar los **elementos** que intervienen en los procesos de negocio que están involucrados en el problema, tales como:
  - a) Interesados (pueden ser los que requieren el producto y los que aportan en la identificación de la necesidad)
  - b) Productos de trabajo o entregables
  - c) Recursos hardware y software (Interno - Externo)
  - d) Datos
  - e) Áreas de negocio
  - f) Procesos
  
2. Ubicar estos elementos dentro de la siguiente tabla (ver tabla 14) según corresponda a la categoría *Stakeholders, Hardware, Software*.

**Tabla 14.** *Elementos o participantes que intervienen en el proceso de negocio*

Stakeholders	Hardware(Interno y Externo)	Software(Interno y Externo)	Procesos
...	...	...	...




Fuente: Elaboración propia

Para la columna de Stakeholders, realice lo siguiente:

- 2.1. Analizar y agrupar si es necesario, dependiendo de si estos realizan o no procesos o actividades en común. Estos grupos se utilizarán para la definición de los carriles de la estructura.
- 2.2. Establecer un nombre descriptivo para cada grupo (Columna - Carril), conforme a los procesos que sus miembros ejecuten.
- 2.3. Para la columna de Procesos, identificar los procesos que son ejecutados por cada grupo y en qué orden estos lo hacen.

**Nota:** Para distribuir y visualizar los elementos identificados en la tabla anterior, se sugiere usar los siguientes estereotipos del modelado BPMN (ver tabla 15):


**Tabla 15.** *Estereotipos generales de BPMN*

Símbolo	Nombre	Descripción
	Evento de inicio	Indica el primer paso de un proceso
	Flujo de secuencia	Conecta los objetos de flujo en un orden secuencial adecuado.
<b>CARRIL</b>	Responsabilidad	Los carriles se usan para organizar los aspectos de un proceso en un diagrama BPMN. Agrupan visualmente los objetos en carriles y cada aspecto del proceso se agrega a un carril separado
	Símbolo exclusivo	Evalúa el estado del proceso de negocio y, según esa condición, separa el flujo en una o más rutas que se excluyen mutuamente. Por ejemplo, se escribirá un informe si el supervisor otorga la aprobación; no se generará un informe si el supervisor no concede la aprobación.

Fuente: Elaboración propia

Adicionalmente se sugieren utilizar los siguientes estereotipos, propuestos por los autores de este trabajo de grado como extensión de BPMN, para representar elementos específicos como las características de calidad y otros elementos considerados en el diseño del patrón, que hacen parte del modelado en general:

**Tabla 16. Estereotipos propuestos como extensión de BPMN**

Símbolo	Nombre	Descripción
<b>{{FUENTE}}</b>	Fuente de producto	Represente quien o quienes solicitan el producto.
<b>-----: LETRA índice</b>	Característica de Calidad	Representa la característica de calidad asociada a un proceso, donde " <b>LETRA</b> " indica la primera letra de la característica de calidad y el " <b>índice</b> " es un número, como identificador ya que pueden existir una o más actividades relacionadas con una característica.
<b>C</b>	Colaboración	Este tipo de relación, se presenta cuando dos o más procesos interactúan para llevar a cabo un objetivo en común. Esta letra se ubicada entre la línea que flujo de secuencia entre dos procesos.
 <b>D</b>	Dependencia	Este se representa con un cuadrado de color rojo y un letra D, que encierra el proceso donde se encuentra la característica de calidad mas relevante para el producto software

Fuente: Elaboración propia

**3. Creación del modelo:**

**3.1.** Para los grupos de *Stakeholders*, crear por cada uno un carril y asignar un nombre descriptivo.

**3.2.** Cree todos los carriles.

**3.3.** Para los procesos, ordenarlos en su respectivo carril y conectarlos (**relación**), según el flujo que estos sigan y dependiendo de quién los ejecuta (**responsable**) y de la lógica de negocio.

**3.4.** A partir del modelo obtenido, identificar cuál es la **fuente (Solicitante)** del producto, e indicarlo, haciendo uso del estereotipo correspondiente, (ver tablas 15,16).

**4.** A partir del modelo, desglosar los procesar modelados en términos de **característica de calidad** que puedan estar relacionados. Para este desglose se sugiere utilizar la sección 1 del instrumento de registro (ver tabla 17).

**Tabla 17. Sección 1 del Instrumento de registro de patrón**

ID del proceso	Nombre del proceso	Actividad	ID de característica de calidad	Nombre de característica de calidad	Dependencia
P1	Nombre proceso p1	Actividad A1	c1	Nombre característica c1	
			c2	Nombre característica c2	X
		Actividad A2	c3	Nombre característica c3	
		Actividad A3	c4	Nombre característica c4	

Para el diligenciamiento de esta tabla, realice lo siguiente:

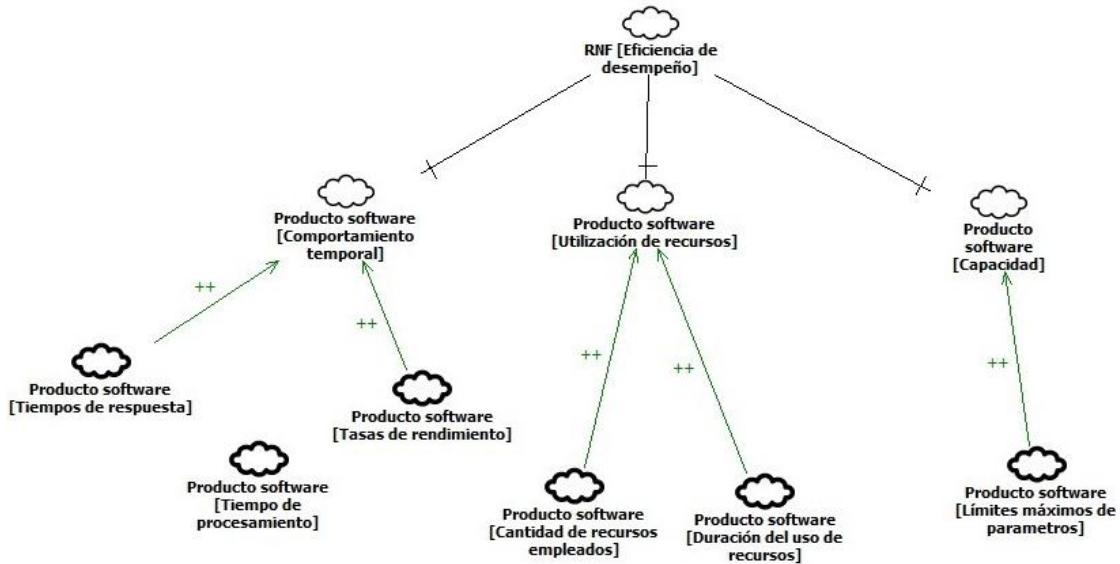
- 4.1. Ubicar en la columna **Nombre del proceso**, cada uno de los procesos identificados en la “tabla 17” y asignar un ID a cada uno de ellos, en la columna ID del proceso. Analizar y listar las actividades que considere relevantes pertenecientes a cada proceso. Estas actividades permitirán la definición del **dominio** del requisito.
  - 4.2. Para cada una de las actividades pertenecientes a cada proceso, identificar y registrar las **características de calidad (requisitos adicionales)** que podrían estar involucradas a esta, haciendo uso de la columna “Nombre características de calidad”.
  - 4.3. Establecer un identificador a cada característica de calidad listada, sobre la columna identificador, y haciendo uso del estereotipo extendido de BPMN ubíquela en el modelo.
5. Resaltar con color verde en el modelo el identificador de las características de calidad que sean fáciles de representar en el prototipo, como puede ser el caso de la usabilidad, seguridad, entre otras. Para el resto de las características, que son un poco más complejas de representar en prototipos, se deben resaltar con color rojo, pues serán las que usen la propuesta de representación brindada por este trabajo de grado para este conjunto de RNF. Sin embargo, puede haber una excepción cuando el cliente solicite la visualización obligatoria de algunas de estas características de calidad en los *mockups*, siempre y cuando esto sea posible.

**Nota:** Para identificar los **requisitos adicionales** que pueden estar asociados a una actividad, se sugiere apoyarse de la norma ISO/IEC/IEEE 25010, y de los grafos de atributos de calidad, presentados en (Arias & Rosero, 2021). A continuación, se muestran algunos de los



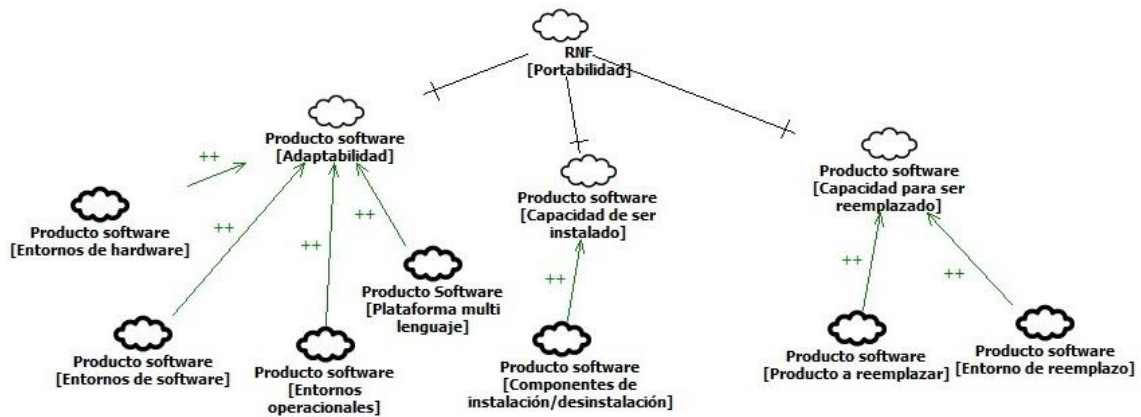
grafos de atributos de calidad. Tenga en cuenta que los grafos de fiabilidad y mantenibilidad se encuentra descritos en la sección 2.2.1.4, los cuales contienen las características del alcance del presente trabajo de grado:

**Ilustración 10.** *Grafo de eficiencia de desempeño*



Fuente: Arias & Rosero, 2021

**Ilustración 11.** *Grafo de portabilidad*



Fuente: Arias & Rosero, 2021

**Ilustración 12.** Grafo de seguridad



Fuente: Arias & Rosero, 2021

- Finalmente se debe identificar cuál es el RNF más relevante para el producto software. Una vez identificado, se debe marcar en la tabla con una X en la columna de dependencia. De igual manera esta debe marcarse en el modelado realizado anteriormente, teniendo en cuenta lo establecido para el concepto de **dependencia** que se encuentra en la tabla de estereotipos (ver tabla 16). Esto permitirá tomar decisiones en cuanto a las prioridades de implementación de los RNF que se logren especificar a través del patrón.

Para la identificación de las características de calidad que se pueden plasmar a través de los estereotipos del modelo, el analista puede usar como herramienta de apoyo los grafos de características de calidad. Finalmente se ha definido un **instrumento** de registro que permite apoyar la identificación y la captura de los requisitos no funcionales.

Teniendo en cuenta la solución seleccionada en la capa anterior (capa de entendimiento), se modela el proceso que está implicado en esta. En consecuencia, se presentan ahora cada uno de los pasos a seguir, que permiten a su finalización, la obtención del propósito de la capa:

### Capa de datos

Luego de obtener el modelado del proceso, junto con las características de calidad involucradas, se procede a establecer un detalle de estas características, identificando RNF específicos. El propósito de esta capa es brindar un detalle específico de cada RNF, información que permite tener un conocimiento completo de los datos más importantes para su futura implementación. Esta información aportará a las etapas posteriores al desarrollo de software.

Para plasmar los datos pertenecientes a esta sección, continúe relacionando los RNF de cada una de las características de los procesos, para lo cual se tendrán dispuestas las cuatro secciones faltantes del instrumento de captura de la guía.

1. Los RNF se describen en la sección 2 del instrumento de captura, como se puede visualizar en los campos que muestra la siguiente tabla (ver tabla 18):

**Tabla 18.** *Sección 2 del Instrumento de registro del patrón - RNF*

ID de la característica de calidad	Sub característica de calidad(Nivel 2 grafo)	ID RNF	Atributo de Calidad(nivel 3 grafo)	Valor del atributo nivel 3	Justificación
Característica - c1	Nombre de sub característica1	ID RNF1	Atributo1	Valor1	Justificacion1
	Nombre de sub característica2	ID RNF2	Atributo2	Valor2	Justificacion2

Fuente: Elaboración propia.

Por tanto, para el diligenciamiento de la tabla se deben realizar las siguientes acciones:

- 1.1. Tomando como base las características que se asocian a cada actividad, identificadas en el modelo, listar las sub características de calidad (nivel 2 grafo) que pertenezcan a cada una de estas, por medio del establecimiento de un **Nombre** en las columnas “sub característica de calidad”.

**Nota:** Remítase a los grafos presentados en la anterior sección y asigne como nombre la sub característica (nivel 2) que pertenece a una de las características.

- 1.2. Para la columna atributos de calidad, listar por cada sub característica, todos los atributos de calidad (nivel 3 del grafo) que estén contenidos en esta.

1.3. Asociar un RNF a cada atributo de calidad, e indicarlo por medio de la asignación de un identificador sobre la columna “ID RNF”.

**Nota:** El identificador irá descrito de la siguiente manera: los primeros dos caracteres corresponden al ID de la característica de calidad, seguido de las tres primeras letras correspondientes al nombre de la sub característica y un número incremental, dado que puede haber más atributos relacionados a la sub característica.

1.4. Para cada atributo de calidad encontrado, asignar un **valor** en la columna “valor del atributo”, solo si este lo requiere.

1.5. A partir de la identificación del RNF, escribir brevemente una **justificación** de cada uno de ellos, desde la perspectiva de negocio.

2. Determinar una prioridad para los RNF en cuestión, para ello se establece la sección 3 del instrumento llamada “prioridad del RNF” y se procede a utilizar la tabla 19:

**Tabla 19.** Sección 3 del Instrumento de registro del patrón – Prioridad del RNF

ID RNF	Importante	Urgente	Intervalo de tiempo	Valor de la prioridad	Descripción del valor de prioridad
ID RNF1	x	x	Intervalo1	Valor1	Descripción1
ID RNF2		x	Intervalo2	Valor2	Descripción2

Fuente: Elaboración propia.

Para el diligenciamiento de la tabla, se deben seguir los siguientes ítems:

2.1. En primer lugar, establezca una escala de valores bajo los cuales se definirá la prioridad. Siendo uno de estos valores correspondientes a alta prioridad y por consiguiente el otro extremo representará el valor de baja prioridad.

- 2.2. El RNF contenidos en el proceso cuya relación sea de dependencia, se le va a asignar por defecto la prioridad más alta, incluso sin llenar los campos de importante y urgente.
- 2.3. Definir si el RNF es importante o no, en términos de las consecuencias que pueda tener para el proyecto. De igual manera, definir si el requisito es urgente o no para el proyecto, en cuanto al tiempo. Para esto, marque con una X si lo es o no, para cada una según corresponda.
- 2.4. Establecer el tiempo en el cual el requisito mantiene la importancia establecida en el anterior punto. Este tiempo puede ser expresado en términos de **pasado**, si el requisito fue importante para el sistema; **presente**, si el requisito es importante solo en el presente y futuro, si el requisito no es importante en el momento, pero puede considerarse importante en el futuro. En este último intervalo “**Futuro**”, se deben considerar también las opciones de solución no seleccionadas como mejor solución en el ítem de fuerzas de la capa de entendimiento, pues tales soluciones podrían considerarse a futuro.
- 2.5. Teniendo en cuenta los valores asignados de importancia y/o urgencia, clasificar cada requisito. Para realizar esta clasificación se debe ubicar el/los requisitos basándose en el siguiente plano, indicando si es importante y/o urgente según corresponda los valores asignados.

**Ilustración 13.** *Matriz de Stephen Covey*



Fuente: Elaboración propia.

2.6. Una vez ubicado cada requisito, se procede a establecer su prioridad en la columna “descripción de valor de prioridad”, de la siguiente manera:

2.6.1. “Hazlo Inmediatamente” y tendrán prioridad 1, corresponden los requisitos que fueron ubicados como importantes y urgentes.

2.6.2. Los requisitos que sean importantes, pero no urgentes se clasifican como “Hazlo en segunda instancia” y tendrán prioridad 2.

2.6.3. Los que sean Urgentes, pero no importantes clasifican como “Delégalo” y tendrán prioridad 3.

2.6.4. Por último, los requisitos que no sean importantes ni urgentes clasifican como “No lo hagas” y tendrán prioridad 4.

