

**ELICITACION Y ESPECIFICACION DE REQUERIMIENTOS  
NO FUNCIONALES EN APLICACIONES WEB**

**Silvana del Valle, Rojo**

Director/es:  
**Alejandro Oliveros**

**“Tesis presentada para obtener el grado de Magister en  
Ingeniería de Software”**

**Facultad de Informática – Universidad Nacional de La Plata**

**Junio, 2013**

## Resumen

Este trabajo de tesis presenta lineamientos específicos para la aplicación de Ingeniería de Requerimientos Web al tratamiento de los RNF. Presenta procesos de Elicitación y Especificación de RNF Web y ofrece las plantillas diseñadas particularmente para la captura y especificación de los RNF en los procesos definidos.

Si bien actualmente la Ingeniería de Requerimientos proporciona numerosas técnicas y herramientas para identificar, describir, validar y gestionar requerimientos, estos no son aplicados muy a menudo, en el desarrollo de aplicaciones Web. La madurez del proceso de ingeniería de requerimientos parece ser insuficiente, esa insuficiencia demanda nuevos enfoques o evolución de los actuales para el tratamiento adecuado de los RNF.

Los RNF de los sistemas software no son un todo homogéneo, hay falta de consenso para responder a los interrogantes: *¿Qué son? ¿Cómo se clasifican dentro del contexto de desarrollo de software? y ¿Cómo se clasifican en el desarrollo de aplicaciones Web?*

Este trabajo parte de una revisión del estado de arte de los conceptos de RNF en la literatura de la Ingeniería de Requerimientos y establece como bases conceptuales que los RNF son requerimientos de calidad y son restricciones. Luego ofrece un análisis comparativo de seis enfoques existentes de desarrollo de aplicaciones Web para estudiar qué tratamiento aplican a los RNF, en el cual se pudo determinar que las metodologías de aplicaciones Web estudiadas: *Contemplan los RNF, aunque se carece de consenso en su significado. No disponen de técnicas específicas para la elicitación de RNF, ni lineamientos uniformes para la especificación y la validación. No brindan técnicas que soporten el tratamiento de los RNF a lo largo del ciclo de vida. Hay carencia de consenso para establecer en qué fase del ciclo de vida de desarrollo se identifican los RNF. Hay RNF que no son identificados durante la fase de relevamiento y análisis.*

A partir de este análisis y dada la importancia de los RNF, se proponen procesos para la elicitación y especificación de los RNF Web (de calidad y restricciones), se describen las plantillas propuestas que dan soporte a los procesos y finalmente se presenta la validación de los conceptos y técnicas propuestos mediante un caso de estudio en un proyecto real.

*Palabras claves:* Ingeniería de Requerimientos, Requerimientos No Funcionales, Aplicaciones Web, Calidad, Restricciones, Procesos de elicitación y especificación de RNF Web, Matriz de RNF.

## Abstract

This thesis introduces specific guidelines to apply Web Requirements Engineering to the treatment of Non-Functional Requirements. It introduces the process of Elicitation and Specification for Web Non-Functional Requirements and offers spreadsheets designed to capture and specify Non-Functional Requirements in the defined processes.

While at present Requirement Engineering has numerous techniques and tools to identify, describe, validate and manage requirements, these are not frequently applied in the development of Web applications. The maturity of the Requirements Engineering process seems insufficient; this insufficiency demands new approaches or the evolution of current process to the correct treatment of Non-Functional Requirements.

Non-Functional Requirements of software systems are not homogenous; there is a lack of consensus to answer the following questions: *What are Non-Functional Requirements? How they are classified within the context of software development? and How the Non-Functional Requirements are classified within the development of web software application?*

This thesis starts with a revision of the state of art of the concept of Non-Functional Requirements in Requirements Engineering literature and establishes as conceptual foundations that Non-Functional Requirements are quality requirements and restrictions. Then, there is a comparative analysis of six different existing approaches in Web application development to study what treatment applies to Non-Functional Requirements, in which methodologies of the Web applications studied: *Non-Functional Requirements are considered, but there is no consensus in their meaning. There are neither specific techniques for elicitation of Non-Functional Requirements nor uniform guidelines for the specification and validation. There are no techniques supporting the treatment of the Non-Functional Requirements along the lifetime cycle. There is a lack of consensus to establish within which phase of development lifetime cycle Non-Functional Requirements are identified. There are Non-Functional Requirements that are not identified during requirement capture and analysis phases.*

After this analysis and the give importance of the Non-Functional Requirements, a process for the elicitation and specification of the Web Non-Functional Requirements (quality and restrictions) is proposed, exposed spreadsheets are described to

support processes and finally, validations of the concepts and techniques are introduced applied to a real project.

*Keywords:* Requirements Engineering, Non-Functional Requirements, Web Application, Quality, Restrictions, Elicitation Process and Web Non-Functional Requirements Specification, Non-Functional Requirements Matrix.

## Índice General

RESUMEN.....	I
ABSTRACT .....	III
CAPÍTULO: 1.....	1
1. INTRODUCCIÓN .....	1
1.1 MOTIVACIÓN .....	1
1.1 CONTRIBUCIÓN .....	2
1.2 ESTRUCTURA DEL DOCUMENTO .....	3
CAPÍTULO: 2.....	6
2. REQUERIMIENTOS NO FUNCIONES Y LAS APLICACIONES WEB. ....	6
2.1 REQUERIMIENTOS NO FUNCIONALES ¿QUÉ SON? .....	6
2.2 CONCEPTOS A APLICAR EN ESTE TRABAJO .....	19
CAPÍTULO: 3.....	24
3. TÉCNICAS UTILIZADAS EN LA INGENIERÍA DE REQUERIMIENTOS .....	24
CAPÍTULO: 4.....	32
4. METODOLOGÍAS DE DESARROLLO DE APLICACIONES WEB Y LOS REQUERIMIENTOS. ....	32
4.1 APLICACIONES WEB .....	32
4.2 METODOLOGÍAS DE DESARROLLO WEB Y LOS RNF .....	33
4.2.1 WSDM: WEB SITE DESIGN METHOD .....	37
4.2.2 HFPM: HYPERMEDIA FLEXIBLE PROCESS MODELING .....	41
4.2.3 UWE: UML-BASED WEB ENGINEERING.....	46
4.2.4 WEBML: WEB MODELING LANGUAGE.....	50
4.2.5 NTD: NAVIGATIONAL DEVELOPMENT TECHNIQUES .....	54
4.2.6 DDDP: DESIGN-DRIVEN REQUIREMENTS ELICITATION .....	62
CAPÍTULO: 5.....	66
5. TRATAMIENTO DE LOS RNF EN LAS METODOLOGÍAS DE DESARROLLO WEB. ....	66
5.1 TIPOS DE REQUERIMIENTOS NO FUNCIONALES EN CADA METODOLOGÍA.....	66

5.2	TÉCNICAS DE INGENIERÍA DE REQUERIMIENTOS PARA EL TRATAMIENTO DE RNF.....	77
5.3	CICLO DE VIDA Y LOS REQUERIMIENTOS NO FUNCIONALES.....	79
5.4	CONCLUSIONES DEL ANÁLISIS COMPARATIVO .....	83
CAPÍTULO: 6.....		85
6.	PROCESOS DE ELICITACIÓN Y ESPECIFICACIÓN DE RNF PARA WEB.....	85
6.1	PROCESO DE ELICITACIÓN DE RNF WEB .....	88
6.1.1	OBJETIVO DE LA ELICITACIÓN DE RNF WEB .....	89
6.1.2	ENTRADAS, .....	90
6.1.3	ACTIVIDADES .....	90
6.1.4	TÉCNICAS/HERRAMIENTAS .....	94
6.1.5	LOS PRODUCTOS/SALIDAS .....	95
6.1.6	INTERACCIÓN.....	95
6.2	PROCESO DE ESPECIFICACIÓN DE RNF WEB .....	96
6.2.1	OBJETIVO DE LA ESPECIFICACIÓN DE RNF WEB .....	98
6.2.2	ENTRADAS .....	98
6.2.3	ACTIVIDADES .....	98
6.2.4	TÉCNICAS/ HERRAMIENTAS .....	101
6.2.5	LOS PRODUCTOS/SALIDAS .....	102
6.2.6	INTERACCIÓN.....	102
CAPÍTULO: 7.....		104
7.	CONCEPTOS BÁSICOS PARA LA DEFINICIÓN DE LAS PLANTILLAS.....	104
7.1	INTRODUCCIÓN .....	104
7.2	CLASES DE STAKEHOLDERS .....	104
7.3	ESTÁNDAR DE CALIDAD INTERNACIONAL ISO/IEC 9126-1.....	107
7.4	CARACTERÍSTICAS CONSIDERADAS EN LA PLANTILLA DE ELICITACIÓN DE RNF DE CALIDAD.....	111
7.5	RESTRICCIONES RELEVANTES PARA APLICACIONES WEB .....	123
7.6	CARACTERÍSTICAS DE UN BUEN REQUERIMIENTO .....	132