3. Para la sección 4 del instrumento, llamada “Otros elementos del RNF”, se procede a utilizar la siguiente tabla (ver tabla 20):

**Tabla 20.** *Sección 4 del Instrumento de registro del patrón - Otros elementos del RNF*

ID RNF	Tipo	Dificultad	Riesgos	Obligatorio	Rol
ID RNF1	Tipo1	Dificultad1	Riesgo1, Riesgo 2,.., Riesgo n	x	Rol1
ID RNF2	Tipo2	Dificultad2	Riesgo1, Riesgo 2,.., Riesgo n		Rol2

Fuente: Elaboración propia.

Para el diligenciamiento de la tabla, se deben seguir estos pasos:

3.1. Clasifique el **Tipo** de requisito como básico o derivado, siendo derivado un requisito que depende de otro y básico un requisito independiente.

- 3.2.** Establezca la **dificultad** asumida para el requisito, por medio de la siguiente escala (Fácil / Normal / Difícil), esto dependiendo el grado de complejidad en cuanto a implementación del requisito se refiere.
- 3.3.** Identifique y liste los **riesgos** que pueden afectar al requisito, bien sean relacionados a pérdidas financieras, pérdidas de negocio, pérdidas de confianza de *stakeholders*.
- 3.4.** Indique el grado de **obligatoriedad** que representa el requisito (1. Obligatorio, 2. Deseable)
- 3.5.** Indique el rol de cada requisito, entendiéndose rol como el conjunto de funciones que puede desempeñar un requisito en una especificación, por lo cual debe escoger una de las siguientes opciones:
- 3.5.1. Propiedad Futura:** Puede especificar propiedades del futuro sistema.
- 3.5.2. Regla o hecho:** Puede indicar hechos o reglas en el entorno del sistema que influyen en el diseño y la implementación del sistema.
- 3.5.3. Comportamiento:** Puede especificar cómo debe comportarse un actor en el entorno del sistema cuando interactúa con este.
- 4.** Para la sección 5 que fue nombrada “Datos del elemento”, se van a detallar los datos de los elementos a mayor profundidad, a través de los campos de la siguiente tabla (ver tabla 21):

**Tabla 21.** *Sección 5 del Instrumento de registro del patrón - Datos del RNF*

ID RNF	Elemento	Datos a almacenar	Forma de almacenamiento
ID RNF1	Elemento1	Dato1	Forma1
		Dato2	Forma2
		Dato3	Forma3
	Elemento2	Dato4	Forma4

Fuente: Elaboración propia.

Por tanto, para el diligenciamiento de la tabla, se debe realizar los siguientes puntos:

- 4.1. Identificar la información que se va a almacenar o que va a ser tenida en cuenta, asociada a él o los requisitos en cuestión. Estos posibles elementos para almacenar se deben ubicar en la columna **Elementos**.
  
- 4.2. De los elementos listados en la columna **Elementos**, se debe indicar los **datos que se van a almacenar** y la manera o el formato en que se va a realizar este almacenamiento, en las columnas “Datos a almacenar” y “forma de almacenamiento” respectivamente, esto por cada uno de los atributos encontrados. Esta **Descripción** permite definir la **representación** del requisito, la cual puede ser de forma Cuantitativa, Cualitativa, Declarativa. Los requisitos de rendimiento se representan de forma cuantitativa (medibles). Cualitativas, por medio de cumplimiento de objetivos (comerciales) y, por último, Declarativa que se hace en restricciones, pues solo describe una situación requerida.

Después de realizar el proceso de detallar los RNF identificados, se continúa con la proyección gráfica la información recolectada en cada una de las capas anteriores, haciendo uso de interfaces gráficas.

### **Capa de vista**

En esta capa final se pretende integrar la información recopilada en cada una de las capas anteriores, con el apoyo de un artefacto visual que sintetice todos los datos obtenidos. Dicha información será plasmada sobre los *mockups* que defina la organización. Dado que algunos RNF son complejos de representar gráficamente, se presenta una propuesta que permita ilustrarlos. Se pretende que tal propuesta de representación mejore con el tiempo a partir de las experiencias adquiridas por los analistas que hagan uso de esta, aplicando el concepto de reutilización.

Para que el patrón tenga un secuencia lógica y ordenada, en esta última capa se parte del modelado obtenido en la capa de estructura. En este modelado, se le asociará un número a cada uno de los procesos que contienen RNF y que pueden ser representados gráficamente. Este



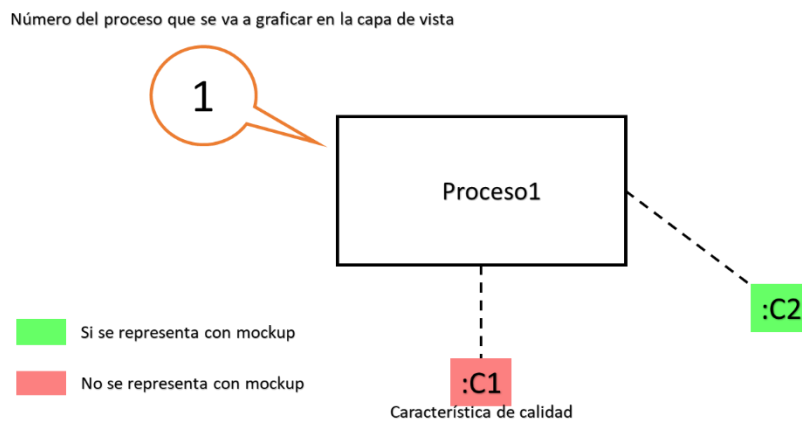
número, permitirá, posteriormente, referenciar la representación gráfica que corresponde a ese proceso.

La representación gráfica incluye una plantilla descriptiva por cada RNF que esta contenga. En dicha plantilla solo se encuentra la información de los datos asociados al RNF que el analista quiera visualizar o que considere pertinente, de acuerdo a las necesidades de la organización. En seguida, se presentan los pasos a seguir para lograr los artefactos visuales para RNF, plantilla descriptiva con mockups y plantilla descriptiva acompañado de la propuesta de representación:

**Nota:** Tenga en cuenta que, para desarrollar los siguientes pasos, debe elegir una herramienta de prototipado que este a su alcance.

1. A partir del modelado obtenido en la capa de estructura, seleccionar los procesos que contienen RNF y que fueron previamente marcados con el color correspondiente en el identificador de cada característica de calidad, y que se desean representar, asociándole a cada proceso, un número encerrado en un círculo, de la siguiente manera, (ver ilustración 14)

**Ilustración 14.** *Colores para representación de RNF para proceso del diagrama de estructura*



Fuente: Elaboración propia.

Para cada proceso seleccionado realice los siguientes pasos:

2. Realizar el *mockup* que contenga el proceso.
3. Por cada RNF que se encuentre inmerso en el proceso, usar una plantilla descriptiva como la que se muestra en la imagen, (ver ilustración 15).

**Ilustración 15.** *Plantilla descriptiva para la representación de los RNF en los prototipos*

▲ Identificador: Nombre del RNF	
Identificador: 01	
Tipo: Básico	
Representación: Operacional	
Descripción: ...	👁
Prioridad: 1 ...	👁
Dificultad: 3	
Dominio: ...	👁
Obligatorio: Si	

Elementos de la capa de datos

Fuente: Elaboración propia.

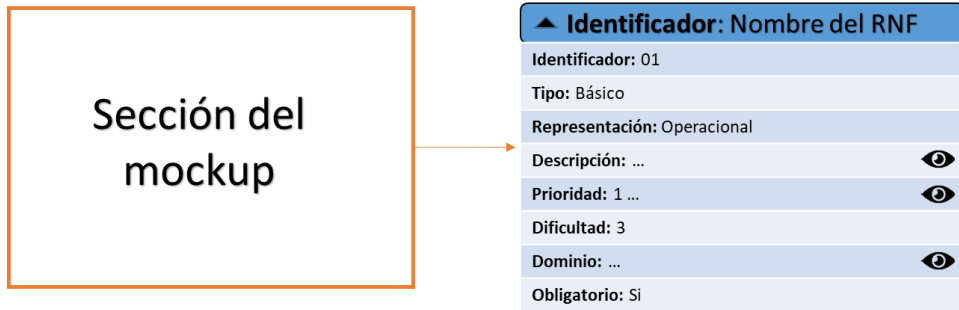
Para la elaboración de esta plantilla, se tuvieron en cuenta los siguientes elementos:

- En la parte superior, el identificador del RNF en cuestión.
- Los elementos de la capa de datos, los cuales, según su naturaleza, pueden o no visualizarse, usando como mecanismo de observación el símbolo de un “ojo”, el cual lleva a observar con más detalle este elemento.

Para diligenciar la plantilla descriptiva realice los siguientes pasos:

- 3.1.** En la parte superior de la plantilla, escriba el ID y nombre de RNF (nombre del atributo de calidad concatenado con el nombre del elemento).
  - 3.2.** Ordenar en la plantilla cada uno de los elementos de la capa de datos que usted considere necesario y que sean esenciales para dar una completa descripción del RNF.
- 4.** Encerrar con un cuadro cada sección del *mockup*, en las cuales esté inmerso el o los RNF y asociar con una flecha la plantilla descriptiva con la(s) sección(es) correspondiente(s), de la siguiente manera (ver ilustración 16).

**Ilustración 16.** Sección asociada a la plantilla de representación de los RNF en los prototipos



Fuente: Elaboración propia.

Solo para los RNF que no sea posible visibilizar y relacionar con una sección del *mockup*, haga lo siguiente:







5. Realizar una representación gráfica en la cual se pueda ver reflejado los aspectos más relevantes del RNF. Esta representación es de uso libre de acuerdo con la experiencia y experticia de los analistas, y con base en los estereotipos que usan en su organización. Sin embargo, se han propuestos algunos símbolos, que pueden ser modificados de acuerdo con la experiencia del analista y de proyecto. Aquí algunos símbolos propuestos (ver imágenes 19, 20):

**Tabla 22.** *Propuesta de estereotipos de representación para fiabilidad*

Símbolo	Nombre	Descripción
[n1,n2]	Intervalo de tiempo	Indica el valor considera como apto para realizar un proceso
	Flujo de entrada	Refleja la entrada de un proceso en el sistema
	Flujo de salida	Refleja la salida de un proceso en el sistema
	Procesamiento en el sistema	Indica un proceso ejecutado dentro del sistema
	Usuario	Usuario que realizar una acción en el sistema
	Sistema	Representa el sistema que interactúa con el usuario
	Disponibilidad	Indica que el sistema debe estar operativo, disponible y accesible para su uso cuando se requiere
	Tolerancia a fallos	Indica que el sistema debe operar según lo previstos en presencia de fallos hardware o software
	Recuperación	Indica que el sistema debe tener la capacidad de recuperación ante errores

Fuente: Elaboración propia.

**Tabla 23.** *Propuesta de estereotipos de representación para mantenibilidad*

Símbolo	Nombre	Descripción
	Flujo de entrada	Refleja la entrada de un proceso en el sistema
	Flujo de salida	Refleja la salida de un proceso en el sistema
	Procesamiento en el sistema	Indica un proceso ejecutado dentro del sistema
	Usuario	Usuario que realizar una acción en el sistema
	Sistema	Representa el sistema que interactúa con el usuario
	Codificación	Indica que el código debe estar comentado y con buenas practicas
	Documentación	Indica que la documentación es completa y clara
	Modularidad	Indica que el sistema (compuesto por componentes), al realizar un cambio en uno de sus componentes, deberá tener un impacto mínimo en los demás. <b>Nota:</b> Puede ser visualizado por medio del diagrama de arquitectura de la aplicación

Fuente: Elaboración propia.

6. Al finalizar las representaciones gráficas tanto en el *mockup* como en la representación gráfica. Disponer en una carpeta compartida en el repositorio respectivo que se haya creado para el proyecto, para obtener accesibilidad a toda la información previamente diligenciada a través patrón (**Consideraciones para la implementación**), de manera que este conocimiento tanto técnico como de artefactos visuales, pueda ser reutilizada para futuros proyectos.

Se muestra el resultado final del ejemplo implementado en la interfaz (Implementación), consideraciones para la prueba se pasó a la Capa de Datos al igual que satisfacción.

### 3.6.1 Aplicación del patrón a través de un ejemplo practico

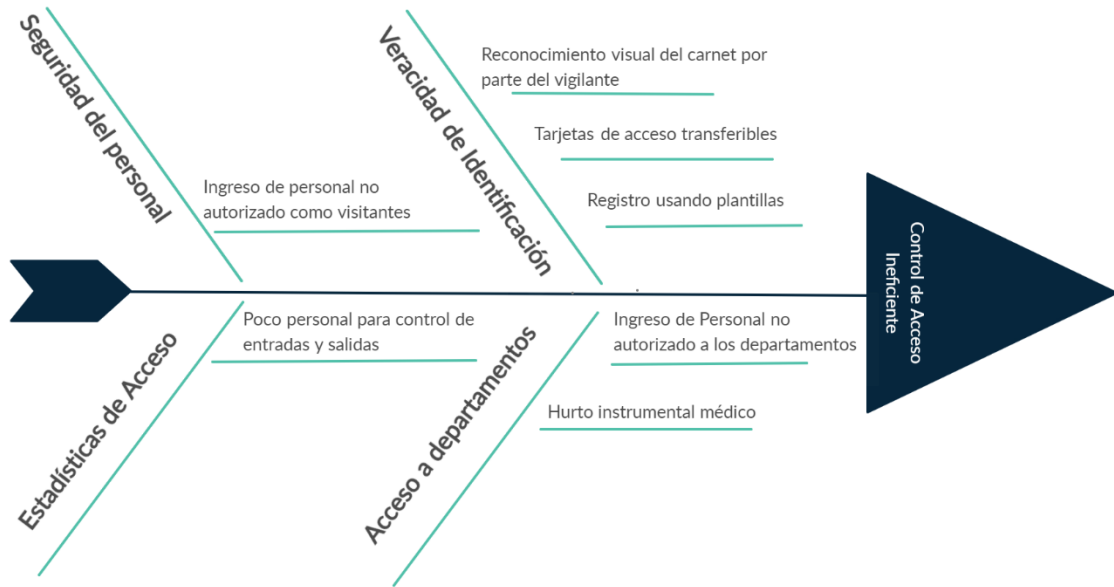
En esta sección se presenta un ejemplo ficticio, creado por los autores de este estudio, el cual fue adaptado en el área de software que se desenvuelve en el contexto de un hospital. En seguida se muestra el resultado de la aplicación del patrón sobre el ejemplo mencionado.

#### Capa de entendimiento inicial o general del problema

**Introducción:** Debido a problemas de seguridad con el ingreso del personal y visitantes a las instalaciones del hospital, el gerente requiere una mejora en el sistema de acceso, para que eleve los niveles de seguridad y permita tener un mejor control de estos accesos a las instalaciones. La identidad del personal que ingrese debe estar verificada por la base de datos de la registraduría, de igual manera las entradas y salidas de este personal, deben quedar en un registro que servirá de histórico para el departamento de auditoría.

**Patrones/ técnicas previas:** El hospital realizó un estudio del sistema anteriormente implementado (tarjetas de acceso) por medio de la técnica de espina de pescado, la cual ayudó a analizar el problema en términos de causas y efectos, como se muestra en la siguiente ilustración (ver ilustración 17):

**Ilustración 17. Técnica espina de pescado - Ejemplo**



Fuente: Elaboración propia.

**Fuerzas:** en este elemento se analizó cada una de las posibles soluciones para el problema. Al detallar cada solución en términos de pros y contras, se puede observar que la opción óptima, teniendo en cuenta los criterios mencionados y a conveniencia de la organización, es la de la implementación de reconocimiento por huella dactilar. Todos los detalles se capturan en el instrumento de captura del elemento de fuerzas (ver tabla 24).

**Tabla 24.** *Instrumento de captura del elemento – Ejemplo*

Nombre de la propuesta o alternativa de solución	Criterio	Pro/Ventajas	Contra/Desventajas
Ingreso con reconocimiento facial	Costos		<ul style="list-style-type: none"> <li>• Costos elevados tanto hardware como software</li> </ul>
	Rendimiento		<ul style="list-style-type: none"> <li>• Mayor tiempo en el almacenamiento de los registros</li> </ul>
	Seguridad	<ul style="list-style-type: none"> <li>• Intransferible</li> <li>• Alto grado de autenticidad</li> </ul>	
Ingreso con reconocimiento con escaneo de retina	Costos		<ul style="list-style-type: none"> <li>• Costos elevados tanto para el hardware como software</li> </ul>
	Rendimiento		<ul style="list-style-type: none"> <li>• Mayor tiempo en el almacenamiento de los registros</li> </ul>
	Seguridad	<ul style="list-style-type: none"> <li>• Intransferible</li> <li>• Alto grado de autenticidad</li> </ul>	
Ingreso con reconocimiento de huella dactilar	Costos	<ul style="list-style-type: none"> <li>• Costo medios</li> <li>• Desarrollo mas llevadero</li> </ul>	
	Rendimiento	<ul style="list-style-type: none"> <li>• Menor tiempo en el procesamiento de la info.</li> </ul>	
	Seguridad	<ul style="list-style-type: none"> <li>• Pocas probabilidad de suplantación</li> </ul>	<ul style="list-style-type: none"> <li>• Seguridad media-alta</li> </ul>

Fuente: Elaboración propia.

### Capa de estructura del requisito

Se procede a mostrar la lista de los elementos que intervienen en los procesos de negocio del ejemplo de seguridad, después de aplicar cada uno de los pasos y que se captura en el instrumento de captura presentado a continuación (ver tabla 25).

**Tabla 25.** *Identificación de los elementos que intervienen en el proceso de negocio – Ejemplo*

Stakeholders	Hardware(Interno y Externo)	Software(Interno y Externo)	Procesos
<ul style="list-style-type: none"> <li>•Personal hospital</li> <li>•Departamento de Auditoría</li> </ul>	<ul style="list-style-type: none"> <li>•Hardware-Lector de huella</li> <li>•Hardware externo</li> </ul>	<ul style="list-style-type: none"> <li>•Software interno del hospital</li> <li>•Software Registraduría</li> </ul>	<ul style="list-style-type: none"> <li>•Registrar usuarios</li> <li>•Verificar identidad</li> <li>•Crear registro de usuarios</li> <li>•Asignar Id</li> <li>•Leer huella</li> <li>•Ingreso al sistema</li> <li>•Confirmar o denegar acceso</li> <li>•Almacenar registro de acceso</li> <li>•Realizar auditoria</li> </ul>

Fuente: Elaboración propia.

Justificación de los requisitos asociados a una actividad y un proceso:

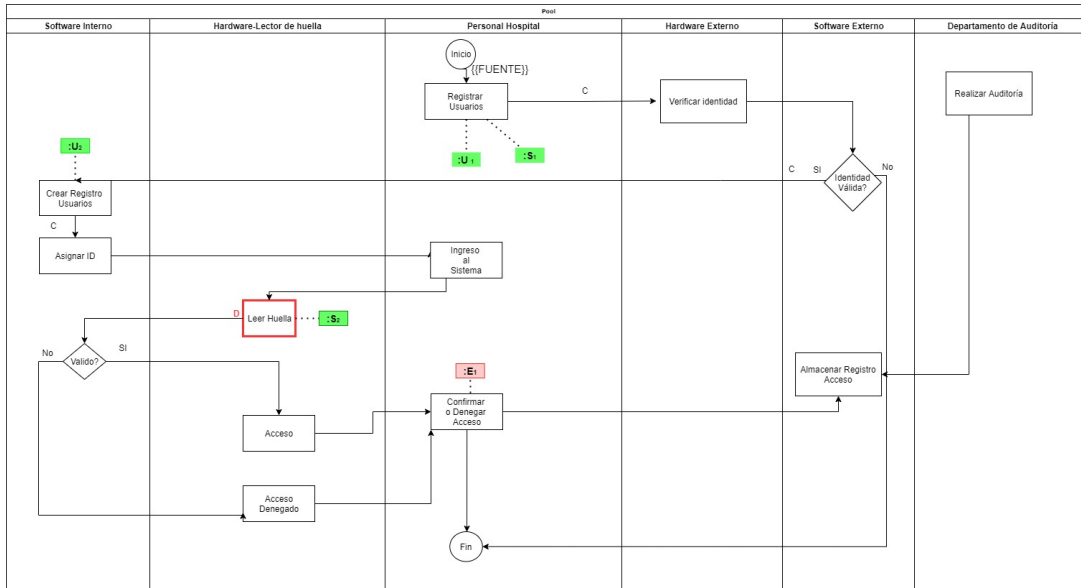
**Tabla 26.** *Sección 1 Instrumento de captura del patrón – Ejemplo*

ID del proceso	Nombre del proceso	Actividad	ID de característica de calidad	Nombre de característica de calidad	Dependencia
P1	Registrar usuarios	Almacenar datos del formulario	U1	Usabilidad	
		Guardar imagen de huella	S1	Seguridad	
		Guardar registro	S1	Seguridad	

Resultado luego de aplicar los pasos para el modelado del problema a especificar (ver ilustración 18):



### Ilustración 18. Modelado 1 – Ejemplo de seguridad



Fuente: Elaboración propia.

### Capa de Datos del requisito

En esta capa se evidencian los resultados de las cuatro secciones faltantes del instrumento de captura. A continuación, se muestra la tabla donde se evidencia dicha información, que a manera de ilustración se realiza completamente para uno de los 11 procesos que conforman la capa de estructura. (ver tablas 26, 27, 28, 29, 30):

**Tabla 27. Sección 2 Instrumento de captura del patrón – Ejemplo**

ID de la característica de calidad	Sub característica de calidad(Nivel 2 grafo)	ID RNF	Atributo de Calidad(nivel 3 grafo)	Valor del atributo nivel 3	Justificación
S2	Autenticidad	S2-AUT1	Número de verificaciones	Cantidad de huella verificadas	Este requisito es importante porque permite demostrar la identidad de usuario que usa el dispositivo

Fuente: Elaboración propia.

### Establecer una prioridad

La escala de valores seleccionada esta entre 1 y 4, siendo 1 el valor más alto y 4 el valor más bajo.

**Tabla 28.** Sección 3 Instrumento de captura del patrón – Ejemplo

ID RNF	Importante	Urgente	Intervalo de tiempo	Valor de la prioridad	Descripción del valor de prioridad
S1-AUT1	x	x	Pasado - presente	1	Hazlo inmediatamente

Fuente: Elaboración propia.

**Tabla 29.** Sección 4 Instrumento de captura del patrón – Ejemplo

ID RNF	Tipo	Dificultad	Riesgos	Obligatorio	Rol	Representación
S1-AUT1	Básico	Alta	-Ingresos no autorizados -Acceso denegado a personal autorizado	x	Propiedades del futuro sistema	cuantitativa

Fuente: Elaboración propia.

**Tabla 30.** Sección 5 Instrumento de captura del patrón – Ejemplo

ID RNF	Elemento	Datos a almacenar	Forma de almacenamiento
S1-AUT1	Accesos verificados	5 imágenes de huella	Imagen.txt

Fuente: Elaboración propia.



**Ilustración 21.** Vista general de RNF de seguridad en prototipo de interfaz – Ejemplo

**1 REGISTRO VISITANTE**

Nombres \* Apellidos \*

Tipo de identificación\* Identificación \*

CC

Correo electrónico Telefono\*

Dirección\* Código postal\*

Registra huella

Coloque dedo en el sensor

Cancelar Guardar

**▼ RNF(U1): Facilidad de registro**

**▲ RNF(S1): Seguridad del registro**

Identificador: 01

Tipo: Básico

Representación: Operacional

Descripción: ...

Prioridad: 1 ...

Dificultad: 3

Dominio: ...

Obligatorio: Si

Fuente: Elaboración propia.

**Ilustración 22.** Vista de opción ver prioridad de la plantilla descriptiva – Ejemplo

**1 REGISTRO VISITANTE**

Nombres \* Apellidos \*

Tipo de identificación\* Identificación \*

CC

Correo electrónico Telefono\*

Dirección\* Código postal\*

Registra huella

Coloque dedo en el sensor

Cancelar Guardar

**▼ RNF(U1): Facilidad de registro**

Nombre del RNF	Importante	Urgente	Intervalo de tiempo	Prioridad	Descripción de prioridad
Autenticidad de huella (Dependencia)	POR DEFECTO	POR DEFECTO	POR DEFECTO	1	Hazlo inmediatamente
Facilidad de registro	X		Presente	2	Hazlo en segunda instancia
Seguridad en el registro	X	X	Presente	1	Hazlo inmediatamente

**▲ RNF(S1): Seguridad del registro**

Identificador: 01

Tipo: Básico

Representación: Operacional

Descripción: ...

Prioridad: 1 ...

Dificultad: 3

Dominio: ...

Obligatorio: Si

Volver

Fuente: Elaboración propia.

**Ilustración 23.** *Plantilla descriptiva del prototipo*


<b>RNF(Id): Nombre del RNF</b>	
Identificador:	
Característica de calidad:	
Dependencia:	
Sub Característica de calidad:	
Atributo de calidad:	
Valor del atributo:	
Prioridad:	
Dificultad:	
Obligatoriedad:	

Fuente: Elaboración propia.

**Tabla 31.** *Sección RNF complejos de representa en los prototipos de interfaz*

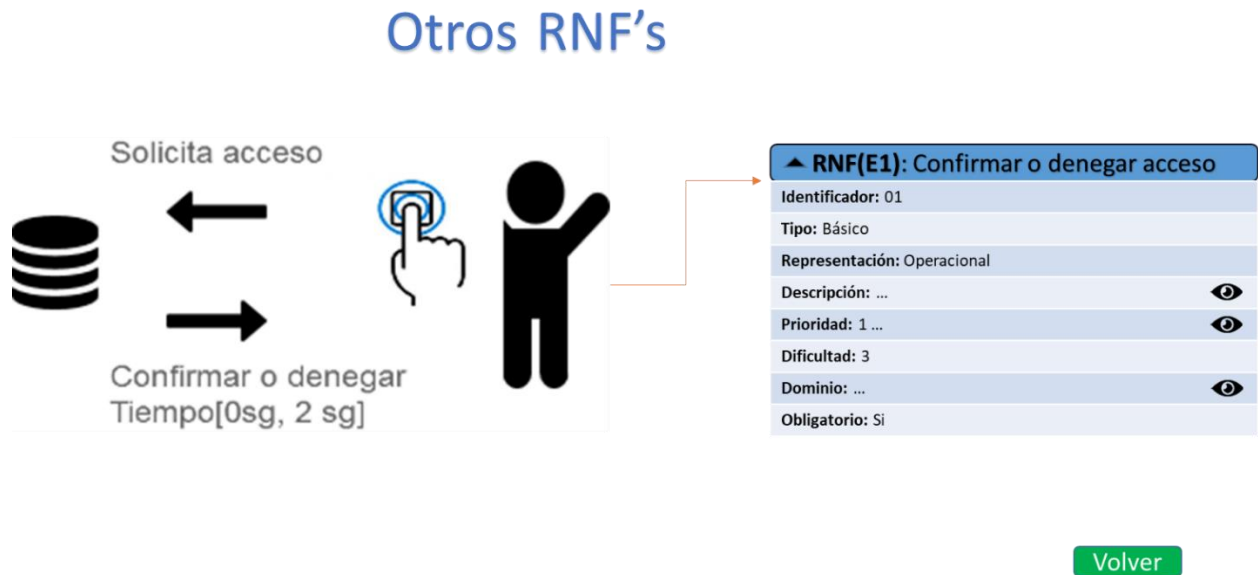
## Otros RNF

En esta sección se ubican los RNF's que son complejos de plasmar en un mockup

<b>Id</b>	<b>Nombre</b>	<b># Proceso</b>	
<b>E1</b>	<b>Confirma o denegar acceso</b>	<b>3</b>	
...	...	...	

Fuente: Elaboración propia.

**Ilustración 24.** Representación en prototipo de interfaz para RNF de Eficiencia de desempeño – Ejemplo



Fuente: Elaboración propia.

En el siguiente enlace se puede interactuar con los prototipos del ejemplo de seguridad:  
<https://daomz8.axshare.com/#c=2>

## Capítulo 4. Evaluación del patrón para la especificación de RNF

En este capítulo se presenta la aplicación y evaluación del patrón **MDV (Modelo Datos Vista)** desarrollada a través de un estudio de caso, donde se incluye el contexto de la investigación, resultados y análisis. Esta evaluación se hace a través del uso del estudio de caso y se basa en el diseño propuesto por Yin, R.K (2006), debido a que se aplicará la propuesta de patrón en un proyecto dentro de una organización, simulando así un estudio del fenómeno en su contexto real, en este caso en un proyecto de Ingeniería de Requisitos de la Universidad del Cauca, manteniendo así la integridad y las características significativas de los eventos.

Siguiendo las consideraciones definidas en (Yin, R.K 2006), el estudio de caso se describe a partir de un conjunto de etapas ordenadas de la siguiente manera: definición de las preguntas de investigación, diseño, constructos y definiciones operacionales, proposiciones teóricas, unidad de análisis, fuentes e instrumentos de recolección de datos, ejecución de estudio de caso, recolección, análisis de los resultados y conclusiones.

En la parte final del capítulo se dan a conocer las lecciones aprendidas, donde se documentan todos los hallazgos, entre estos aspectos positivos y falencias encontradas en el proceso de investigación. Lo anterior, con el fin de mejorar la anterior propuesta e incluir mejoras en futuras versiones de esta.

### 4.1. Preguntas de investigación

Las preguntas de investigación principales y secundarias que apoyan este patrón para la especificación se describen en la siguiente tabla (ver tabla 32):

**Tabla 32.** Preguntas de investigación de estudio de caso

Tipo	Preguntas
<b>Principal(P1)</b>	¿El patrón MDV, es idóneo para la especificación de los RNF de las características de calidad de fiabilidad y mantenibilidad de un producto de software?
<b>Secundarias (PS1)</b>	¿El patrón MDV permite llegar a especificar de manera completa los RNF de las características de calidad de fiabilidad y mantenibilidad en los instrumentos de captura?

<b>Secundarias (PS2)</b>	¿El patrón MDV permite llegar a especificar de manera correcta los RNF de las características de calidad de fiabilidad y mantenibilidad en los instrumentos de captura?
<b>Secundarias (PS3)</b>	¿El patrón MDV permite especificar los RNF de las características de calidad de fiabilidad y mantenibilidad provenientes del cliente de manera práctica?

Fuente: Elaboración propia.

#### 4.2. Diseño del estudio de caso

Según el enfoque propuesto por Yin, R.K (2006) el tipo de estudio de caso para esta investigación es caso único, diseño holístico (una unidad de análisis y única medición). En este estudio de caso se analizan los resultados obtenidos al abordar la unidad de análisis. A continuación, se describe el diseño en términos de: objetivo, objeto, aspectos a evaluar, contexto de la organización participante, criterio de selección de las organizaciones.

**Objetivo:** evaluar la idoneidad del patrón en términos de su completitud, correctitud y practicidad.

**Objeto:** el objeto del estudio de caso es el patrón MDV que va a ser usado en un proyecto en específico, de manera que se pueda aplicar y observar qué pasa con la especificación de los RNF.

**Aspecto evaluado:** Este estudio de caso pretende evaluar la idoneidad del patrón propuesto en cuanto a: (i) la completitud: hace referencia a si todos de los RNF contemplados en el dominio de este trabajo de grado (fiabilidad y mantenibilidad), se permiten especificar a través del patrón propuesto; (ii) correctitud: indica si la información plasmada en los instrumentos de captura del patrón corresponden a la naturaleza de la información solicitada en cada uno de sus campos; (iii) practicidad: se refiere a si el patrón, junto con sus instrumentos de captura, son eficientes y fáciles de usar por parte de los analistas a la hora de realizar la especificación de los RNF.

**Contexto y unidades de análisis:** El patrón se aplicó en la Universidad del Cauca, la cual es una organización dedicada a prestar servicios de educación universitaria, con una amplia



trayectoria. En seguida, se detallan las características de la organización donde se llevó a cabo el proceso de validación del patrón (ver tabla 33).

**Tabla 33.** *Características de la organización involucrada en el estudio de caso*

<b>Nombre de la Organización</b>	<b>Número de empleados</b>	<b>Tiempo de trayectoria empresarial</b>	<b>Actividad principal</b>
Universidad del Cauca	1256 docentes, 17188 Estudiantes	194 años	Docencia Universitaria

Fuente: Elaboración propia.

Dentro de la organización se maneja un software para la gestión académica, compuesto por funcionalidades como gestión de notas, gestión de matrículas, cancelación de materias, gestión de faltas, entre otras. Esta herramienta facilita el registro, modificación y eliminación de las notas y faltas de los estudiantes de pregrado y posgrado. Actualmente existe un inconveniente funcional a nivel de proceso de cancelación de materias, debido a que este se realiza de forma manual donde el estudiante debe realizar un gran número de trámites para llevar a cabo la tarea de cancelación. A causa de esta situación, la Universidad del Cauca busca que en un futuro este proceso se pueda realizar de manera automatizada para el estudiante y en el menor tiempo posible.

**Criterio de selección:** En criterio para seleccionar la organización fue que esta permitiera ejecutar un proceso de captura de RNF, ya sea para una nueva herramienta *software* o para adicionar mejoras a un producto existente, y que además contará con la mayor disposición para aplicar tal proceso. Posteriormente, surgió la oportunidad de apoyar a la Universidad del Cauca, en un proceso de levantamiento de RNF, en el cual se aplicó el presente estudio de caso. Se capturaron los requisitos no funcionales, y se obtuvo un documento establecido con el fin de cumplir con el proceso de desarrollo de software que espera la organización.