<b>CAPÍTULO: 8</b> .....	<b>135</b>
<b>8. PLANTILLAS PARA LA ELICITACIÓN DE RNF WEB</b> .....	<b>135</b>
8.1 INTRODUCCIÓN .....	135
8.2 ESTRUCTURA DE LA PLANTILLA DE ELICITACIÓN DE RNF DE CALIDAD .....	135
8.3 DESCRIPCIÓN DE LA PLANTILLA DE ELICITACIÓN DE RNF DE CALIDAD .....	137
8.3.1 ENCABEZADO DE LA PLANTILLA DE ELICITACIÓN DE RNF DE CALIDAD .....	139
8.3.2 CUERPO DE LA PLANTILLA DE ELICITACIÓN DE RNF DE CALIDAD .....	140
8.4 ESTRUCTURA DE LA PLANTILLA PARA ELICITACIÓN DE RNF RESTRICCIONES.....	153
8.5 DESCRIPCIÓN DE LA PLANTILLA DE ELICITACIÓN DE RNF RESTRICCIONES .....	154
8.5.1 ENCABEZADO DE LA PLANTILLA DE ELICITACIÓN DE RNF-RESTRICCIONES.....	156
8.5.2 CUERPO DE LA PLANTILLA DE ELICITACIÓN DE RNF -RESTRICCIONES .....	156
<b>CAPÍTULO: 9</b> .....	<b>168</b>
<b>9. PLANTILLAS PARA ESPECIFICACIÓN DE RNF WEB</b> .....	<b>168</b>
9.1 INTRODUCCIÓN .....	168
9.2 PATRÓN PARA ESPECIFICACIÓN DE RNF. ....	168
9.3 ESTRUCTURA DE LA PLANTILLA DE ESPECIFICACIÓN DE RNF WEB.....	172
9.4 DESCRIPCIÓN DE LA PLANTILLA DE ESPECIFICACIÓN DE RNF WEB .....	174
9.4.1 ENCABEZADO DE LA PLANTILLA DE ESPECIFICACIÓN DE RNF WEB .....	176
9.4.2 CUERPO DE LA PLANTILLA DE ESPECIFICACIÓN DE RNF WEB.....	176
9.5 ESTRUCTURA DE LA MATRIZ DE RNF.....	179
9.6 DESCRIPCIÓN DE LA MATRIZ DE RNF .....	180
9.6.1 ENCABEZADO DE LA MATRIZ DE RNF .....	181
9.6.2 CUERPO DE LA MATRIZ DE RNF .....	181
9.6.3 TABLA DE TOTALES DE RNF .....	184
9.6.4 GRÁFICO DE ESTADO DE RNF .....	184
<b>CAPÍTULO: 10</b> .....	<b>185</b>
<b>10. CASO DE ESTUDIO</b> .....	<b>185</b>



---

10.1	PRESENTACIÓN DE DOMINIO .....	185
10.2	APLICACIÓN DE LA IR WEB PARA EL TRATAMIENTO DE LOS RNF .....	186
10.2.1	APLICACIÓN DEL PROCESO DE ELICITACIÓN DE RNF.....	188
10.2.2	APLICACIÓN DEL PROCESO DE ESPECIFICACIÓN DE RNF.....	199
10.3	RESULTADOS .....	206
CAPÍTULO: 11 .....		211
11.	CONCLUSIONES Y TRABAJOS FUTUROS.....	211
12.	BIBLIOGRAFÍA.....	217
13.	ANEXOS .....	221
13.1	PLANTILLA DE ELICITACIÓN DE RNF DE CALIDAD .....	221
13.2	PLANTILLA DE ELICITACIÓN DE RNF DE RESTRICCIONES .....	221
13.3	PLANTILLA DE ESPECIFICACIÓN DE RNF WEB .....	222
13.4	MATRIZ DE RNF .....	222
13.5	APLICACIÓN DEL PROCESOS DE ELICITACIÓN Y ESPECIFICACIÓN WEB .....	223
13.5.1	STAKEHOLDERS CLAVES DE LA APLICACIÓN WEB DEL CASO DE ESTUDIO.....	223
13.5.2	PLANTILLA DE ELICITACIÓN DE RNF DE CALIDAD DEL CASO DE ESTUDIO.....	225
13.5.3	PLANTILLA DE ELICITACIÓN DE RNF RESTRICCIONES DEL CASO DE ESTUDIO .....	225
13.5.4	MATRIZ DE RNF 1.0 - VERSIÓN INICIAL DEL CASO DE ESTUDIO.....	225
13.5.5	MATRIZ DE CONTROL DE AUTORIZACIÓN DE ACCESO DEL CASO DE ESTUDIO.....	225
13.5.6	PROTOTIPOS PARA SMARTPHONE Y PC DEL CASO DE ESTUDIO .....	226
13.5.7	PROTOTIPO DE GRÁFICO DE ACTIVIDAD DE PÓRTICOS DEL CASO DE ESTUDIO .....	227
13.5.8	MATRIZ DE RNF 1.1 - VERSIÓN INTERMEDIA DEL CASO DE ESTUDIO.....	228
13.5.9	MATRIZ DE RNF 1.2 – VERSIÓN DEFINITIVA DEL CASO DE ESTUDIO .....	228
13.5.10	PLANTILLA DE ESPECIFICACIÓN DE RNF WEB.....	228

## Índice de Figuras

FIGURA 2.1: CLASIFICACIÓN DE REQUERIMIENTOS NO FUNCIONALES – MAMANI .....	11
FIGURA 2.2: CLASIFICACIÓN DE REQUERIMIENTOS NO FUNCIONALES – SOMMERVILLE. ....	12
FIGURA 2.3: ÁRBOL DE CARACTERÍSTICAS DE LA CALIDAD DEL SOFTWARE – BOEHM .....	12
FIGURA 2.4: UNA TAXONOMÍA DE REQUERIMIENTOS NO FUNCIONALES- CYSNEIROS.....	13
FIGURA 4.1: METODOLOGÍAS PARA APLICACIONES WEB Y LOS REQUERIMIENTOS.....	36
FIGURA 4.2: DIAGRAMA DE LAS FASES DE WSDM .....	38
FIGURA 4.3: DIAGRAMA DE LAS FASES DE HPPM .....	42
FIGURA 4.4: DIAGRAMA DE PROCESO DE UWE .....	47
FIGURA 4.5: DIAGRAMA DE PROCESO DE WEBML.....	51
FIGURA 4.6: DIAGRAMA DE PROCESO DE NDT .....	56
FIGURA 4.7: DIAGRAMA DE PROCESO DE DDDP.....	63
FIGURA 5.1: REQUERIMIENTOS NO FUNCIONALES Y METODOLOGÍAS DE DESARROLLO WEB - PARCIAL.....	74
FIGURA 5.2: REQUERIMIENTOS NO FUNCIONALES Y METODOLOGÍAS DE DESARROLLO WEB.....	76
FIGURA 6.1: FRAMEWORK PARA EL PROCESO DE INGENIERÍA DE REQUERIMIENTOS. ....	86
FIGURA 6.2: PROCESO DE ELICITACIÓN Y ESPECIFICACIÓN DE RNF WEB. ....	87
FIGURA 6.3: PROCESO DE ELICITACIÓN DE RNF WEB.....	89
FIGURA 6.4: PROCESO DE ESPECIFICACIÓN DE RNF WEB.....	97
FIGURA 7.1: CARACTERÍSTICAS Y SUBCARACTERÍSTICAS DEL ESTÁNDAR ISO/IEC 9126-1. ....	109
FIGURA 7.1-CONTINUACIÓN:CARACTERÍSTICAS Y SUBCARACTERÍSTICAS DEL ESTÁNDAR ISO/IEC 9126-1 ....	110
FIGURA 7.2: CARACTERÍSTICAS DE ISO/IEC 9126-1 .....	111, 112
FIGURA 7.5: MATRIZ DE CONTROL DE AUTORIZACIÓN DE ACCESOS .....	145
FIGURA 8.1: ESTRUCTURA DE LA PLANTILLA DE ELICITACIÓN DE RNF DE CALIDAD.....	136
FIGURA 8.2: PLANTILLA PARA ELICITACIÓN DE RNF DE CALIDAD – APLICACIONES WEB.- VISTA PARCIAL .....	138
FIGURA 8.4: ESTRUCTURA DE LA PLANTILLA DE ELICITACIÓN DE RNF RESTRICCIONES .....	153
FIGURA 8.5: PLANTILLA PARA ELICITACIÓN DE RNF DE RESTRICCIONES - VISTA PARCIAL.....	155
FIGURA 9.1: PATRÓN PARA ESPECIFICACIÓN DE RNF .....	169