### 4.3. Constructos y definiciones operacionales

**Tabla 34.** *Constructos y definiciones operacionales*

<b>Constructos</b>	<b>Definición</b>	<b>Operatividad</b>
Compleitud de la especificación de los RNF de fiabilidad y mantenibilidad	La completitud hace referencia a si todos los RNF contemplados en el dominio de este trabajo de grado (fiabilidad y mantenibilidad), se permiten especificar a través del patrón propuesto.	Cantidad de RNF de fiabilidad y mantenibilidad especificados completamente por medio del patrón con respecto a la totalidad de RNF capturados
Correctitud de la especificación de los RNF de fiabilidad y mantenibilidad	La correctitud hace referencia si la información plasmada en los instrumentos de captura del patrón, corresponden a la naturaleza de la información solicitada en cada uno de sus campos.	Cantidad de RNF que se lograron especificar de manera correcta con base a los campos de los instrumentos de registro, para los cuales se valida el correcto diligenciamiento frente a lo definido en la guía de aplicabilidad del patrón
Practicidad de la especificación de los RNF de fiabilidad y mantenibilidad	La practicidad hace referencia a si el patrón, junto con sus instrumentos de captura, son eficientes y fáciles de usar por parte de los analistas, en términos de tiempo y satisfacción al momento de realizar la especificación de los RNF.	Percepción de usuario frente a la facilidad de uso de los instrumentos, tiempo y número de RNF que se lograron especificar en la aplicación del patrón

Fuente: Elaboración propia.

### 4.4. Proposiciones teóricas

El patrón MDV desarrollado permite:

- Adquirir nuevo conocimiento o experiencia por parte de los analistas en procesos de captura de RNF, pues este brinda una serie de artefactos, siendo el caso del artefacto de preguntas, los cuales aportan a la consecución de tal objetivo.
- Visualizar en los prototipos de interfaz los RNF de fiabilidad y mantenibilidad, por medio de una propuesta de visualización.

- Incorporar en el desarrollo de la herramienta de software de gestión académica aspectos claves de mantenibilidad, como lo son la integración de nuevas funcionalidades causando el menor impacto posible, entre otros.
- Promover que la herramienta de software de gestión académica en términos Fiabilidad, permita que esta misma esté disponible cuando el usuario lo requiera y además que esta pueda responder correctamente ante fallos o excepciones, entre otros.

#### 4.5. Unidad de análisis

En cuanto a los sujetos de investigación, el equipo de investigación estuvo compuesto por: dos funcionarios de la universidad, conocedores de todos los procesos que lleva a cabo en la aplicación mencionada; dos estudiantes de Ingeniería de Sistemas, con el rol de analistas, encargadas de realizar el proceso de captura de requerimientos; y dos estudiantes de Ingeniería de Sistemas, con el rol de observadores del proceso de captura y especificación.

##### 4.5.1 Muestra

Para el presente estudio se seleccionó el tipo de muestreo intencional que hace referencia a un procedimiento no probabilístico donde se selecciona a un grupo de individuos para una muestra con el objetivo de satisfacer criterios específicos. Este tipo de muestreo tiene con el fin de aportar en investigaciones o estudios de exploración, debido a que existen limitaciones en temas de recursos que impiden el uso de un muestreo aleatorio tradicional (Laureto *et al.*, 2012).

**Tabla 35.** Descripción de los estudiantes participantes

Nombre	Rol	Institución	Descripción
Angélica Pinto Rebolledo	Analista	Universidad del Cauca	Encargada de realizar el proceso de captura de requerimientos y especificación de los RNF identificados
Mayra Alejandra Castillo Motta	Analista	Universidad del Cauca	Encargada de realizar el proceso de captura de requerimientos y

			especificación de los RNF identificados
--	--	--	---

Fuente: Elaboración propia.

#### 4.6. Fuentes e instrumentos de recolección de datos

Para capturar la información que permita dar respuesta a las preguntas de investigación que se definieron para el presente estudio de caso, se establecieron un conjunto de indicadores y métricas que se muestran a continuación en la tabla 36.

##### Indicadores y métricas:

**Tabla 36.** *Indicadores y métricas - Estudio de caso*

Preguntas de investigación	Indicadores	Tipo de indicador	Mediciones	Instrumentos
¿El patrón MDV permite llegar a especificar de manera completa los RNF de las características de calidad de fiabilidad y mantenibilidad en los instrumentos de captura?	RNF identificados	Cuantitativo	Cantidad de RNF identificados en el diagrama de la capa de estructura	-Información registrada en los instrumentos de captura de la guía del patrón. -Observación de campo -Entrevista
	RNF especificados completamente	Cuantitativo	Cantidad de RNF especificados haciendo uso de los instrumentos de captura de la capa de datos.	-Información registrada en los instrumentos de captura de la guía del patrón.
	RNF representados en prototipo	Cuantitativo	Cantidad de RNF que se lograron representar completamente en los prototipos propuestos	-Información registrada en los instrumentos de captura de la guía del patrón. -Archivo de prototipado

				realizado con la herramienta de AXURE RP
¿El patrón MDV permite llegar a especificar de manera correcta los RNF de las características de fiabilidad y mantenibilidad en los instrumentos de captura?	Numero de campos diligenciados correctamente por RNF	Cuantitativo	Cantidad de campos que se diligenciaron correctamente en la capa de datos por cada RNF	-Información registrada en los instrumentos de captura de la guía del patrón.
	RNF representados correctamente	Cuantitativo	Cantidad de RNF que se representaron correctamente en los prototipos propuestos	-Información registrada en los instrumentos de captura de la guía del patrón. -Archivo de prototipado realizado con la herramienta de AXURE RP
¿El patrón MDV permite especificar los RNF de las características de fiabilidad y mantenibilidad provenientes del cliente de manera práctica?	Tiempo promedio de especificación de los RNF	Cuantitativo	Tiempo utilizado para la especificación de los RNF involucrados	-Observación de campo -Entrevista Información registrada en los instrumentos de captura de la guía del patrón. -Archivo de prototipado realizado con la herramienta de AXURE RP

Fuente: Elaboración propia.

**Instrumentos de evaluación:** Para esta investigación los instrumentos empleados, aportan datos que posteriormente son analizados y evaluados para dar validez a la información recogida. Para esta investigación los instrumentos seleccionados son:

- ✓ **Entrevista.** Este instrumento permitió al investigador obtener información cualitativa y cuantitativa de cómo los analistas lograron capturar los requerimientos del cliente, haciendo uso del documento de lista de chequeo que propone el patrón.

- ✓ **Observación de campo.** Acción que permitió a los investigadores obtener información cualitativa y observar el comportamiento de los analistas durante el proceso de captura, haciendo uso del patrón propuesto.
- ✓ **Encuesta.** Herramienta que permitió al investigador obtener información cuantitativa y cualitativa, acerca del nivel de practicidad del patrón, usando los instrumentos de registro en términos de tiempo.
- ✓ **Información registrada en los instrumentos de captura del patrón:** corresponde a la información escrita en los instrumentos propuestos, como lo son la lista de chequeo y la guía de aplicabilidad del patrón, después de ser aplicado.

**Tabla 37.** Recursos de apoyo

Recursos	Propósito
7 computadores	Herramienta principal de comunicación para las reuniones que se llevaron a cabo, y además se usó para completar los documentos digitales.
Documentos en el software de ofimática Office 2019 (Word, Excel y PowerPoint)	Registrar la información obtenida en el estudio de caso.
Draw.io	Permite realizar el modelado de la capa de estructura.
Axure RP para prototipos	Permite realizar la implementación de los prototipos de manera profesional.
Software de Videoconferencia Google Meet	Llevar a cabo la comunicación virtual entre los participantes de estudio.

Fuente: Elaboración propia.

#### 4.7. Ejecución del estudio de caso

Teniendo en cuenta la metodología de investigación propuesta por Yin, R.K (2006) para realizar la ejecución del actual estudio de caso, se tomaron en cuenta las siguientes etapas: preparación, ejecución y seguimiento. Estas etapas se describen a mayor detalle a continuación:

##### **Preparación:**

1. Presentación e introducción a la propuesta, en diapositivas con la herramienta ofimática de PowerPoint.
2. Capacitación teórica del patrón MDV.
3. Capacitación en el uso de los instrumentos de captura y registro del patrón MDV.
4. Reunión de captura con el cliente para aplicar el patrón MDV usando el instrumento de lista de chequeo.
5. Reuniones de especificación y refinamiento de los RNF usando la guía de aplicabilidad del patrón y adicionalmente se realizó una sección para la aclaración de dudas.
6. Diligenciamiento de la encuesta por parte de los analistas donde se indaga acerca del nivel de conocimiento, aspectos de correctitud, completitud y practicidad del patrón, por último, una serie de preguntas de percepción sobre la aplicación de este.
7. Recepción de los documentos y archivos diligenciados por los analistas.
8. Retroalimentación por parte de los analistas hacia los investigadores.

### **Ejecución:**

El apoyo de los funcionarios de la Universidad del Cauca con el rol de clientes concedores del proyecto a través de las reuniones, permitieron a los responsables de la captura de RNF involucrados en el proyecto software alcanzar un conocimiento concreto sobre estos mismos. De acuerdo con las evidencias tempranas del proceso de validación, la organización realizaba de manera informal esta especificación, generando de esta manera la omisión en cuanto a captura de RNF de fiabilidad y mantenibilidad, que es precisamente el principal problema sustentado en este proyecto.

Dentro de las actividades realizadas en el estudio de caso, también se incluyen las reuniones de capacitación previas al proceso de captura de requerimientos, con el fin de obtener retroalimentación con respecto a cómo se debe aplicar el patrón MDV, apoyándose de algunos artefactos como lo son la guía de aplicación del patrón y el documento de lista de chequeo. A continuación, se muestran todas las actividades que se llevaron a cabo en el estudio de caso. (ver tabla 39)

**Tabla 39.** Plan desarrollado en el proceso de intervención

Tipo de finalidad de la socialización	Fechas de ejecución	Objetivo
<b>Reuniones previas al proceso de captura</b>		
Reunión virtual (plataforma <i>Meet</i> ) que se llevó a cabo para la capacitación de la aplicación del patrón para la especificación de RNF por medio de un ejemplo de seguridad	20-03-2021	<ul style="list-style-type: none"> <li>-Dar a conocer una vista general acerca del patrón MDV</li> <li>-Capacitar al equipo de ingeniería de requisitos, para que logren una buena aplicabilidad del patrón.</li> </ul>
Reunión virtual (plataforma <i>Meet</i> ) en donde se realiza la capacitación del documento de apoyo “lista de chequeo”.	24-03-2021	<ul style="list-style-type: none"> <li>-Entregar y explicar el documento de “<b>lista de chequeo</b>” que servirá como apoyo en la entrevista con el cliente, este contiene preguntas claves para la recolección de la información necesaria para aplicar el patrón.</li> <li>-Responder las dudas relacionadas con el documento de “lista de chequeo”.</li> </ul>
<b>Reunión de captura</b>		
Reunión de captura de requerimientos con el cliente	25-03-2021	<ul style="list-style-type: none"> <li>-Capturar los requerimientos del cliente en el documento de lista de chequeo, para obtener toda la información necesaria que posteriormente permita la aplicabilidad completa del patrón.</li> </ul>
<b>Reuniones posteriores a la captura para la especificación de RNF</b>		
Primera reunión para la especificación de RNF apoyándose de la guía de aplicabilidad del patrón	26-03-2021	<ul style="list-style-type: none"> <li>-Se realizó una reunión conjunta entre analistas y observadores para la especificación de los RNF, en donde el grupo de analistas desarrolló los pasos planteados en la guía para aplicar el patrón.</li> </ul>



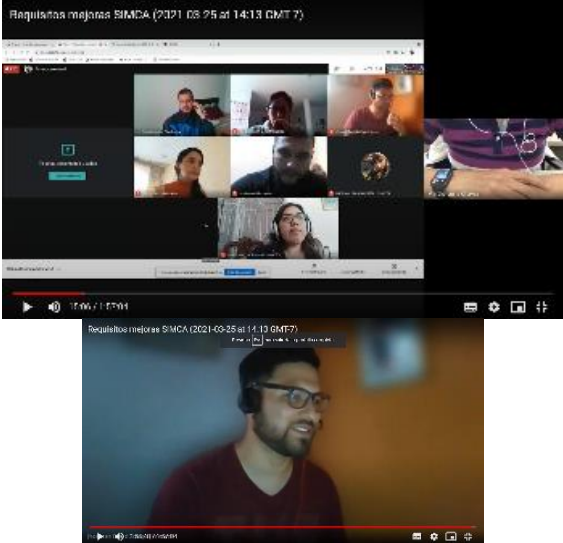
Segunda reunión para la especificación de RNF y diligenciamiento de la encuesta	27-03-2021	-Se realizo una reunión conjunta entre analistas y observadores para la especificación de los RNF, en donde el grupo de analistas desarrolló los pasos finales en la capa de vista que fueron planteados en la guía para aplicar el patrón. Por ultimo los analistas diligencian la encuesta y dan retroalimentación a los investigadores sobre el proceso de aplicación del patrón.
---	------------	--

Fuente: Elaboración propia.

Se realiza una reunión vía meet con el cliente para la ejecución del proceso de captura de requerimientos, con el fin de obtener la información necesaria que sirva para realizar la propuesta de aplicación del patrón. De igual manera, el equipo de analistas dirigió la sesión guiándose en el documento de “lista de chequeo” que se encuentra en el Anexo 8, el cual a su vez se apoya en el complemento de preguntas del anexo 7, para poder extraer dicha información.

Transversalmente durante el desarrollo de la sesión, los observadores plasmaron la información en los instrumentos preparados previamente para la aplicación de la propuesta del patrón, dando así un primer acercamiento a la solución de tal propuesta. Al finalizar la sesión con el cliente, se realizó una retroalimentación entre el equipo de analistas y observadores, discutiendo aspectos que validen si la información capturada es suficiente para llevar a cabo la aplicación del patrón, o si se debe solicitar una nueva sesión que permita culminar el proceso y obtener la información faltante, a lo cual se concluyó que la información capturada sí es suficiente para poder llevar a cabo dicho desarrollo.



<p>Jhonatan David Zúñiga Jiménez, Miller Santiago Castillo Muñoz</p>	<p>Observadores</p>	
--	---------------------	--

Fuente: Elaboración propia.

### Seguimiento:

A partir de la información capturada y registrada en la lista de chequeo, la cual sería un primer acercamiento a la solución del estudio de caso con la propuesta del patrón, se procedió a refinar tal solución haciendo uso de la guía de aplicabilidad del patrón a través de los instrumentos propuestos por ésta, con el fin de obtener de manera completa la especificación de los RNF.

Luego de que los analistas aplicaran la guía del patrón, se realizó una reunión conjunta entre analistas y observadores, donde se socializaron los resultados obtenidos y finalmente se refinó el informe de aplicación de la propuesta en la organización. Adicionalmente de reciben los documentos y archivos diligenciados en el presente estudio también la retroalimentación de los analistas hacia los investigadores.

### 4.8. Recolección:

En el proceso de aplicación del patrón se especificaron veintitrés RNF, de los cuales solo se mostrarán doce a manera de ejemplo, debido a que son de interés para la presente investigación y además por lo extenso del instrumento de captura, por tanto, se adjunta un documento de anexos (ver anexo 6) con la totalidad de la información de los demás RNF.

A continuación, se detallan los resultados obtenidos de cada capa, como lo describe la propuesta del patrón.

### **Capa de entendimiento**

En esta capa se describe el resultado obtenido por cada ítem que conforman esta misma, el cual fue discutido y concluido en conjunto con los clientes y analistas del presente estudio. Para el ítem de técnicas previas se observó que el cliente realizó un proceso de análisis del problema, en el cual mezcló un conjunto de técnicas con base en su experiencia y lineamientos de su organización para desarrollarlo. Posteriormente se realiza una introducción sobre el proyecto y finalmente se discute las posibles soluciones al problema planteado, donde se elige una de ellas. A continuación, se muestra el resumen de los resultados de la capa.

**Técnicas previas:** Descripción del proceso de análisis por parte del cliente.

- A partir de la experiencia y uso de la aplicación de SIMCA se identifica un problema.
- Se presenta como propuesta de cambio y actualización del sistema.
- Se redacta en un documento de requerimientos por parte del cliente, en el cual contiene capturas de pantalla y descripción de los problemas encontrados.
- Se realiza la entrega al equipo de ingeniería de requisitos, por medio de una reunión virtual para su posterior análisis.