<b>FIGURA 9.2: ESTRUCTURA DE LA PLANTILLA DE ESPECIFICACIÓN DE RNF WEB .....</b>	<b>173</b>
<b>FIGURA 9.3: PLANTILLA DE ESPECIFICACIÓN DE RNF WEB .....</b>	<b>175</b>
<b>FIGURA 9.4: MATRIZ DE RNF .....</b>	<b>179</b>
<b>FIGURA 9.5: MATRIZ DE RNF – GRÁFICO DE ESTADO DE REQUERIMIENTOS.....</b>	<b>180</b>
<b>FIGURA 10.1: ESQUEMA DE REUNIONES DE RELEVAMIENTO DE LA APLICACIÓN WEB .....</b>	<b>187</b>
<b>FIGURA 10.2: LISTADO DE REQUERIMIENTOS FUNCIONALES DE LA APLICACIÓN WEB .....</b>	<b>189</b>
<b>FIGURA 10.3: DOMINIO Y OBJETIVOS DE LA APLICACIÓN WEB .....</b>	<b>190</b>
<b>FIGURA 10.4: RF Y SU RELACIÓN CON TIPOS DE RNF .....</b>	<b>191</b>
<b>FIGURA 10.5: ROL DE LOS STAKEHOLDERS CLAVES EN LAS PLANTILLAS DE ELICITACIÓN DE RNF .....</b>	<b>191</b>
<b>FIGURA 10.6: ELICITACIÓN DE RNF DE CALIDAD - CARACTERÍSTICAS SELECCIONADAS .....</b>	<b>192</b>
<b>FIGURA 10.7: ELICITACIÓN DE RNF - RESTRICCIONES.....</b>	<b>193</b>
<b>FIGURA 10.8: RNF CANDIDATOS (MATRIZ DE RNF SALIDA DE ELICITACIÓN).....</b>	<b>197</b>
<b>FIGURA 10.9: RELACIONES ENTRE RNF CANDIDATOS .....</b>	<b>199</b>
<b>FIGURA 10.10: RNF CANDIDATOS - POSPUESTOS .....</b>	<b>202</b>
<b>FIGURA 10.11: RNF CANDIDATO - RECHAZADO .....</b>	<b>203</b>
<b>FIGURA 10.12: GRÁFICO DE ESTADO DE RNF – INICIO ESPECIFICACIÓN.....</b>	<b>204</b>
<b>FIGURA 10.13: GRÁFICO DE ESTADO DE RNF – FIN DE ESPECIFICACIÓN .....</b>	<b>205</b>
<b>FIGURA 10.14: CLASIFICACIÓN DE RNF SEGÚN SU ESTADO – FIN DE ESPECIFICACIÓN .....</b>	<b>206</b>
<b>FIGURA 10.15: VISTA DE LA PLANTILLA DE ESPECIFICACIÓN DE RNF WEB .....</b>	<b>207</b>
<b>FIGURA 13.1: USUARIOS POTENCIALES DE LA APLICACIÓN WEB DEL CASO DE ESTUDIO .....</b>	<b>223</b>
<b>FIGURA 13.2: OTROS STAKEHOLDERS DE LA APLICACIÓN WEB DEL CASO DE ESTUDIO .....</b>	<b>224</b>
<b>FIGURA 13.3: MATRIZ DE CONTROL DE AUTORIZACIÓN DE ACCESOS DEL CASO DE ESTUDIO.....</b>	<b>226</b>
<b>FIGURA 13.4: PROTOTIPOS PARA DISPOSITIVOS CELULARES DEL CASO DE ESTUDIO. ....</b>	<b>226</b>
<b>FIGURA 13.5: PROTOTIPOS PARA PC DEL CASO DE ESTUDIO. ....</b>	<b>227</b>
<b>FIGURA 13.6: PROTOTIPO PARA GRÁFICO DE ACTIVIDAD DE PÓRTICOS DEL CASO DE ESTUDIO. ....</b>	<b>228</b>

## Índice de Tablas

TABLA 2.1: EJEMPLOS DE DISCREPANCIAS EN ESQUEMAS DE CLASIFICACIÓN DE RNF .....	18
TABLA 2.2: REQUERIMIENTOS NO FUNCIONALES COMO REQUERIMIENTOS DE CALIDAD .....	22
TABLA 4.1: REQUERIMIENTOS TRATADOS POR CADA METODOLOGÍA ... ..	35
TABLA 5.1: REQUERIMIENTOS NO FUNCIONALES DE CALIDAD Y LAS METODOLOGÍAS WEB.. ..	67
TABLA 5.2: REQUERIMIENTOS NO FUNCIONALES – RESTRICCIONES Y LAS METODOLOGÍAS WEB. ....	69
TABLA 5.3: REQUERIMIENTOS IDENTIFICADOS POR CADA METODOLOGÍA .....	72
TABLA 5.4: RNF IDENTIFICADOS Y TRATADOS POR CADA METODOLOGÍA DE DESARROLLO WEB - PARCIAL ...	73
TABLA 5.5: RNF IDENTIFICADOS Y TRATADOS POR CADA METODOLOGÍA DE DESARROLLO WEB .....	75
TABLA 5.6: TÉCNICAS DE INGENIERÍA DE REQUERIMIENTOS Y HERRAMIENTAS CASE APLICADAS A RNF .....	79
TABLA 5.7: LOS REQUERIMIENTOS NO FUNCIONALES Y EL CICLO DE VIDA .. ..	80
TABLA 5.8: INFLUENCIA DE LAS RESTRICCIONES DE PROCESO EN EL CICLO DE VIDA.....	82
TABLA 6.1: PROCESO DE ELICITACIÓN DE RNF WEB.....	95-96
TABLA 6.2: PROCESO DE ESPECIFICACIÓN DE RNF WEB.....	103

## Capítulo: 1

---

### 1. Introducción

#### 1.1 Motivación

Un requerimiento es definido como una condición o capacidad que debe satisfacer o poseer un sistema o un componente de un sistema para satisfacer un contrato, un estándar, una especificación u otro documento formalmente impuesto (IEEE Standard 610.12-1990).

Las tareas de elicitación, especificación y validación de requerimientos no son sencillas, la complejidad radica en parte en cómo identificar las funcionalidades que el sistema deberá implementar para satisfacer las necesidades expresadas y esperadas de los usuarios. Otra fuente de dificultades en el desarrollo de los requerimientos la constituyen los Requerimientos No Funcionales, esto es, las restricciones que debe satisfacer el sistema en construcción.

Los requerimientos de los sistemas software en general y los de las aplicaciones Web en particular no son un todo homogéneo; se pueden clasificar en dos grandes tipos, requerimientos funcionales y no funcionales.

Si bien el estado de arte de desarrollo de aplicaciones Web en la actualidad dista en aprovechar plenamente los avances en la Ingeniería del Software en los años recientes, las comunidades de Ingeniería Web y de Ingeniería de Requerimientos han dado pasos en el camino de reconocer la importancia de entender, antes de comenzar el proceso de diseño, las necesidades que deberá satisfacer una Aplicación Web. En el año 2010 la comunidad científica realizó el “1st Workshop on The Web and Requirements Engineering (WeRE)” en el contexto de la mayor conferencia de Ingeniería de Requerimientos, esto denota fuerza y mayor relevancia por considerar la contribución de los requerimientos al desarrollo de aplicaciones Web; eventos similares se llevaron a cabo en 2011 y 2012.

El crecimiento en el uso aplicaciones web de los últimos años fue acompañado con el surgimiento de metodologías para el desarrollo de aplicaciones web. Para Escalona y Koch [1], la mayoría de estos enfoques están orientados a las actividades de diseño dentro de

su ciclo de vida y contemplan en menor escala o no incluyen, las tareas de ingeniería de requerimientos, testeo y administración de la calidad. El trabajo de Escalona y Koch [1], aporta una comparación de metodologías de desarrollo web, destacando entre otros aspectos, técnicas de ingeniería de requerimientos y tipos de requerimientos que manejan en sus diferentes etapas. Respecto a los Requerimientos No Funcionales menciona cuales metodologías detectan los mismos, de hecho solo seis de las diez revisadas corresponden con ello y los agrupa a fin de resolver las diferencias entre las metodologías sobre estos requerimientos.

En la actualidad la Ingeniería de Requerimientos (IR) proporciona numerosas técnicas y herramientas sin embargo estas no son aplicadas habitualmente en el desarrollo de aplicaciones Web. Los trabajos realizados en [37, 40] dejan expuesto el importante retraso que sufre el tratamiento de los Requerimientos No Funcionales (RNF). En el campo de la Ingeniería de Requerimientos no se ofrece homogeneidad conceptual sobre los mismos y por su parte, las metodologías disponibles para el desarrollo de aplicaciones Web no son completas y registran imprecisiones o inconsistencias en el tratamiento de los RNF a lo largo del ciclo de vida.

La madurez del proceso de IR para aplicaciones Web parece ser insuficiente y demanda que surjan nuevos enfoques o se evolucionen de los enfoques existentes para el tratamiento de los RNF en particular.

Por su importancia, las actividades para captura, especificación y validación de los Requerimientos No Funcionales, merecen contar con técnicas y lineamientos específicos para el desarrollo de aplicaciones Web.

## **1.1 Contribución**

El objetivo de la presente tesis es desarrollar técnicas y lineamientos para los procesos de Elicitación y Especificación de la Ingeniería de Requerimientos, que sean aplicables al tratamiento de los requerimientos no funcionales en un marco de desarrollo de aplicaciones Web.

Esta tesis posee dos aportes significativos. En primer lugar provee procesos para la elicitación y la especificación de los requerimientos no funcionales de aplicaciones Web y un patrón para escribir requerimientos no funcionales. En segundo lugar provee todas las plantillas definidas como soporte de los procesos de elicitación y especificación construidas en formato Excel, con los lineamientos particulares para que puedan ser completadas por un Ingeniero de requerimientos, estas son:

- Plantillas para la elicitación de los RNF web de calidad y Matriz de Control de Autorización de Accesos.
- Plantillas para la elicitación de los RNF web - restricciones,
- Una plantilla de especificación de los RNF Web y
- Una matriz de RNF que da soporte y permite el seguimiento del estado de los requerimientos durante la ejecución de los dos procesos.

Como parte del proceso de elaboración de esta tesis se presentó un paper en ASSE 2012 - 13th Argentine Symposium on Software Engineering (La Plata, Argentina, Agosto 2012), “*Requerimientos No funcionales para aplicaciones Web*”, publicado en *Anales de las 41 JAIIO*, Buenos Aires, 2012, ISSN 1850-2776 [40].

## 1.2 Estructura del documento

El alcance del presente trabajo parte de una revisión del estado de arte de los conceptos de RNF en la literatura de la Ingeniería de Requerimientos, luego se estudian los enfoques completos de metodologías existentes para el desarrollo de aplicaciones Web y se ofrece un análisis comparativo de estos enfoques buscando dar respuestas a los siguientes interrogantes:

*¿Consideran las metodologías de desarrollo Web los RNF?*

*¿Qué técnicas proponen (si lo hacen) para manejar a los RNF?*

*¿Utilizan técnicas de la IR para el tratamiento de los RNF?,*

*¿Hay herramientas que soporten a las metodologías de desarrollo web?*

*¿Consideran estas herramientas la gestión de RNF?*

*¿Cómo es el tratamiento de los RNF en el ciclo de vida de desarrollo?*

A partir de los resultados del análisis comparativo se proponen procesos para la Elicitación y Especificación de los RNF Web (de calidad y restricciones), se describen las plantillas propuestas que dan soporte a los procesos y finalmente se presenta la validación de los conceptos y técnicas propuestos mediante un caso de estudio en un proyecto real.

El documento se encuentra organizado de la siguiente manera:

En el capítulo 2, para responder a la pregunta *¿Qué son los Requerimientos No Funcionales?*, se presenta una revisión y análisis de definiciones y esquemas de clasificación existentes en la literatura de IR. Al final de este capítulo se presentan los conceptos base sobre los tipos de RNF a ser considerados en este trabajo.

El capítulo 3, muestra una revisión de las técnicas de captura, especificación, y validación de requerimientos predominantes en la IR.

En el capítulo 4, se realiza un análisis de seis metodologías de desarrollo web tomadas a partir del estudio realizado por Escalona y Koch [1], señaladas por los autores como aquellas que identifican RNF. En este capítulo se analiza el ciclo de vida completo de cada enfoque a fin de establecer cuáles tipos de RNF son identificados en cada uno y que técnicas son empleadas para su elicitación, especificación, validación y gestión de requerimientos.

En el capítulo 5, se realiza una comparación de las metodologías estudiadas en el capítulo 4, a fin de analizar:

- a) Conceptos de RNF identificados en cada metodología.
- b) Técnicas de elicitación, validación y especificación que utilizan para el tratamiento de los RNF. Técnicas que proponen (si lo hacen) para la gestión de los Requerimientos No Funcionales.
- c) Los RNF identificados por cada metodología y su relación con las fases del ciclo de vida.

Al final de este capítulo se presentan las conclusiones que se desprenden de este análisis.

En el capítulo 6, se describen los procesos de Elicitación y Especificación propuestos para la aplicación de ingeniería de requerimientos al tratamiento de RNF en el desarrollo de aplicaciones Web.



El capítulo 7 presenta conceptos base y consideraciones particulares empleados en la definición de las plantillas que propone esta tesis.

Los capítulos 8 y 9, describen las plantillas y el Patrón para especificación de RNF que se proponen este trabajo para dar soporte a los procesos de Elicitación y Especificación de RNF de aplicaciones Web estas son:

- a) Plantilla de Elicitación de RNF de Calidad
- b) Plantilla de Elicitación de RNF de Restricciones
- c) Patrón para especificación de RNF
- d) Plantilla de especificación de RNF Web
- e) Matriz de RNF.

En el capítulo 10, se demuestra la aplicación de los procesos presentados y el uso de las plantillas de Elicitación y Especificación de RNF para aplicaciones Web, mediante un caso real.

El capítulo 11 presenta las conclusiones y se mencionan líneas de investigación o trabajos que se desprenden a partir de esta tesis.

## Capítulo: 2

---

### 2. Requerimientos No funciones y las aplicaciones Web.

#### 2.1 Requerimientos No Funcionales ¿Qué son?

La bibliografía existente presenta una gran diversidad de enunciados y conceptos para definir Requerimientos No Funcionales, si bien el término es utilizado desde hace años, en la actualidad no hay consenso en la comunidad de la IR para establecer “qué son” y por consiguiente tampoco hay unificación de criterios para establecer cómo se elicitán, como se especifican, ni como se validan.

La carencia de consenso se ve reflejada en la diversidad de términos utilizados en las definiciones, con significados poco claros, que dan lugar a la ambigüedad sobre su alcance o lo que representan los mismos; las discrepancias conceptuales también pueden encontrarse en las clasificaciones propuestas para Requerimientos No Funcionales, que incluyen árboles o categorías de conceptos que se muestran en forma separada en otras definiciones.