**Introducción:** Dentro de las instalaciones de la Universidad del Cauca, en la división de SIMCA, se hace uso de una aplicación que gestiona todo lo relacionado a matrículas, cancelaciones, etc., de los estudiantes, profesores y demás. En una de las funcionalidades de “**cancelaciones de materias**” se evidencia un inconveniente con el proceso de asignación de la condición en la que se encuentra la asignatura y el motivo por el cual se cancela la misma, para lo cual se desea obtener una mejora en esta funcionalidad, y adicionalmente el proceso de cancelación por parte del estudiante se hace de manera manual con el llenado de un documento físico.

**Fuerzas:** Aquí se evidencian las distintas soluciones posibles que serán evaluadas de acuerdo con los criterios y necesidad de la organización:

**Tabla 38. Elementos de fuerza – capa de entendimiento**

Nombre de la propuesta o alternativa de solución	Criterio	Pro/Ventajas	Contra/Desventajas
Proceso de cancelación de materias, a través de el aplicativo web	Rendimiento	<ul style="list-style-type: none"> <li>Disminución en tiempo de trabajo para el proceso de cancelación</li> <li>Mayor optimización del proceso</li> </ul>	
	Seguridad		<ul style="list-style-type: none"> <li>Restricciones de seguridad referente a la autenticación para prestar el servicio web</li> </ul>
Proceso de cancelación a través de un cargue masivo en formato Excel	Rendimiento		<ul style="list-style-type: none"> <li>Posibles problemas de seguridad, a través del cargue de los datos directamente a la base de Datos</li> </ul>
	Seguridad	<ul style="list-style-type: none"> <li>Disminución en tiempo en el proceso de cancelaciones</li> </ul>	

Fuente: Elaboración propia.

Dado que es indispensable mejorar los procesos dentro de la Universidad de Cauca, se ha decidido optar por la primera opción (la fila seleccionada se debe marcar de color verde), ya que se pretenden disminuir considerablemente los tiempos del proceso de cancelación de materias para los estudiantes y hacerlo de manera óptima por medio de la aplicación web, adicionando una nueva funcionalidad en la aplicación que se encuentra activa actualmente.

Seguidamente se encuentra el proceso de la capa de estructura teniendo en cuenta la solución seleccionada anteriormente.

### Capa de estructura

En este apartado se describen los resultados obtenidos de cada paso para desarrollar la capa de estructura, estos muestran el camino recorrido para finalmente obtener el modelado del proceso junto con la identificación de las características de calidad que espera la organización.

1. Identificación de los elementos:
  - a. Interesados: Estudiante, Facultad, Decano, DARCA, Administrador SIMCA
  - b. Productos de trabajo o entregables: Resolución de cancelación
  - c. Recursos *hardware* y *software*: Dispositivo que diligencia y software de SIMCA
2. Se procede a ubicar los elementos identificados en la siguiente tabla (ver tabla 40):

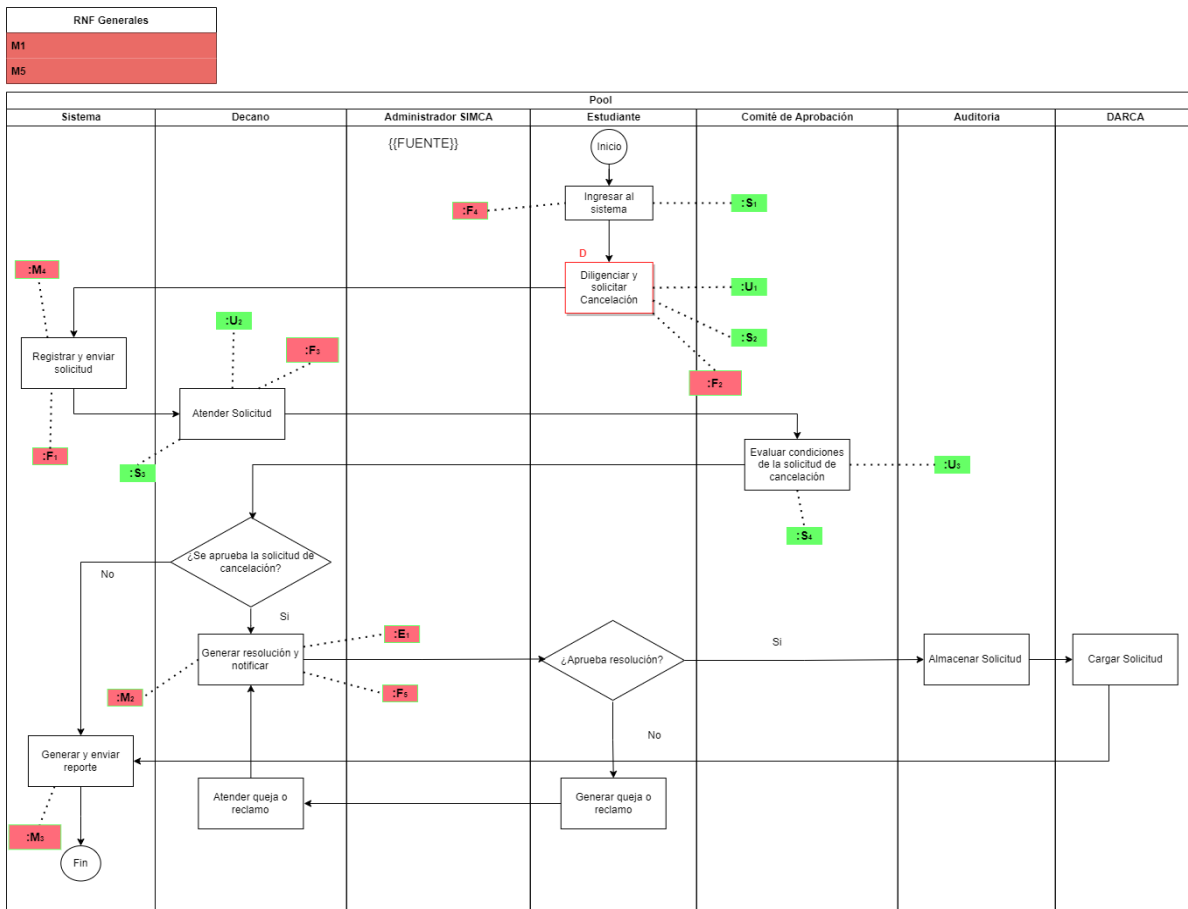
**Tabla 39.** Elementos identificados en el proceso de negocio – Estudio de caso

Stakeholders	Hardware(Interno y Externo)	Software(Interno y Externo)	Procesos
<ul style="list-style-type: none"> <li>Sistema</li> <li>Decano</li> <li>Administrador SIMCA</li> <li>Estudiante</li> <li>Comité de aprobación</li> <li>Auditoria</li> <li>DARCA</li> </ul>	<ul style="list-style-type: none"> <li>No aplica</li> </ul>	<ul style="list-style-type: none"> <li>Sistema SIMCA</li> </ul>	<ul style="list-style-type: none"> <li>Ingresar al sistema</li> <li>Diligenciar y solicitar cancelación</li> <li>Registrar solicitud y enviar solicitud</li> <li>Atender solicitud</li> <li>Enviar condiciones de la solicitud de cancelación</li> <li>Generar resolución y notificar</li> <li>Almacenar solicitud</li> <li>Cargar solicitud</li> <li>Generar y enviar reporte</li> <li>Generar queja o reclamo</li> <li>Atender queja o reclamo</li> </ul>

Fuente: Elaboración propia.

3. Creación del modelo: En la imagen que se presenta a continuación (ver ilustración 25) se encuentra el modelo obtenido con la aplicación de los ítems establecidos.

**Ilustración 25.** Modelado inicial de los procesos – capa de estructura del estudio de caso



Fuente: Elaboración propia.

4. A partir del modelo, se desglosan todos procesos identificados en términos de **característica de calidad** en la siguiente tabla (ver tabla 34):

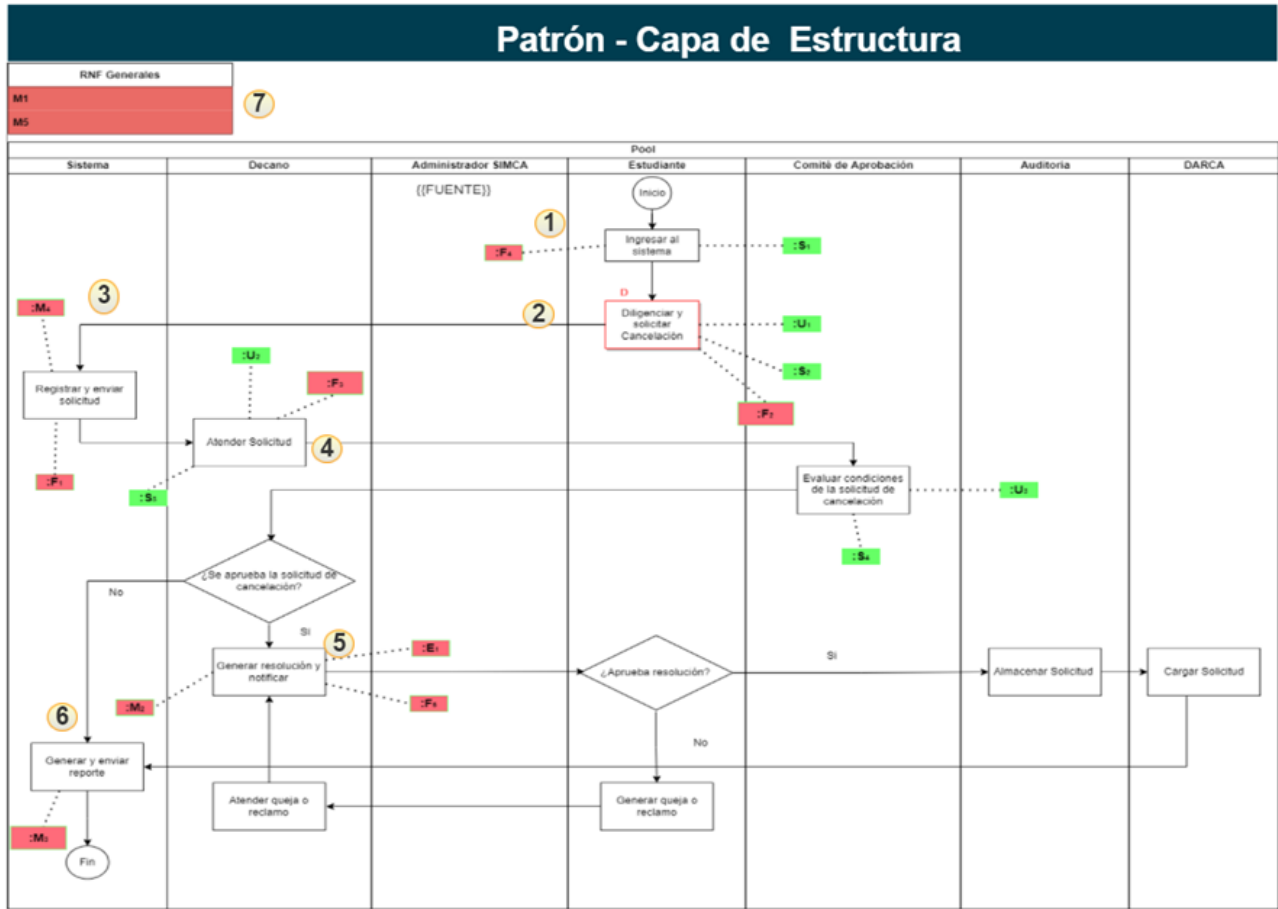
**Tabla 40.** *Sección 1 Instrumento de registro - Estudio de caso*

ID proceso	Nombre proceso	Actividad	ID característica de calidad	Característica de calidad	Dependencia
<b>RNF asociados a las características de calidad de Fiabilidad</b>					
<i>P1</i>	<i>Ingreso al sistema</i>	<i>Login del sistema</i>	<i>F4</i>	<i>Fiabilidad</i>	
<i>P2</i>	<i>Diligenciar y solicitar</i>	<i>Llenar formato</i>	<i>F2</i>	<i>Fiabilidad</i>	
<i>P3</i>	<i>Registrar y enviar solicitud</i>	<i>Enviar solicitud</i>	<i>F1</i>	<i>Fiabilidad</i>	
<i>P4</i>	<i>Atender Solicitud</i>	<i>Revisar Solicitud</i>	<i>F3</i>	<i>Fiabilidad</i>	
<i>P6</i>	<i>Generar Resolución y</i>	<i>Generar resolución</i>	<i>F5</i>	<i>Fiabilidad</i>	
<b>RNF asociados a las características de calidad de Mantenibilidad</b>					
<i>General</i>	<i>No Aplica</i>	<i>No Aplica</i>	<i>M1</i>	<i>Mantenibilidad</i>	
<i>P3</i>	<i>Registrar y enviar solicitud</i>	<i>Validar información del registro</i>	<i>M4</i>	<i>Mantenibilidad</i>	
<i>P6</i>	<i>Generar Resolución y notificar</i>	<i>Notificar al estudiante</i>	<i>M2</i>	<i>Mantenibilidad</i>	
<i>P9</i>	<i>Generar y enviar reporte</i>	<i>Generar reporte</i>	<i>M3</i>	<i>Mantenibilidad</i>	
<i>General</i>	<i>No Aplica</i>	<i>No Aplica</i>	<i>M5</i>	<i>Mantenibilidad</i>	

Fuente: Elaboración propia.

5. Se muestra el modelado final de la capa de estructura incluyendo las características de calidad encontradas con su respectivo color.

**Ilustración 26.** Modelado final de los procesos en la capa de estructura - Estudio de caso



Fuente: Elaboración propia.

**Nota:** En la ilustración 26, adicionalmente se muestran numerados los procesos que contienen los RNF que se desean representar en prototipos de interfaz.

### Capa de Datos

En esta etapa se dan a conocer los resultados de la Capa de Datos que se obtuvieron en el estudio de caso, en donde se presentan cuatro tablas desarrolladas que corresponden a las cuatro secciones de instrumento de registro propuesto para en la guía. Es de aclarar que cada tabla solo contiene la información de doce RNF de los veintitrés especificados, esto por cuestión de extensión del presente documento; la información completa se encuentra disponible en el Anexo 6. En seguida, se muestran las tablas con cada uno de los resultados (ver tablas 41,42,43,44):



**Tabla 41.** Sección 2” RNF” del Instrumento de registro - sección 2 RNF – estudio de caso

ID característica de calidad	Sub Característica (Nivel 2 Grafo)	ID RNF	Atributo de Calidad (nivel 3 Grafo)	Valor del atributo Nivel 3	Justificación
<b>RNF asociados a las características de calidad de Fiabilidad</b>					
F4	Madurez	F4-PRE1	Precisión	No aplica	El sistema debe tener alta precisión en el momento de realizar la autenticación de usuario, validando correctamente los datos enviados para el ingreso.
F2	Disponibilidad	F2-DIS1	Viabilidad Lógica	24 hrs	El sistema debe garantizar su disponibilidad cuando el usuario lo requiera siempre y cuando la solicitud de la cancelación de materia esté dentro del periodo definido anteriormente.
	Madurez	F2-MAD1	Veracidad de datos	Respuesta rápida (ms)	El sistema debe responder ante excepciones o fallos de una solicitud de cancelación antes de ser enviada por el usuario.
F1	Disponibilidad	F1-DIS2	Tiempo entre fallos	1 semana	La disponibilidad del sistema debe ser total, mientras se encuentre dentro de las fechas y horas establecidas
F3	Disponibilidad	F3-DIS1	Tiempo de disponibilidad	24 hrs/día	El sistema deberá garantizar la revisión de solicitudes en cualquier parte del día y en cualquier fecha
F5	Disponibilidad	F5-DIS1	Tiempo de disponibilidad	24 hrs/día	El sistema deberá garantizar la generación de resoluciones 24/7.
<b>RNF asociados a las características de calidad de Mantenibilidad</b>					
M1	Modularidad	M1-MOD1	Dependencia de otros componentes	Dos dependencias	El sistema deberá contar con componentes independientes de tal manera que si el sistema crece en cuanto a funcionalidades, este no se ve afectado por algún cambio externo
M4	Analizabilidad	M4-ANAI	Sencillez	Alta	El sistema debe contar con una implementación de código con alta sencillez, que permita un fácil mantenimiento al futuro
M2	Capacidad para ser modificado	M2-CAPMOD1	Posibilidad de cambiar	alta	El sistema debe permitir modificar los mensajes y el formato de notificación cuando resulte necesario de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
M3	Capacidad para ser probado	M3-CAPPRO1	Comprobabilidad	%de cobertura de código	El sistema deberá permitir establecer un porcentaje alto de cobertura de código por medio de prueba unitaria, que garantice que los resultados de las operaciones sean correctos
M5	resusabilidad	M5-REU1	Documentado	Altamente documentado	El sistema deberá tener de manera explícita y precisa el propósito, estrategia e intención de cada elemento software (métodos, clases, parámetros, interfaces, atributos, etc.)
		M5-REU2	Especificado	No aplica	El sistema deberá especificar cada funcionalidad incluyendo precondiciones y postcondiciones

Fuente: Elaboración propia.