Para ilustrar lo expuesto se citarán ejemplos de definiciones y esquemas de clasificación extraídos de trabajos relacionados a esta problemática como [31] [32] y en fuentes adicionales:

- Glinz en [31] desarrolla una discusión sobre la problemática de la noción de Requerimientos No Funcionales, basado en el análisis de trece definiciones seleccionadas de la literatura y la web.
- Chung y Leite en [32] presentan una revisión del estado de arte del tratamiento de los Requerimientos No Funcionales, dentro de la literatura del área de requerimientos, mencionan diez definiciones de diferentes autores, siete de las cuales fueron citadas en el trabajo elaborado por Glinz [31]. Además analizan cuatro esquemas de clasificación.
- Cysneiros en su tesis de doctorado [4] menciona tres esquemas de clasificación de Requerimientos No Funcionales propuestas por diferentes autores, y propone una taxonomía para los mismos.

Las definiciones y esquemas de clasificación los Requerimientos No Funcionales son presentados a continuación indicando su autor y la fuente de la cual fueron extraídas:

- a) “Describen aspectos del comportamiento de un sistema, capturando las propiedades y restricciones bajo las cuales un sistema debe operar”. Anton en [31] [32].
- b) [Son] “atributos requeridos del sistema, incluyendo portabilidad, confiabilidad, eficiencia, la ingeniería humana, capacidad de prueba, compresión y capacidad de modificación”. Davis en [31] [32].
- c) [Son] “requerimientos que no se refieren específicamente a la funcionalidad de un sistema. Se imponen restricciones sobre el producto que se está desarrollando y el proceso de desarrollo, y que especifican restricciones externas que el producto debe cumplir”. Kotonya y Sommerville en [31] [32].
- d) “... Requerimientos globales en el desarrollo o los costos operativos, performance confiabilidad, mantenibilidad, portabilidad, robustez, etc. (...) No hay una definición formal o una lista completa de Requerimientos No Funcionales”. Mylopoulos, Chung y Nixon en [31] [32].
- e) [Son las] “propiedades de comportamiento que las funciones especificadas deben tener, tales como performance, usabilidad”. Ncube en [31] [32].
- f) [Es una] “propiedad, o cualidad, que el producto debe tener, como una apariencia, o una propiedad de la velocidad o precisión”. Robertson and Robertson en [31] [32].
- g) [Es una] “descripción de una propiedad o característica que un sistema de software debe mostrar o una restricción que se debe respetar, que no sea el comportamiento observable del sistema”. Wiegers en [31] [32].
- h) “... Tipos de problemas: problemas funcionales asociados con los servicios a ser proporcionados, y los problemas no funcionales relacionados con la calidad del servicio - como seguridad, precisión, performance, etc”. Van Lamsweerde en [32].
- i) “El término "Requerimiento no-funcional" es utilizado para delinear requerimientos enfocándose en "lo bien" que el software hace algo en contraposición a los requerimientos funcionales, que se centran en el "qué" hace el software”. Paech, Kerkow en [32].

- j) “Dicho de otro forma, NFR constituyen las justificaciones de las decisiones de diseño y restringen la forma en la cual la funcionalidad requerida puede ser realizada”. Landes, Studer en [32].
- k) Los estándares de IEEE no definen este término. El estándar IEEE 610.12 distingue requerimientos de diseño, requerimientos de implementación, requerimientos de interfaz, requerimientos de performance y los requerimientos físicos. [31].
- l) Este término tampoco está definido en el estándar IEEE 830-1998. El estándar define las categorías de funcionalidad, interfaces externas, performance, atributos (portabilidad, seguridad, etc), y las limitaciones de diseño. Los requerimientos del proyecto (por ejemplo, calendario/tiempo, costo, o los requerimientos de desarrollo) están explícitamente excluidos. [31].
- m) [Un RNF es] “un NFR es un requerimiento que especifica propiedades del sistema, tales como restricciones de implementación y de ambiente, performance, dependencias de plataforma, mantenimiento, capacidad de extensión y confiabilidad. Un requerimiento que especifica restricciones físicas sobre un requerimiento funcional”. Jacobson, Booch y Rumbaugh en [31].
- n) [Un NFR es] “un requerimiento de un servicio que no tiene que ver con la funcionalidad, sino que describe atributos, restricciones, consideraciones de performance, diseño, calidad del servicio, consideraciones de ambiente, fallas y recuperación”. SCREEN Glossary en [31].
- o) [Requerimientos No Funcionales: Requerimientos que especifican los criterios que se pueden utilizar para juzgar el funcionamiento de un sistema, en lugar de comportamientos específicos]. Wikipedia en [31].

Hay dentro de la literatura existente, dos definiciones que nos interesa sumar a las descritas. Cysneriros en [7] se refiere a los Requerimientos No Funcionales como:

- p) ... [los NFR] “son requerimientos de calidad, que representan restricciones o las cualidades que el sistema debe tener tales como: Precisión, usabilidad, seguridad, rendimiento, confiabilidad, performance entre otras”. Cysneiros en [7].
- q) ... [los NFR] “definen restricciones globales sobre el sistema, subsistema, sobre un requerimiento funcional, el proceso de desarrollo o sobre el proceso de despliegue...

lo cual implica que estos surgen de todas partes del sistema y sus interacciones”. Cysneiros en [7].

Del análisis de las definiciones identificamos que en la mayoría de estas, se utilizan los términos *atributo*, *restricción*, *característica*, *propiedad*, *calidad*, para decir “que son” Requerimientos No Funcionales. Algunas definiciones no mencionan ejemplos que aporten claridad o precisión sobre su significado, por lo que estos pueden ser interpretados como un concepto diferente en cada definición. No hay consenso sobre la terminología a usar, tampoco sobre los conceptos que cada término describe, si los analizamos en forma aislada encontramos que:

*Atributo*. Para Davis en b) cada requerimiento no funcional es un *atributo* del sistema; en IEEE 830-1998 en l) los *atributos* son una categoría que se subdivide en una colección de cualidades específicas y en SCREEN glossary n) el término hace referencia a *atributos* del servicio.

Solo la IEEE 830-1998 en l) detalla ejemplos de *atributos* (portabilidad, seguridad etc.), Davis en b) y SCREEN Glossary en n) usan este término en forma general.

*Restricción*. Para Kotonya y Sommerville en c) los requerimientos son *restricciones*; en otras definiciones como Anton en a), IEEE 830-1998 en l), Jacobson, Booch and Rumbaugh en m), Wiegers en g) y SCREEN glossary n), Cysneiros en q) se refieren al término como una limitación o condición.

Kotonya y Sommerville en s) ofrecen una clasificación y subclasificación de *restricciones* (propone Requerimientos de producto, de organización y externos).

Las otras definiciones si bien delimitan el término, no mencionan ejemplos específicos, la IEEE 830-1998 en l) restringe el término a *restricciones* de diseño; Jacobson, Booch y Rumbaugh en m) a *restricciones* físicas; Cysneiros en q) a restricciones globales sobre el sistema, subsistema, sobre un requerimiento funcional, el proceso de desarrollo o sobre el proceso de despliegue.

Anton en a), Wiegers en g) y SCREEN glossary n), usan el término de forma general.

Los términos *característica* y *propiedad* tienen un significado similar, hacen referencia a algo que el sistema debe tener. Las definiciones citadas, no son claras respecto si

consideran las *restricciones* como *propiedades*, ambos términos pueden formar parte de una misma definición como en Antón a) y Wiegers g).

Ncube en e) detalla ejemplos de *propiedades* (la performance, la usabilidad), para Robertson and Robertson en f) una *propiedad* puede ser la apariencia, velocidad o precisión; Anton en a) y Wiegers en g) se refieren al término *propiedad* como algo general y no especifican ejemplos.

*Calidad*. Tanto Robertson y Robertson en f), Van Lamsweerde en h), como Cysneiros en p) cuando mencionan este término, hacen referencia a un conjunto de cualidades específicas que el producto o sistema debe tener.

Robertson y Robertson en f) menciona como ejemplos la velocidad o precisión; Van Lamsweerde en h) la seguridad, precisión, performance y Cysneiros en p) a la precisión, usabilidad, seguridad, rendimiento, confiabilidad, performance.

SCREEN glossary n) usa el término de forma general y Davis en b) si bien se refiere a los Requerimientos No Funcionales como *atributos*, se basa en el árbol de *calidad* presentado por Boehm en t) para enumerarlos.