**Tabla 42.** Instrumento de registro - sección 3 Prioridad del RNF – estudio de caso

<b>ID del RNF</b>	<b>Importancia</b>	<b>Urgencia</b>	<b>Intervalo de Tiempo</b>	<b>Valor de la Prioridad</b>	<b>Descripción del valor de prioridad</b>
<b>RNF asociados a las características de calidad de Fiabilidad</b>					
<i>F4-PRE1</i>	<i>x</i>		<i>Presente-pasado</i>	<i>2</i>	<i>Hazlo en segunda instancia</i>
<i>F2-DIS1</i>	<i>x</i>	<i>x</i>	<i>Presente-pasado</i>	<i>1</i>	<i>Hazlo inmediatamente</i>
<i>F2-MAD1</i>	<i>x</i>		<i>Presente-pasado</i>	<i>2</i>	<i>Hazlo en segunda instancia</i>
<i>F1-DIS2</i>	<i>x</i>		<i>Presente-pasado</i>	<i>2</i>	<i>Hazlo en segunda instancia</i>
<i>F3-DIS1</i>	<i>x</i>	<i>x</i>	<i>Presente-pasado</i>	<i>1</i>	<i>Hazlo inmediatamente</i>
<i>F5-DIS1</i>	<i>x</i>		<i>Presente-pasado</i>	<i>2</i>	<i>Hazlo en segunda instancia</i>
<b>RNF asociados a las características de calidad de Mantenibilidad</b>					
<i>M1-MOD1</i>	<i>x</i>	<i>x</i>	<i>Presente-pasado</i>	<i>1</i>	<i>Hazlo inmediatamente</i>
<i>M4-ANA1</i>	<i>x</i>		<i>Presente-pasado</i>	<i>2</i>	<i>Hazlo en segunda instancia</i>
<i>M2-CAPMOD1</i>	<i>x</i>		<i>Presente-pasado</i>	<i>2</i>	<i>Hazlo en segunda instancia</i>
<i>M3-CAPPRO1</i>	<i>x</i>		<i>Presente-pasado</i>	<i>2</i>	<i>Hazlo en segunda instancia</i>
<i>M5-REU1</i>	<i>x</i>		<i>Presente-pasado</i>	<i>2</i>	<i>Hazlo en segunda instancia</i>
<i>M5-REU2</i>	<i>x</i>		<i>Presente-pasado</i>	<i>2</i>	<i>Hazlo en segunda instancia</i>

Fuente: Elaboración propia.

**Tabla 43.** Instrumento de registro - sección 4 Otros elementos del RNF – estudio de caso

ID del RNF	Tipo	Dificultad	Riesgos	Obligatoriedad	Rol	Representación
<b>RNF asociados a las características de calidad de Fiabilidad</b>						
<i>F4-PRE1</i>	<i>Derivado de S1-AUT1</i>	<i>Normal</i>	1. Privacidad de datos violada 2. Acciones no autorizadas por el usuario		<i>Regla o hecho</i>	<i>Cualitativa</i>
<i>F2-DIS1</i>	<i>Derivado</i>	<i>Normal</i>	1. no disponible en lapso de tiempo 2. Disponible en fechas no permitidas	<i>x</i>	<i>Regla o hecho</i>	<i>Cuantitativo</i>
<i>F2-MAD1</i>	<i>Básico</i>	<i>Fácil</i>	1. envío de datos incorrectos	<i>x</i>	<i>Regla o hecho</i>	<i>Cuantitativo</i>
<i>F1-DIS2</i>	<i>Básico</i>	<i>Normal</i>	1. Perdidas de solicitudes de cancelaciones 2. Baja confianza de usuarios	<i>x</i>	<i>Propiedad futura</i>	<i>Cuantitativa</i>
<i>F3-DIS1</i>	<i>Básico</i>	<i>Normal</i>	1. No ingresos del personal 2. Solicitud de cancelaciones vencidas por no revisión a causa de disponibilidad	<i>x</i>	<i>Regla o hecho</i>	<i>Cuantitativo</i>
<i>F5-DIS1</i>	<i>Básico</i>	<i>Normal</i>	1. No envió de resolución, causando confusiones. 2. Generación inoportuna de resolución.	<i>x</i>	<i>Regla o hecho</i>	<i>Cuantitativo</i>
<b>RNF asociados a las características de calidad de Mantenibilidad</b>						
<i>M1-MOD1</i>	<i>Básico</i>	<i>Difícil</i>	1. Retrabajo al agregar nuevas funcionalidades.	<i>x</i>	<i>Propiedades del futuro sistema-</i>	<i>Declarativa</i>
<i>M4-ANAI</i>	<i>Básico</i>	<i>Normal</i>	1. Introducción de nuevo errores en etapas de mantenimiento 2. Sobrecostos en mantenimiento 3. Aumenta la complejidad en la legibilidad del código	<i>x</i>	<i>Regla o hecho</i>	<i>Cualitativa</i>
<i>M2-CAPMOD1</i>	<i>Derivado de M1-MOD1</i>	<i>Normal</i>	1. Introducir nuevos errores al realizar modificaciones. 2. Información incorrecta mostrada a los usuarios. 3. Sobrecostos en los procesos de soporte y mantenimiento.	<i>x</i>	<i>Regla o hecho</i>	<i>Cualitativo</i>
<i>M3-CAPPRO1</i>	<i>Derivado de M4-ANAI</i>	<i>Normal</i>	1. Posibilidad de obtener resultados erróneos.	<i>x</i>	<i>Regla o hecho</i>	<i>Cuantitativo</i>
<i>M5-REU1</i>	<i>Básico</i>	<i>Normal</i>	1. Eleva los costos de analizabilidad y modificabilidad	<i>x</i>	<i>Regla o hecho</i>	<i>Cualitativo</i>
<i>M5-REU2</i>	<i>Básico</i>	<i>Normal</i>	1. Eleva costos de analizabilidad y de capacidad de probar	<i>x</i>	<i>Regla o hecho</i>	<i>Cualitativo</i>

Fuente: Elaboración propia.

**Tabla 44.** Instrumento de registro - sección 5 Datos del RNF – estudio de caso

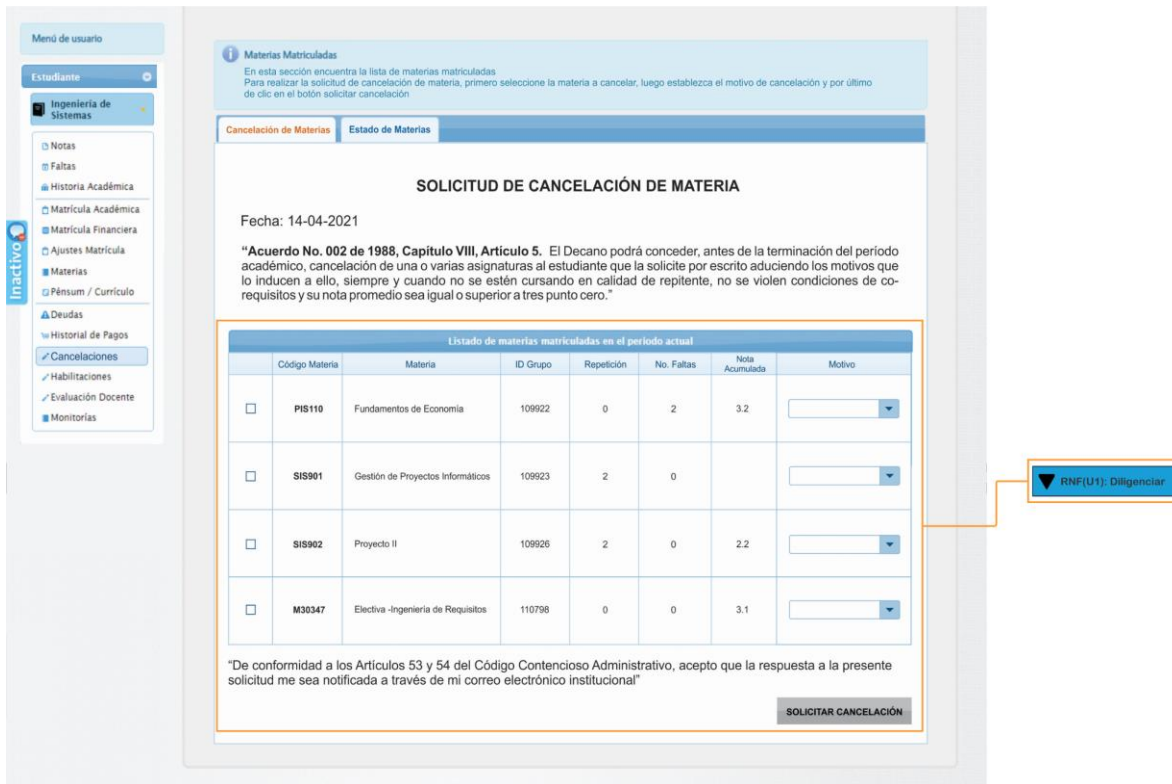
<b>ID del RNF</b>	<b>Elemento</b>	<b>Datos a almacenar</b>	<b>Forma de almacenamiento</b>
<b>RNF asociados a las características de calidad de Fiabilidad</b>			
<i>F4-PRE1</i>	<i>Formulario de Login</i>	<i>Usuario</i>	<i>Modelo de usuario en BD</i>
		<i>Contraseña</i>	<i>Modelo de contraseña en BD</i>
<i>F2-DIS1</i>	<i>Tiempo de disponibilidad</i>	<i>Valor del tiempo</i>	<i>numérico</i>
<i>F2-MAD1</i>	<i>Tiempo de respuesta de excepción o fallo</i>	<i>Valor del tiempo</i>	<i>numérico</i>
<i>F1-DIS2</i>	<i>Tiempo entre fallos</i>	<i>Valor del tiempo</i>	<i>Tipo de dato double en BD</i>
<i>F3-DIS1</i>	<i>Disponibilidad de revisión.</i>	<i>24 hrs/día</i>	<i>numérico</i>
<i>F5-DIS1</i>	<i>Disponibilidad de generación.</i>	<i>24 hrs/día</i>	<i>numérico</i>
<b>RNF asociados a las características de calidad de Mantenibilidad</b>			
<i>M1-MOD1</i>	<i>No aplica</i>	<i>No aplica</i>	<i>No aplica</i>
<i>M4-ANA1</i>	<i>No aplica</i>	<i>No aplica</i>	<i>No aplica</i>
<i>M2-CAPMOD1</i>	<i>No aplica</i>	<i>No aplica</i>	<i>No aplica</i>
<i>M3-CAPPRO1</i>	<i>Cobertura de código</i>	<i>%de cobertura de código</i>	<i>numérico</i>
<i>M5-REU1</i>	<i>No aplica</i>	<i>No aplica</i>	<i>No aplica</i>
<i>M5-REU2</i>	<i>No aplica</i>	<i>No aplica</i>	<i>No aplica</i>

Fuente: Elaboración propia.

## Capa de vista

A continuación, se muestran algunos de los prototipos que se optaron por representar para los RNF asociados a las características de calidad de fiabilidad y mantenibilidad (ver ilustración 27):

**Ilustración 27. Prototipo 1 de usabilidad - estudio de caso**



Fuente: Elaboración propia.

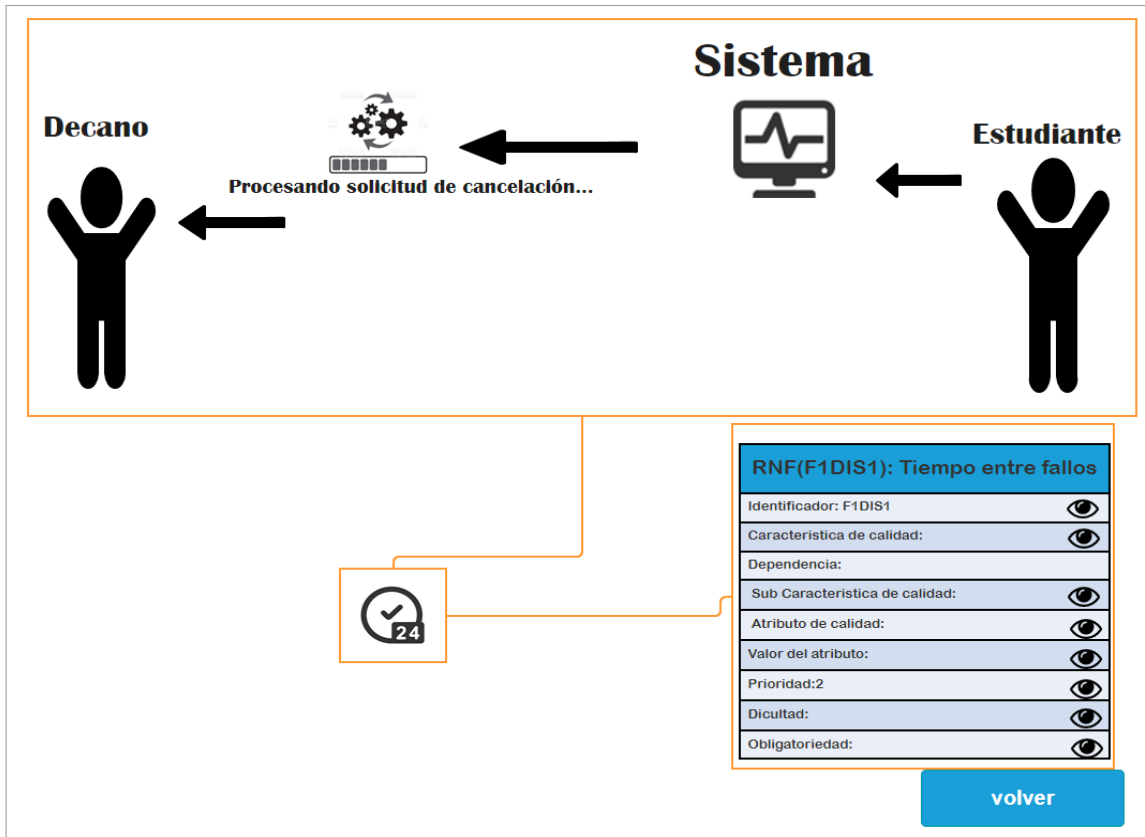
**Ilustración 28. Plantilla descriptiva del prototipo 1 de usabilidad - estudio de caso**

<b>RNF(U1): Diligenciar</b>	
Identificador: U1-	
Característica de calidad:	
Dependencia:	
Sub Característica de calidad:	
Atributo de calidad:	
Valor del atributo:	
Prioridad:	
Dicultad:	
Obligatoriedad:	

Fuente: Elaboración propia.

A continuación, se muestra el prototipo para el RNF de fiabilidad con código F1DIS1 (ver ilustración 29):

**Ilustración 29.** Plantilla descriptiva del prototipo 1 de fiabilidad - estudio de caso



Fuente: Elaboración propia.

Para interactuar de mejor manera con los prototipos y revisar la información a más detalle de cada RNF especificado, por favor ingresar al siguiente link: <https://kiy6e9.axshare.com/#c=2>

#### 4.9. Análisis de los resultados

En el cumplimiento a lo expuesto en el marco metodológico, se presentan el análisis de los datos obtenidos en la aplicación del patrón MDV, teniendo en cuenta las preguntas de investigación planteadas.

**PS1. ¿El patrón MDV permite llegar a especificar de manera completa los RNF de las características de calidad de fiabilidad y mantenibilidad en los instrumentos de captura?**

Con el fin de determinar la cantidad de RNF de fiabilidad y mantenibilidad involucrados en el proyecto que fueron especificados completamente, se realizó una validación en el proceso de recolección de datos descrito en las secciones anteriores con la ayuda de los instrumentos de

captura y de representación en prototipos de interfaz, validando que cada uno de los campos o ejercicios propuestos se diligenciaran completamente. Cabe resaltar que las capas de entendimiento y la capa de estructura no fueron tenidas en cuenta para este análisis, ya que a partir de la propuesta seleccionada por el cliente en la capa de entendimiento se inicia a modelar el proceso en la capa de estructura, y tal modelado resultante debe ser el mismo para todos los participantes.