Algunas de las definiciones citadas, no utilizan los términos descritos, sino que usan otras palabras para definir a los Requerimientos No Funcionales:

Landes, Studer en j) se refieren a estos como... “*las justificaciones de las decisiones de diseño*”...

Paech, Kerkow en i) los describen como... “*el término que es utilizado para delinear requerimientos enfocándose en “lo bien” que el software hace algo*”.

Wikipedia en o) los define como... “*los criterios que se pueden utilizar para juzgar el funcionamiento de un sistema*”.

En mi opinión, dentro del conjunto de definiciones citadas i) y o) son las más imprecisas y ambiguas. La variedad y discrepancia de terminología y conceptos utilizados en las definiciones de Requerimientos No Funcionales, deja expuesta la necesidad de una definición clara y concisa.

---

### ***Clasificación de Requerimientos No Funcionales:***

Las figuras 2.1, 2.2, 2.3 y 2.4 ilustran las clasificaciones de Requerimientos No Funcionales propuestas por Mamani, Sommerville, Boehm y Cysneiros.

r) Propuesta por Mamani en [4]:

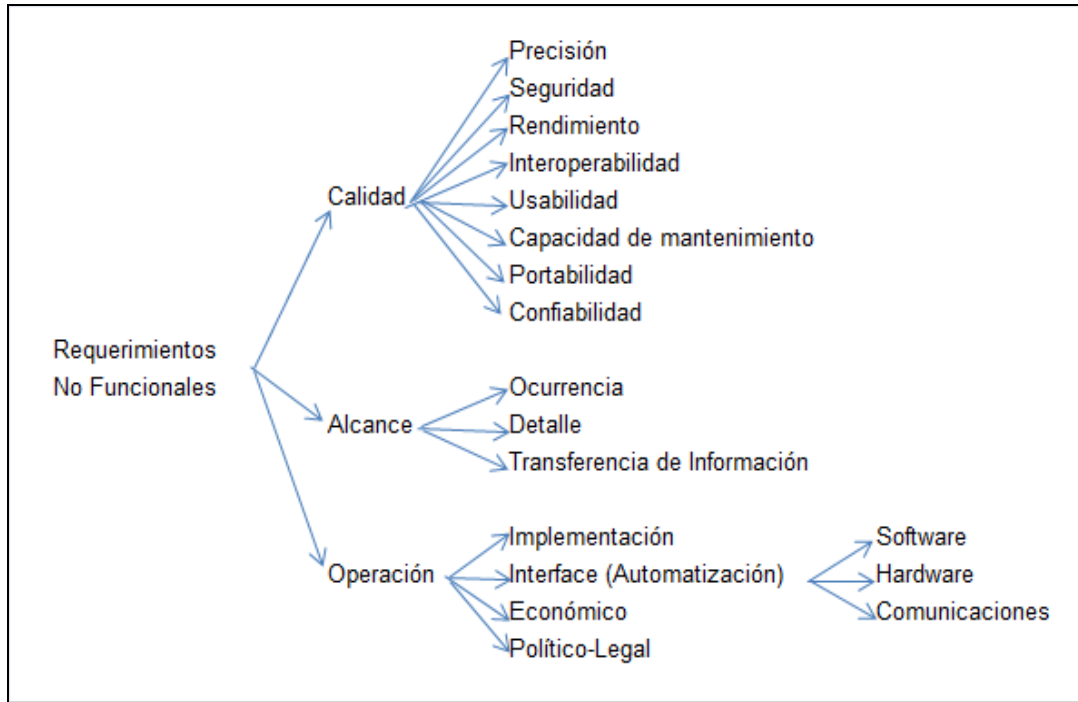


FIGURA 2.1: Clasificación de Requerimientos No Funcionales – Mamani.

s) Propuesta por Sommerville en [4]:

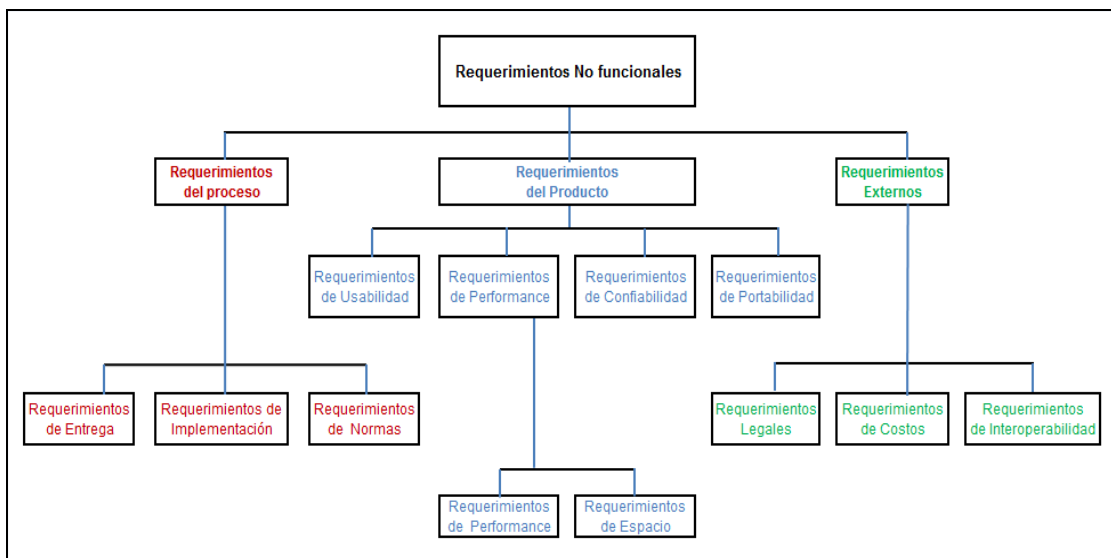


FIGURA 2.2: Clasificación de Requerimientos No Funcionales – Sommerville.

t) Árbol de características de la calidad del software propuesta por Boehm en [4], [32].

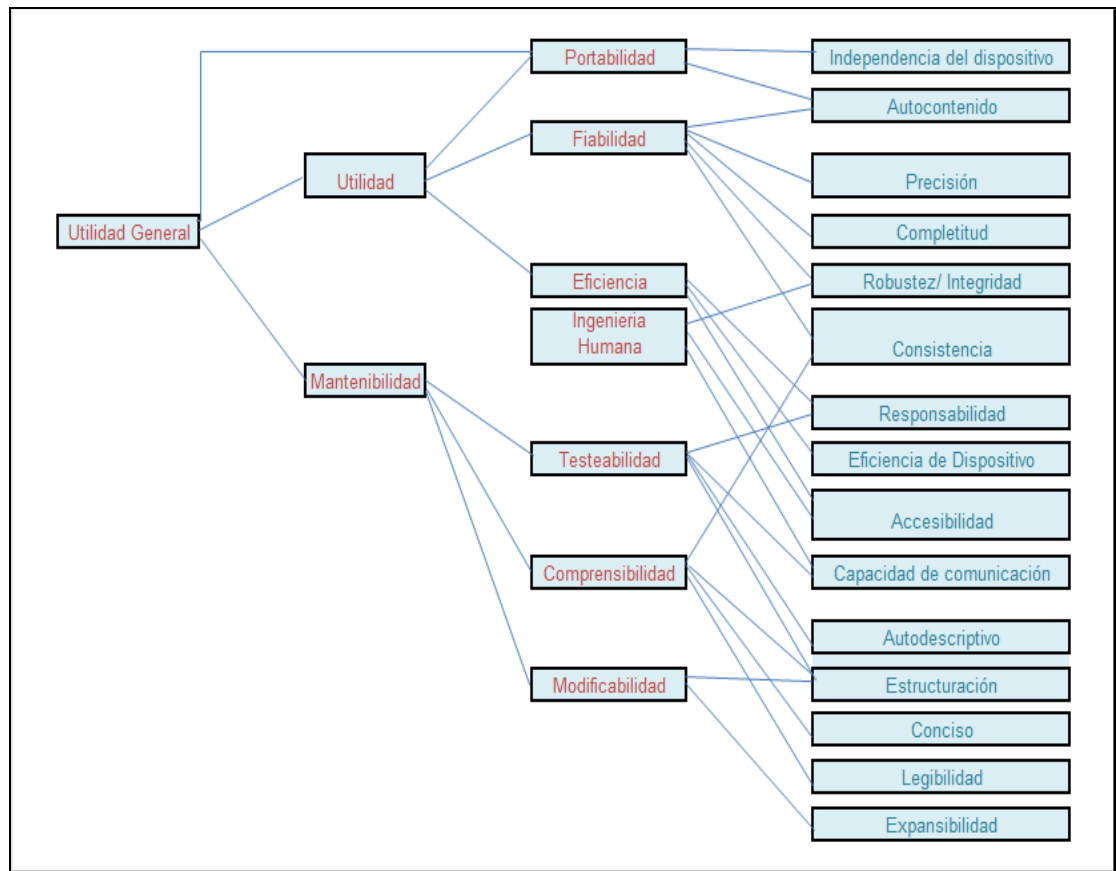


FIGURA 2.3: Árbol de Características de la Calidad del Software – Boehm

u) Taxonomía de Requerimientos No Funcionales propuesta por Cysneiros en [4]:



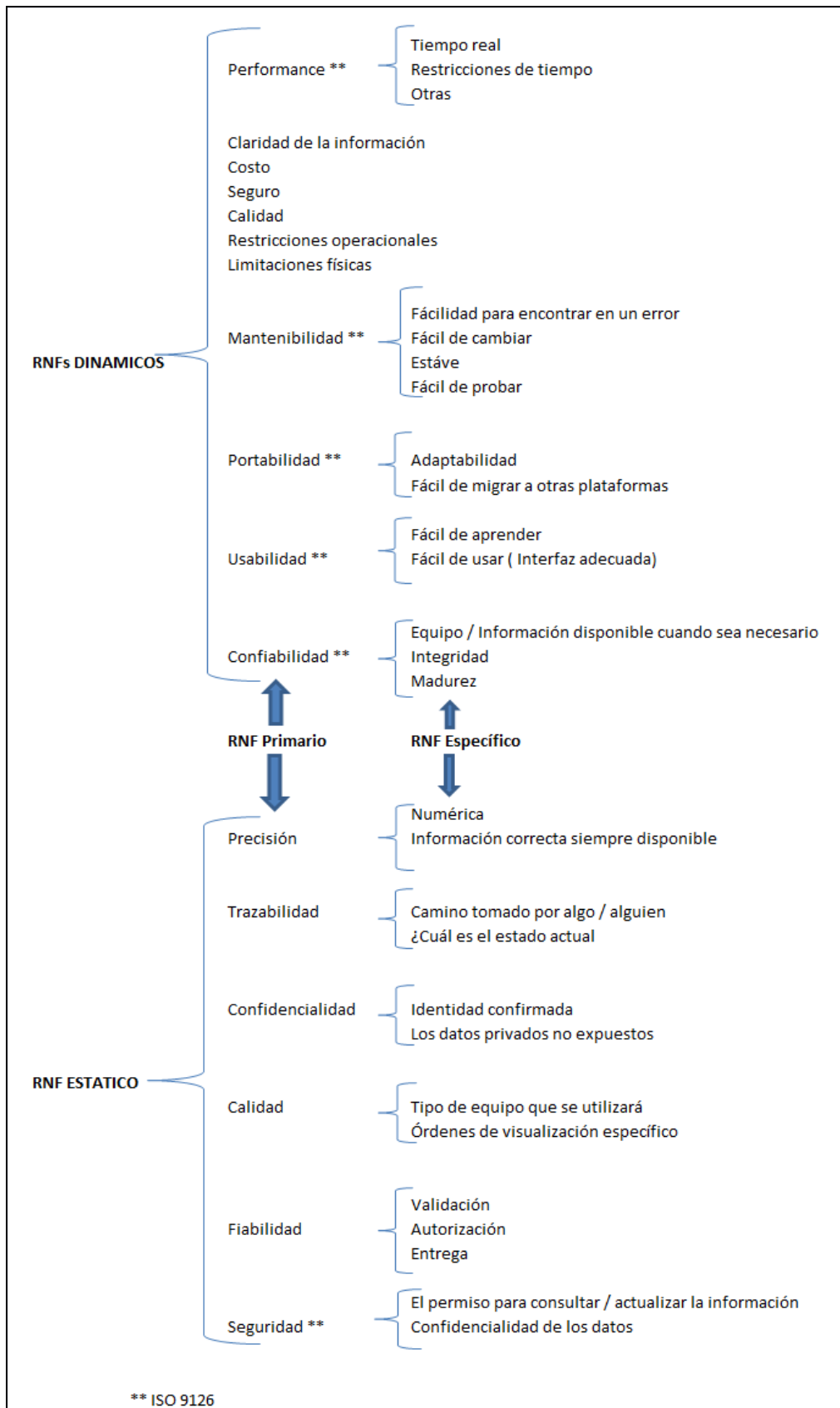


FIGURA 2.4: Una Taxonomía de Requerimientos No Funcionales- Cysneiros

v) Estándar de clasificación IEEE 810-1998 en [31]: clasifica a los Requerimientos No Funcionales en:

- *Requerimientos de Interfaz externa*: Describen los requerimientos que afecten a la:
  - Interfaz de usuario,
  - Interfaz con otros sistemas (hardware y software)
  - Interfaces de comunicaciones.
- *Requerimientos de Performance*: Detallaran los requerimientos relacionados con la carga esperada que debe soportar el sistema. (número de conexiones simultaneas, número de transacciones por segundo que deberá soportar el sistema, etc. También, si es necesario, se especificarán aquellos requerimientos que afecten a la información, por ejemplo, la frecuencia de uso, las capacidades de acceso y la cantidad de registros que se espera almacenar (decenas, cientos, miles o millones).
- *Restricciones de Diseño*: Todo aquello que restrinja las decisiones relacionadas al diseño de la aplicación: Restricciones de otros estándares, limitaciones del hardware, etc.
- *Atributos del Sistema*: Los atributos de calidad (las "ilities") del sistema: Fiabilidad, mantenibilidad, portabilidad, la seguridad deberá especificar que tipos de usuario están autorizados, o no, a realizar ciertas tareas y cómo se implementarán los mecanismos de seguridad (por ejemplo, por medio de un login y una password).

w) Estándar de calidad internacional ISO/IEC 9126-1 en [34], [32]: Distingue cuatro niveles de calidad. El modelo de calidad del primer nivel propone las seis características que se detallan a continuación, cada una de ellas involucra a su vez subcaracterísticas.

- *Funcionalidad*: Un conjunto de atributos que tienen que ver con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son las que satisfacen las necesidades explícitas o implícitas.

Subcaracterísticas: Adecuación, Precisión, Interoperabilidad, Cumplimiento de la funcionalidad y Seguridad (que reúne características orientadas a la seguridad de la información y los recursos del sistema.

- *Confiabilidad*: Un conjunto de atributos que influyen en la capacidad del software para mantener su nivel de rendimiento bajo las condiciones establecidas para un período determinado de tiempo.

Subcaracterísticas: Madurez, Tolerancia a fallas, Recuperabilidad, Cumplimiento de confiabilidad.

- *Usabilidad*: Un conjunto de atributos que influyen en el esfuerzo necesario para el uso del software por usuarios determinados.

Subcaracterísticas: Comprensibilidad, Facilidad de aprendizaje, Operatividad, Atractivo, Cumplimiento de usabilidad.

- *Eficiencia*: Un conjunto de atributos que influyen en la relación entre el nivel de rendimiento o performance del software y la cantidad de recursos utilizados, bajo condiciones indicadas.

Subcaracterísticas: Tiempo de Comportamiento, Utilización de recursos, Cumplimiento de eficiencia.

- *Mantenibilidad*: Un conjunto de atributos que tienen que ver con el esfuerzo necesario para realizar las modificaciones especificadas.

Subcaracterísticas: Analizabilidad, Variabilidad, Estabilidad, Capacidad de prueba, Cumplimiento de mantenimiento.

- *Portabilidad*: Un conjunto de atributos que influyen en la capacidad del software para ser transferido de un ambiente a otro.