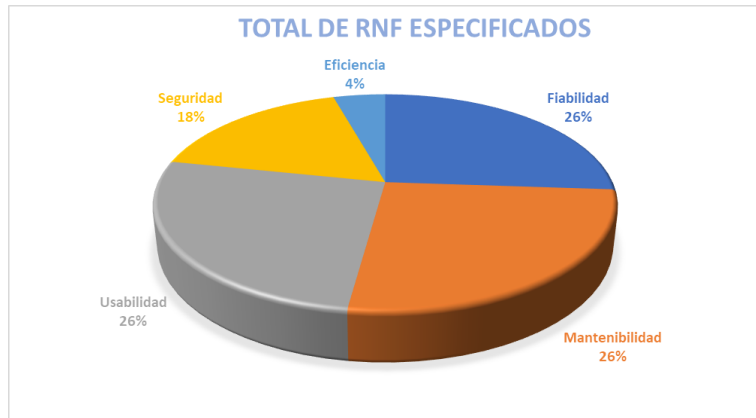
Partiendo de la información registrada en la capa de datos se puede evidenciar que los RNF de fiabilidad y mantenibilidad estudiados en este proyecto son los de mayor número de ocurrencia entre los especificados, inclusive alcanzan un mayor valor que algunos de los RNF que generalmente son los más ocurrentes en proyectos de software, lo que es bueno para el estudio, ya que en la mayoría de los proyectos de *software* no son tan tenidos en cuenta. Lo anterior evidencia que se está aportando al objetivo del estudio (Ver Tabla 45, Ilustración 30). Además, los doce RNF de fiabilidad y mantenibilidad capturados, fueron especificados completamente.

**Tabla 45.** Total, de RNF especificados por característica de calidad

Característica de calidad	Id del RNF	Total
<i>Fiabilidad</i>	<i>F4-PRE1</i>	<b>6</b>
	<i>F2-DIS1</i>	
	<i>F2-MAD1</i>	
	<i>F1-DIS2</i>	
	<i>F3-DIS1</i>	
	<i>F5-DIS1</i>	
<i>Mantenibilidad</i>	<i>M1-MOD1</i>	<b>6</b>
	<i>M4-ANA1</i>	
	<i>M2-CAPMOD1</i>	
	<i>M3-CAPPRO1</i>	
	<i>M5-REU1</i>	
	<i>M5-REU2</i>	
<i>Usabilidad</i>	...	<b>6</b>
<i>Seguridad</i>	...	<b>4</b>
<i>Eficiencia de desempeño</i>	...	<b>1</b>
	<b>Total</b>	<b>23</b>

Fuente: Elaboración propia.

**Ilustración 30.** *Porcentaje de RNF especificados por característica de calidad*



Fuente: Elaboración propia.

El porcentaje de RNF de fiabilidad y mantenibilidad especificados completamente fue considerable respecto a los demás RNF, ya que los artefactos de captura fueron diseñados específicamente con el objetivo de enfatizar en estos dos RNF.

Posteriormente, en la capa de vista del patrón para los RNF de fiabilidad y mantenibilidad que son complejos de representar, se usó la propuesta de representación y los estereotipos propuestos por esta; de esta manera se logró obtener la visualización de once de los doce RNF especificados, lo que corresponde a una completitud del 91.7 %, lo cual evidencia que se logró un alto porcentaje de completitud en los prototipos de la capa de vista. El prototipo del RNF faltante no se logró representar ya que el atributo “*posibilidad para cambiar*” asociado a la característica de calidad de mantenibilidad se consideró complejo para representar y no encontraron estereotipos apropiados que permitieran representarlo.

Finalmente se concluye que, en la capa de datos el 100% de los RNF de fiabilidad y mantenibilidad que corresponde a doce, fueron especificados completamente, luego en la capa de vista se dio una completitud de 91.7% que equivale a representar once de los doce RNF. Por tanto, se puede concluir que se lograron especificar casi en su totalidad los RNF identificados.

**PS2. ¿El patrón MDV permite llegar a especificar de manera correcta los RNF de las características de calidad de fiabilidad y mantenibilidad en los instrumentos de captura?**



Para determinar la cantidad de RNF de fiabilidad y mantenibilidad involucrados en el proyecto que fueron especificados correctamente, se validó que la información registrada en cada uno de los campos propuestos en los instrumentos de captura, se diligenciaran correctamente frente a lo definido en la guía de aplicabilidad del patrón.

La capa de entendimiento y estructura no se tendrán en cuenta para el presente análisis, ya que estas se realizaron de manera transversal entre observadores y analistas; además el modelado obtenido en la capa de estructura se refinó entre los dos grupos para lograr uniformidad en este. Partiendo de este mismo, cada grupo realizó el diligenciamiento de los campos de la capa de Datos para los RNF de fiabilidad y mantenibilidad. A continuación, se muestra la información encontrada donde se compara el número de campos contenidos en la capa de datos vs el número de campos diligenciados correctamente (ver tabla 46).

**Tabla 46.** Campos diligenciados correctamente en la capa de datos

Capa del patrón	Id del RNF	Nro de campos de la capa	Nro de campos diligenciados correctamente	Porcentaje de correctitud
Capa de datos	<b>RNF de Fiabilidad</b>			
	F4-PRE1	25	25	<b>96,7%</b>
	F2-DIS1	25	23	
	F2-MAD1	25	24	
	F1-DIS2	25	24	
	F3-DIS1	25	24	
	F5-DIS1	25	25	
	<b>Total</b>	<b>150</b>	<b>145</b>	
	<b>RNF de Mantenibilidad</b>			
	M1-MOD1	25	25	<b>95,3%</b>
	M4-ANA1	25	23	
	M2-CAPMOD1	25	24	
	M3-CAPPRO1	25	22	
	M5-REU1	25	24	
M5-REU2	25	25		
<b>Total</b>	<b>150</b>	<b>143</b>		

Fuente: Elaboración propia.

Para verificar la correctitud respecto a los resultados obtenidos en la capa de vista, se tuvo en cuenta dos criterios: que los prototipos realizados por los participantes no estén tan alejados de lo que se propone en la guía de aplicabilidad y que se construyan con los elementos básicos que son la plantilla descriptiva, sección asociada a la plantilla y por ultimo los elementos que contengan

tal sección. Tales elementos debes ser coherentes con el atributo de calidad asociado al RNF en cuestión y con la justificación de capa de datos. Adicionalmente se deben usar los estereotipos propuestos, si se usó un estereotipo distinto, se adicionará como propuesta del participante, para ser usado en futuros proyectos. A continuación, se muestra el análisis de los resultados de la capa de vista en términos de porcentaje de correctitud para los RNF de fiabilidad y mantenibilidad (ver tabla 47).

**Tabla 47. Porcentaje de correctitud de los RNF en la capa de vista**

Capa del patrón	Id del RNF	Plantilla descriptiva (30%)	Sección asociada(10%)	Elementos de la sección(60%)	Porcentaje de correctitud por RNF	Porcentaje total de correctitud
Capa de vista	<b>RNF de Fiabilidad</b>					
	F4-PRE1	100%	100%	80%	88%	89,5%
	F2-DIS1	100%	100%	80%	88%	
	F2-MAD1	90%	100%	80%	85%	
	F1-DIS2	100%	100%	90%	94%	
	F3-DIS1	100%	100%	80%	88%	
	F5-DIS1	100%	100%	90%	94%	
	<b>RNF de Mantenibilidad</b>					
	M1-MOD1	100%	100%	90%	94%	85,2%
	M4-ANA1	100%	100%	80%	88%	
	M2-CAPMOD1	100%	80%	20%	50%	
	M3-CAPPRO1	100%	100%	80%	88%	
	M5-REU1	90%	100%	100%	97%	
M5-REU2	100%	100%	90%	94%		
<b>Total</b>						<b>87,3%</b>

Fuente: Elaboración propia.

La tabla muestra que se obtuvo una correctitud total del 87.3% entre los RNF de fiabilidad y mantenibilidad que fueron representados, teniendo en cuenta los criterios definidos para el análisis de los elementos que componen el prototipo. Para lograr el valor de correctitud total se le dio un peso a manera de porcentaje a cada elemento que constituye el prototipo con base en la dificultad que este conlleva.

Observando el porcentaje de correctitud total de la capa de datos que fue de 96% y el porcentaje de correctitud total de la capa de vista que fue de 87.3%, finalmente, se llega a la conclusión que tanto en la capa de datos como en la de vista, se alcanzó un alto porcentaje de correctitud promedio que es de 91,6% aproximadamente, lo que se considera relevante para cumplir con el objetivo de la presente pregunta de investigación.

**PS3. ¿El patrón MDV permite especificar los RNF de las características de calidad de fiabilidad y mantenibilidad provenientes del cliente de manera práctica?**

Para determinar si el patrón MDV es práctico en el momento de especificar los RNF de fiabilidad y mantenibilidad involucrados en el proyecto, se consideró el tiempo promedio de especificación y prototipado de cada RNF usando como apoyo los instrumentos de registro y propuestas de representación, para las capas de datos y vista respectivamente. Los resultados o hallazgos encontrados después de la aplicación del patrón se muestran a continuación.

**Tabla 48. Métricas obtenidas a través del uso del patrón - Estudio de caso**

Métricas obtenidas a través del uso del patrón	
Métricas	Estudio de caso
Tiempo promedio para especificar un RNF en la capa de Datos	<b>8 min</b> (promedio) en donde se usó: 1 min para identificar característica de calidad + 1 min para <i>subcaracterísticas</i> de calidad + 6 min para diligenciar los campos restantes de la capa)
Tiempo promedio para representar un RNF en la capa de vista	<b>7 min</b> (promedio) para representar cada RNF en la herramienta de prototipado AXURE RP

Fuente: Elaboración propia.

A partir de los datos presentados en la tabla 48, se puede evidenciar que para especificar RNF de fiabilidad y mantenibilidad en la capa de datos, se toma un tiempo aproximado de ocho minutos siendo este un hallazgo en la presente investigación. Adicionalmente, se encontró que para representar los RNF en una aplicación de prototipado aplicando la propuesta de representación del patrón, el tiempo aproximado que se requiere para desarrollar tal actividad es de siete minutos. Para apoyar este hallazgo, se usó como referente de tiempos de especificación de RNF en el proceso de cancelación de materias de la Universidad de Cauca según Buitrón (2021), En la tabla 48 se muestran los siguientes elementos: “la cantidad de prototipos para este proyecto, el promedio de tiempo en minutos de la aplicación en los prototipos por proyecto, así como el total

de RNF especificados en cada proyecto” (p. 243), en donde el proyecto cuatro pertenece proceso de solicitud de cancelaciones de materias en SIMCA.

**Tabla 49.** Total de RNF por proyecto y su promedio de tiempo de especificación por prototipo

<b>Proyectos</b>	<b>Cantidad de prototipos</b>	<b>Promedio de tiempo por prototipo</b>	<b>Total, RNF</b>
Proyecto 1	2	38	15
Proyecto 2	9	70	250
Proyecto 3	17	126	91
<b>Proyecto 4</b> Proyecto 5	48 (24 cada proyecto)	46	359 (promedio)

Fuente: (Buitrón, 2021)

En conclusión, los tiempos promedios de especificación en la capa de datos y representación en la de vista obtenidos por los analistas, reflejan el nivel de practicidad con la que estos alcanzaron al utilizar los instrumentos de especificación del patrón, obteniendo la especificación total de los RNF involucrados por una parte en la capa de datos y por otra parte en la capa de vista, donde se obtienen tiempos similares para tal especificación, comparando los presentes hallazgos con los obtenido por Buitrón (2021). Además, por medio de la encuesta se aporta información útil en cuestión de completitud, correctitud, practicidad y el nivel de conocimiento obtenido en el proceso que llevaron las personas de acuerdo con su percepción. Esta información puede ser consultada a continuación (ver tabla 49).

### **Nivel de conocimiento obtenido por los participantes durante estudio de caso**

En la tabla 50 se presentan los resultados obtenidos de las respuestas de los participantes, en donde se evaluó el nivel de conocimiento adquirido por los participantes y se muestran a continuación:

**Tabla 50.** Promedio de evaluación de los aspectos en la encuesta

Participante	Valor promedio de Aspecto de completitud	Valor promedio de Aspecto de correctitud	Valor promedio de Aspecto de practicidad
Analista 1	4.5	4.5	4.63
Analista 2	4.5	4	4

Fuente: Elaboración propia.

**Tabla 51.** Nivel de conocimiento adquirido por los participantes

Analista	Nivel de conocimiento	Nivel de conocimiento acerca de especificación de RNF de Fiabilidad y Mantenibilidad antes de Usar el Patrón	Nivel de conocimiento acerca de especificación de RNF de Fiabilidad y Mantenibilidad después de usar el patrón
Analista 1		Rango 3, entre 50% y 70%	Rango 4, entre 70% y 100%
Analista 2		Rango 3, entre 50% y 70%	Rango 4, entre 70% y 100%

Fuente: Elaboración propia.

**Tabla 52.** Resultados de las preguntas realizadas en la encuesta a los participantes

Aspecto de respuesta de tipo SI/NO	Numero de respuestas marcadas en SI	Numero de respuestas marcadas en NO
¿Considera que los elementos definidos en el patrón son suficientes?	2	
¿Considera que los pasos presentados en la guía para aplicar el patrón son suficientes?	2	
¿Considera que se debe agregar algún elemento en el diseño del patrón?	1	1

Fuente: Elaboración propia.

## Mejoras propuestas para el patrón

- Dentro de la capa de entendimiento inicial o general del problema, se encuentra el ítem **técnicas previas** usadas para la identificación del problema. Este tenía preestablecidas algunas técnicas, las cuales se brindaban al analista para que las use y que este adquiriera experiencia al aplicar el patrón. Una de las mejoras que se proponen para esta capa es flexibilizar el ítem mencionado anteriormente dejando que el analista elija la técnica o una combinación de ellas, para la identificación del problema. Con esto se pretende buscar la comodidad del analista, evitando que este haga mal uso de la técnica seleccionada por desconocimiento.
- En el paso cinco de la capa de estructura, se definió que la dependencia podría ser solo un RNF ósea dependencia simple. Luego de aplicar el patrón, se propone la mejora de una dependencia múltiple, que consiste en que dos o más RNF pueden ser dependencia y a la vez priorizados, de acuerdo con la experiencia del analista y los intereses más importantes de la organización.
- En la capa de estructura se efectuó un análisis donde se encontró que algunas características de calidad se ven involucradas en todo el producto en general, esto significa que afecta a todo el producto y no a un solo proceso, por lo cual se propone incorporar una sección de características de calidad generales.

### 4.10. Plan de validez

Con el objetivo de tratar las amenazas y evitar el sesgo dentro del plan de validez, se consideraron los siguientes aspectos:

- **Diseño del estudio de caso:** El diseño del presente estudio de caso fue basado en el protocolo planteado por Yin, R.K (2006), como metodología y guía para el presente estudio.
- **Validez de constructo:** Para las preguntas establecidas en el estudio de caso se han definido medidas objetivas, verificables y trazables (ver tabla 36) que podrían permitir la obtención de información fiable acerca de los criterios de correctitud, completitud y practicidad del patrón de especificación en procesos de desarrollo de software. Se han recolectado datos generados por diferentes sujetos de investigación a partir de encuestas (ver anexo 4), documentos de registro y observación participativa. Además, el uso de los

instrumentos relacionados con cada actividad realizada con los participantes permitió tener evidencia de las preguntas de investigación, datos recolectados y análisis.

- **Validez interna:** Los artefactos de registro o recolección de datos fueron diseñados para propósitos específicos. La lista de chequeo se propuso para capturar información de manera rápida y clara en las reuniones con el cliente participante y la guía de aplicabilidad de patrón se realizó como apoyo para guiar al analista en la ejecución del patrón, todo lo anterior con el propósito de facilitar la aplicación de este.
- **Validez Externa:** Con el propósito de maximizar los resultados los integrantes del grupo de investigación observaron a detalle el ejercicio de aplicación del patrón por parte de los participantes y en ocasiones solo si era necesario que se resolvieran algunas dudas e inquietudes sobre actividades del proceso de aplicación. Una dificultad que se identificó en el proceso fue que solo se dispuso de una reunión con el cliente para capturar los requerimientos no funcionales. En caso de tener información faltante no se pudo volver a validar con el cliente por cuestiones de disponibilidad.

#### 4.11. Limitaciones

- La pandemia del COVID 19 no permitió realizar las reuniones de manera presencial, lo que conlleva a tener dificultades en la facilidad de comunicación con el cliente, esto significa que no hubo la posibilidad de validar y de recibir retroalimentación de los sketch físicos por parte de este.
- El número de personas que participaron en este estudio de caso fue reducido, lo que puede ser un inconveniente debido a que el tamaño de la muestra es muy pequeño para la recolección de la información.
- El tiempo con el que se contó para realizar las reuniones fue limitado, lo que impidió una profundización en algunos elementos capturados. Adicionalmente por razones académicas los analistas contaron con tiempo reducido para aplicar el patrón.

#### 4.12. Lecciones aprendidas

- Algunos de los pasos propuestos para aplicar el patrón MDV, resultaron ser un poco confusos, para lo cual se requirió una explicación adicional por parte de los integrantes del

grupo de investigación. Adicionalmente, estos pasos fueron considerados para redactarlos de manera comprensible para el participante.

- Se realizó una adaptación de la guía de aplicabilidad para capturar la información de manera rápida y eficaz en el momento de su aplicación con el cliente, por medio del documento de lista de chequeo. Esto debido a la extensión del documento, ya que requería de mucho tiempo y esfuerzo aplicarlo de la manera inicial.
- Algunas de las sugerencias por parte de los participantes, dieron lugar a mejoras que fueron tomadas en cuenta y se aplicaron en la nueva versión del patrón, como es el caso de la sección general para RNF que se vean reflejados en todo el proyecto, algunos nuevos estereotipos de representación para RNF de fiabilidad y mantenibilidad, etc.
- Los investigadores y analistas adquirieron experiencia en el manejo de reuniones para la captura de requerimientos, en cuanto al manejo del tiempo por pregunta y del cliente, ya que en determinadas situaciones o circunstancias el cliente suele demorar mucho con la respuesta de las preguntas realizadas.
- En el proceso de aplicación del patrón posterior a la captura de requerimientos con el cliente, es importante brindar un acompañamiento constante a los analistas con el objetivo de orientar en el proceso de aplicación y resolver cualquier duda que surja.
- En el proceso creativo al aplicar las propuestas de prototipado se notó un poco confundidos a los analistas ya que es un proceso complejo y nuevo, por tanto, es importante tener actualizado un repositorio donde se almacenen los proyectos que aplicaron el patrón, con el objetivo de tener más ejemplos e ideas para los proyectos nuevos.



## Capítulo 5. Conclusiones, lecciones aprendidas y trabajo futuro.

### 5.1. Conclusiones

El presente trabajo investigativo tuvo como objetivo proponer un patrón para la especificación de RNF, mediante el cual se pudiera lograr capturar dichos requisitos y quedaran reflejados en un prototipo de interfaz; esto quiere decir que se exploró una parte de la especificación de requisitos, que, justificados a través de la revisión de la literatura, se evidenció la omisión de la especificación de RNF en la mayoría de los desarrollos de *software*.

Algunos puntos claves que permitieron la conclusión de este trabajo investigativo fueron:

- ✓ A través de la investigación se pudo evidenciar que hay un número reducido de trabajos enfocados en aplicación de un patrón como estrategia de especificación y representación de RNF.
- ✓ La caracterización realizada tanto para patrones como para requisitos, permitió identificar los elementos más importantes de estos dos conceptos, con el fin de elegir cuáles de los elementos identificados iban a ser utilizados para el desarrollo de la propuesta del “patrón para la especificación”.
- ✓ El uso de iconografía, a manera de estereotipos, como complemento de representación de RNF y la existencia de un repositorio que almacena estos mismos, fortalecen uno de los conceptos más importantes del patrón, como lo es la reutilización de buenas soluciones, generando a su vez un aporte importante a la representación de los RNF en prototipos de interfaz.
- ✓ Al integrar elementos como la lista de chequeo junto con los artefactos de preguntas, contribuyeron a la eficiencia de captura y especificación de RNF en reuniones con el cliente, con el fin de hacerlo más práctico.
- ✓ De los trabajos encontrados y analizados, se evidencia poco trabajo en temas como la especificación de RNF de fiabilidad y mantenibilidad, junto con estrategias de representación de estos RNF. Por tanto, la presente propuesta realizada en este trabajo de grado contribuye en esos aspectos.
- ✓ La encuesta realizada sobre el patrón MVC, permitió obtener y adoptar aspectos clave del patrón más popular usado actualmente, como lo fue el sistema de capas el cual permite agrupar los elementos de acuerdo a su papel en el patrón.

- ✓ La aplicación del patrón en un proyecto real, permitió ver más allá de la teoría, esto en cuanto a establecer algunos conceptos, reforzar algunos enfoques y completar como tal el patrón.
- ✓ En la evaluación se muestra como resultado que la idoneidad del patrón para realizar una especificación de RNF por parte de los analistas es prometedora, ya que se evidenció que los RNF más omitidos en los proyectos software, alcanzaron un alto nivel de especificación, en términos de completitud, correctitud y practicidad.
- ✓ El patrón de especificación, es una buena estrategia para que los RNF de fiabilidad y mantenibilidad sean considerados constantemente en los proyectos software.

## 5.2. Lecciones aprendidas

En la ejecución de este trabajo de investigación se lograron adquirir las siguientes experiencias:

- En la etapa de elaboración de la guía se realizó un proceso creativo que permitió fortalecer cualidades fundamentales para un ingeniero de sistemas, como son la capacidad creativa, razonamiento lógico, entre otras.
- Una investigación donde solo se tenga como referente la literatura, omite varios aspectos que solo pueden ser captados en entornos prácticos o reales, haciendo que estos aspectos sirvan como retroalimentación para realizar mejoras en un trabajo investigativo; por consiguiente, este tipo de investigaciones deben ser en lo posible complementadas mediante aplicaciones de ejemplo, *Focus Group* y estudios de caso.
- Para llevar a cabo de manera adecuada un estudio de caso, se necesita tener una disponibilidad plena por parte de los involucrados, puesto que estos deberán realizar todas las actividades propuestas y de esta manera aportar el conocimiento obtenido a la investigación.
- La situación de pandemia COVID-19, la cual se presentó durante el desarrollo del trabajo de investigación hizo que la interacción con los directores e involucrados fuera realizada de manera virtual, obligando a utilizar diferentes herramientas informáticas en las distintas actividades que comprendió el trabajo investigativo.
- Cabe resaltar que durante la investigación se adquirió experiencia en el manejo de herramientas de modelado de procesos y prototipado como los *Draw.io* y *Axure*, estas

facilitaron la construcción de procesos de modelado y creación de mockups. Al mismo tiempo se logró interpretar y analizar normas internacionales.

### **5.3. Trabajos futuros**

En el presente trabajo de investigación se han analizado algunos puntos que pueden ser tenidos en cuenta para trabajos futuros:

- Automatizar la guía del patrón propuesta, por medio del desarrollo de un servicio web, que permita la creación de proyectos que se quieran especificar, dentro de los cuales se permita el ingreso de la información respectiva de cada capa. Esta información de cada proyecto sería almacenada en un repositorio, sirviendo de ayuda para proyectos de especificación futuros, donde el analista puede reutilizar soluciones anteriores. De esta manera, se fortalecerá aún más el concepto de patrón para la presente investigación. Cabe resaltar que se cuenta con una parte del desarrollo mencionado anteriormente, pero por cuestiones de tiempo y alcance no se finalizó (ver anexo 5).
- Realizar una posterior investigación que tenga como objetivo fortalecer la forma de representación de los RNF de fiabilidad y mantenibilidad en prototipos de interfaz, y que, adicionalmente, incluya el refinamiento de los estereotipos propuestos en la actual tesis.
- Realizar un artículo científico que muestre los resultados de la investigación.

## Referencias Bibliográficas

- Adams, K.M. (2015). Nonfunctional Requirements. *Systems Analysis and Design*. 28: Springer.
- Alexander, C. (1964). *Notes on the Synthesis of Form*. Harvard University Press.
- Ali, I., & Asif, M. (2018). Applying security patterns for authorization of users in IoT based applications. En *2018 International Conference on Engineering and Emerging Technologies (ICEET)* (pp. 1-5). IEEE.
- Arévalo, O., Linares, S., Correa, L., Parra, J., & González, H. (2008). IEEE-STD-830-1998: práctica recomendada para las especificaciones de requisitos del software. *Univ. Nac. Colomb*
- Arias, C. & Rosero, F (2021). Guía para visibilizar requisitos no funcionales en artefactos técnicos del área de desarrollo de requisitos.
- Arnold, W., Eilam, T., Kalantar, M., Konstantinou, A. V., & Totok, A. A. (2007). Pattern based SOA deployment. En *International Conference on Service-Oriented Computing* (pp. 1-12). Springer, Berlin, Heidelberg.
- Bajpai, V. and R.P. Gorthi. (2012). On non-functional requirements: A survey. IEEE Students'
- Bajpai, V., & Gorthi, R. P. (2012). On non-functional requirements: A survey. In *2012 IEEE Students' Conference on Electrical, Electronics and Computer Science* (pp. 1-4). IEEE.
- Beck, K. (1997). *Smalltalk Best Practice Patterns*. Volume 1: Coding. Prentice Hall, Englewood Cliffs, NJ.
- Berczuk, S. P., Berczuk, S., & Appleton, B. (2003). *Software configuration management patterns: effective teamwork, practical integration*. Addison-Wesley Professional.
- Blancarte, O. (2016). *Introducción a los Patrones de Diseño*. México, México DF.
- Braga, R. T. V., Germano, F. S., & Masiero, P. C. (1998). *A confederation of patterns for resource management*. Procs. of PLoP'98.
- Breitman, K. K., Leite, J. C. S., & Finkelstein, A. (1999). The world sa stage: a survey on requirements engineering using a real-life case study. *Journal of the Brazilian Computer Society*, 6(1), 13-37.
- Brereton, P., et al. (2008). Using a Protocol Template for Case Study Planning. in EASE. Citeseer.

- Brown, D. M. (2010). *Communicating design: developing web site documentation for design and planning*. New Riders.
- Buitrón-Ruiz, S. L., Flores-Ríos, B. L., & Pino-Correa, F. J. (2018). Non-Functional Requirements Tendering Based on Knowledge Management: the Merlinn Framework. *Revista Ingenierías Universidad de Medellín*, 17(32), 155-178.
- Buitrón-Ruiz, S. L., & Pino-Correa, F. J. (2021, noviembre). Framework Para La Elicitación De Requisitos No Funcionales Usando Su Representación. *Universidad del Cauca*.
- Buschmann, F., Henney, K., & Schmidt, D. C. (2007). *Pattern-oriented software architecture, on patterns and pattern languages*. 5. John wiley & sons.
- Cao, J., Zieba, K., & Ellis, M. (2015). *The Ultimate guide to prototyping*. E-kirja. UXPin.
- Chung, L., & do Prado Leite, J. C. S. (2009). On non-functional requirements in software engineering. In *Conceptual modeling: Foundations and applications* (pp. 363-379). Springer, Berlin, Heidelberg.
- Clancy, T. (2014). The standish group chaos report. *Project Smart*, 8-9.
- Coad, P., North, D., & Mayfield, M. (1995). *Object models: strategies, patterns, applications*. Yourdon Press.
- Cysneiros, L. M., & do Prado Leite, J. C. S. (2004). Nonfunctional requirements: From elicitation to conceptual models. *IEEE transactions on Software engineering*, 30(5), 328-350.
- Dabous, F. T., Rabhi, F. A., & Yu, H. (2004). Using software architectures and design patterns for developing distributed applications. In *2004 Australian Software Engineering Conference. Proceedings*. (pp. 290-299). IEEE.
- DeMarco, T. (1979). Structure analysis and system specification. En *Pioneers and Their Contributions to Software Engineering* (pp. 255-288). Springer, Berlin, Heidelberg.
- Durán, A., Bernárdez, B., Toro, M., Corchuelo, R., Ruiz, A., & Pérez, J. (1999). Expressing customer requirements using natural language requirements templates and patterns. En *IMACS/IEEE CSCC'99 Proceedings*.
- Fowler, M. (1997). *Analysis patterns: reusable object models*. Addison-Wesley Professional.
- Franch, X., & Botella, P. (1998). Putting non-functional requirements into software architecture. En *Proceedings Ninth International Workshop on Software Specification and Design* (pp. 60-67). IEEE.

- Franch, X. (1998). Systematic formulation of non-functional characteristics of software. En *Proceedings of IEEE International Symposium on Requirements Engineering: RE'98* (pp. 174-181). IEEE.franch
- Gamma, E., Johnson, R., Helm, R., Johnson, R. E., & Vlissides, J. (1995). *Design patterns: elements of reusable object-oriented software*. Pearson Deutschland GmbH.
- Garzás, J., Pino, F. J., Piattini, M., & Fernández, C. M. (2013). A maturity model for the Spanish software industry based on ISO standards. *Computer Standards & Interfaces*, 35(6), 616-628.
- González-Zúñiga, D., Granollers, T., & Carrabina, J. (2015). Sketching Stereoscopic GUIs with HTML5 Canvas. En *International Conference on Ubiquitous Computing and Ambient Intelligence* (pp. 289-296). Springer, Cham.
- Houde, S. y C. Hill. (1997). What do prototypes prototype? in Handbook of human-computer interaction. *Elsevier*. p. 367-381.
- ISO. Normalización, O.I.d., *ISO 25000*. System and Software Quality Requirements and Evaluation.
- ISO. (2013). *ISO / IEC 25010: 2011 Systems and software engineering \_ Systems and software Quality Requirements and Evaluation ( SQuaRE )@\_ System and software quality models*.
- Ketabchi, S., Sani, N. K., & Liu, K. (2011). A norm-based approach towards requirements patterns. En *2011 IEEE 35th Annual Computer Software and Applications Conference* (pp. 590-595). IEEE.
- Lauretto, M. D. S., Nakano, F., Pereira, C. A. D. B., & Stern, J. M. (2012, October). *Intentional sampling by goal optimization with decoupling by stochastic perturbation*. In *AIP Conference Proceedings (Vol. 1490, No. 1, pp. 189-201)*. American Institute of Physics.
- Leblanc, T. (2015). Sketching as a thinking process. En *DS 82: Proceedings of the 17th International Conference on Engineering and Product Design Education (E&PDE15), Great Expectations: Design Teaching, Research & Enterprise, Loughborough, UK, 03-04.09.2015*.
- Maiti, R. R., & Mitropoulos, F. J. (2015). Capturing, eliciting, predicting and prioritizing (CEPP) non-functional requirements metadata during the early stages of agile software development. In *SoutheastCon 2015* (pp. 1-8). IEEE.

- Martin, R. C. (1996). Granularity. *C++ Report*, 8(10), 57-62.
- Miller, R.E. (2009). *The Quest for Software Requirements*: MavenMark Books.
- Pandey, D., Suman, U., & Ramani, A. K. (2010). An effective requirement engineering process model for software development and requirements management. En *2010 International Conference on Advances in Recent Technologies in Communication and Computing* (pp. Pino, F.J., M. Piattini, and G. Horta Travassos. (2013). Managing and developing distributed research projects in software engineering by means of action-research. (68): p. 61-74.
- Pecht, M. (Ed.). (2009). *Product reliability, maintainability, and supportability handbook*. CRC Press.
- Preece, J., Rogers, Y., Sharp, H., & Benyon, D. (1994). Simon Holland et Tom Carey. *Human-computer Interaction*. Addison-Wesley Longman Ltd.
- Rosanigo, Z. B. (2000). Maximizando reuso en software para Ingeniería Estructural: Modelos y Patrones.
- Rodríguez, M., & Piattini, M. (2014). Entorno para la Evaluación y Certificación de la Calidad del Producto Software. *XIX Jornadas de Ingeniería del Software y Bases de Datos JISBD'2014*, 163-176.
- Ross, D.T., K. Schoman, & IEEE. (1977). *Structured analysis for requirements definition*. (1): p. 6-15.
- Rosson, M.B., J.M. Carroll, y N. Hill. (2002). *Usability engineering: scenario-based development of human-computer interaction*: Morgan Kaufmann.
- Saltiveri, T.G., J.L. Vidal, & J.J.C. Delgado. (2011). *Diseño de sistemas interactivos centrados en el usuario*: Editorial UOC
- Spinellis, D. (2006). *Code quality: the open source perspective*. Adobe Press.
- Sommerville, I. (2005). *Ingeniería del software*: Pearson educación
- Supakkul, S., & Chung, L. (2010). Visualizing non-functional requirements patterns. En *2010 Fifth International Workshop on Requirements Engineering Visualization* (pp. 25-34). IEEE.
- Villanueva, I., Sánchez, J., & Pastor, Ó. (2005.). Elicitación de requisitos en sistemas de gestión orientados a procesos. En *WER* (pp. 38-49). 287-291). IEEE.
- Withall, S. (2007). *Software requirement patterns*: Pearson Education

Yin, B., & Jin, Z. (2012). Extending the problem frames approach for capturing non-functional requirements. En *2012 IEEE/ACIS 11th International Conference on Computer and Information Science* (pp. 432-437). IEEE.

Yépez, W. L. P., & Hurtado, J.A. (2015). *Patrones de diseño: construyendo aplicaciones flexibles y reutilizables*. Editorial Universidad del Cauca.

2006, Gliffy 2007, Jumpchart 2008, Balsamiq 2008, Protoshare 2008 y Justinmind