Subcaracterísticas: Adaptabilidad, Facilidad de instalación, Co-Existencia, Intercambiabilidad, Cumplimiento de Portabilidad.

x) Esquema introducido por Roman fue analizado en [32], este describe la siguiente clasificación de Requerimientos No Funcionales:

- *Requerimientos de Interfaz*: Que describen como el sistema va a interactuar con el usuario, el ambiente y otros sistemas.
- *Requerimientos de Performance*: Que describen las limitaciones de rendimiento del sistema que involucran:
  - Límites de tiempo / espacio: como la carga de trabajo, tiempo de respuesta, rendimiento y el espacio de almacenamiento disponible.
  - Fiabilidad que implica la disponibilidad de los componentes y la integridad de la información mantenida y suministrada.
  - Seguridad, tales como los flujos de información permisibles.
  - Supervivencia o la resistencia al fuego, las catástrofes naturales.
- *Requerimientos de Operación*: Son las limitaciones físicas (tamaño, peso), la disponibilidad de personal, las consideraciones de nivel, la accesibilidad del sistema para el mantenimiento, etc.
- *Requerimientos del ciclo de vida*: Se pueden clasificar en dos subcategorías:
  - Calidad del diseño: se mide en términos tales como mantenibilidad, capacidad para ser mejorable, la portabilidad.
  - Límites en materia de desarrollo, tales como las limitaciones de tiempo de desarrollo, disponibilidad de recursos, normas metodológicas, etc.
- *Requerimientos económicos*: los costos inmediatos y/o a largo plazo.
- *Requerimientos políticos*.

y) **FURPS+**: desarrollado en Hewlett-Packard, fue elaborado por Grady y Caswell, citado en [32]: Representa un modelo de clasificación de atributos de calidad del software, o Requerimientos No Funcionales. Es un acrónimo de las diferentes categorías en Inglés (Functionality, Usability, Reliability, Performance, Soportability)

- *Functionality*: Conjunto de características, capacidades, la generalidad, Seguridad.
- *Usability*: Factores humanos, Estética, Coherencia, Documentación
- *Reliability*: Frecuencia/Severidad de la falla, la recuperabilidad, la previsibilidad, La precisión, tiempo medio entre fallas.
- *Performance*: Velocidad, Eficiencia, Consumo de recursos, Rendimiento, Tiempo de respuesta.
- *Supportability*: Capacidad de prueba, Extensibilidad, Adaptabilidad, Mantenimiento, Compatibilidad, Capacidad de configuración y Capacidad de servicio, facilidad de instalación, Localizable, Portabilidad.

El signo “+”: Son requerimientos adicionales que regularmente son restricciones.

Un análisis de los ocho esquemas de clasificación presentados permite ver que son inconsistentes unos con otros, en la cantidad de niveles que tiene cada estructura, y en los conceptos agrupados en cada categoría o característica. Por ejemplo, en algunos casos la *performance* es un atributo o subcaracterística, en otros casos es una categoría es si misma que contiene a un subconjunto de atributos o subcaracterísticas y hay esquemas en los cuales no es mencionado como tal. A continuación la tabla 2.1 muestra un resumen de observaciones similares para otros ejemplos como son: *usabilidad, seguridad, mantenimiento*.

Referencia	Autor de Esquema de Clasificación	Atributos de calidad			
		Performance	Usabilidad	Seguridad	Mantenimiento
r)	Mamani	-	Atributo (Depende de la Categoría Calidad)	Atributo (Depende de la Categoría Calidad)	Atributo (Depende de la Categoría Calidad)
s)	Kotonya y Sommerville	Clasificación (RNF de Producto)	Clasificación (RNF de Producto)	-	-
t)	Boehm	-	-	-	Categoría (Agrupa otras características)
u)	Cysneiros	RFN Primario (Tiene RNF Específicos)	RFN Primario (Tiene RNF Específicos)	RFN Primario (Tiene RNF Específicos)	RFN Primario (Tiene RNF Específicos)
v)	IEEE 810	Característica (Agrupa Subcaracterísticas)	Atributo (Depende de Requerimientos de Interfaz Externa)	Característica (Depende de Atributos de sistema)	Característica (Depende de Atributos de sistema)
w)	ISO/IEC 9126	Subcaracterística (Depende de una característica)	Característica de Calidad	Subcaracterística (Depende de una característica)	Característica de Calidad
x)	Roman	Característica (Agrupa Subcaracterísticas)	Atributo (Depende de Requerimientos de Interfaz)	Atributo (Depende de Requerimientos de Performance)	Atributo (Depende de Requerimientos de Ciclo de Vida)
y)	FURPS	Característica (Agrupa Subcaracterísticas)	Característica (Agrupa Subcaracterísticas)	Atributo (Depende de una Categoría)	Atributo (Depende de una Categoría)

TABLA 2.1: Ejemplos de discrepancias en esquemas de clasificación de RNF.

Otro punto importante a destacar es que algunos esquemas plantean una clasificación basados en el concepto de calidad y atributos de calidad, mientras que otros, consideran atributos de calidad y aportan otras nuevas formas de agrupación, por ejemplo, Roman en x) presenta las categorías: *Requerimientos del ciclo de vida*, *Requerimientos económicos* y *Requerimientos políticos*, Antoya y Sommerville en s) consideran *Requerimientos Legales* dentro de una agrupación de *Requerimientos externos*. Cada esquema de clasificación está ligado al concepto o definición de Requerimientos No Funcionales de su autor.

Quizás no hay que seguir buscando respuestas a la pregunta *¿Qué son los Requerimientos No Funcionales?* y generar definiciones nuevas. Quizás se debería poner foco en trabajar sobre la unificación de criterios de la terminología en las definiciones existentes y recién a partir de ahí, analizar las diferentes categorías de clasificación propuestas a fin de armar un esquema de referencia que agrupe los conceptos y tipos de Requerimientos No Funcionales.

## **2.2 Conceptos a aplicar en este trabajo**

De lo expuesto en la sección 2.1 surgen los siguientes interrogantes:

¿Cómo se aplican estos conceptos en enfoques metodológicos de desarrollo de aplicaciones web?

¿Cuál o cuáles tipos de Requerimientos No Funcionales son significativos en el desarrollo de aplicaciones web?

El crecimiento en el uso aplicaciones web de los últimos años, impulsó el surgimiento de metodologías para el desarrollo de aplicaciones web. Para Escalona y Koch [1], la mayoría de estos enfoques están orientados a actividades de diseño dentro de su ciclo de vida, contemplando en menor escala, o no incluyendo, tareas de ingeniería de requerimientos, testeo y administración de la calidad.

Tanto en los enfoques de desarrollo tradicional (no web) como en el desarrollo de aplicaciones web, los requerimientos juegan un rol relevante, las consecuencias de partir de requerimientos especificados en forma ambigua, vaga o incorrecta, pueden reflejarse en modelos de diseño y arquitecturas de software inadecuados, demoras en la planificación por la aparición de gran cantidad de fallas o defectos que deben ser corregidos, hasta la no aceptación de los usuarios.

Actualmente la Ingeniería de Requerimientos proporciona numerosas técnicas y herramientas para identificar, describir, validar y gestionar requerimientos, sin embargo, estos no son aplicados muy a menudo por los ingenieros de software. En particular en el desarrollo de aplicaciones web las que se construyen en forma ad-hoc, a pesar de que por la complejidad de las aplicaciones web de hoy se requiere un enfoque más sistemático, la

madurez del proceso de Ingeniería de Requerimientos parece ser insuficiente, específicamente con el tratamiento de los Requerimientos No Funcionales.

Durante el proceso de elicitación, los Requerimientos No Funcionales podrían aparecer junto a los requerimientos funcionales, por ejemplo en forma de:

- Políticas de negocio o externas al negocio (económicas, culturales, legales entre otras).
- Presupuesto disponible para el desarrollo del software.
- Fecha de disponibilidad del software importante para el negocio.
- Paradigmas de programación, aspectos de distribución, tecnología de autorización.
- Características del proceso de desarrollo o ciclo de vida a ser aplicado en el desarrollo del software (ágil, secuencial, iterativo, nivel de especificación del de proceso).
- Conocimiento y experiencia del equipo de desarrollo (relacionados al proceso de desarrollo, lenguajes de modelado y programación).
- Infraestructura existente en el entorno de implementación (hardware, software, u otros sistemas).
- Límites en tiempos de respuesta, rendimiento y el espacio de almacenamiento disponible.
- Disponibilidad de los componentes.
- Tipos de usuarios autorizados, o no, a realizar ciertas tareas y cómo se implementarán los mecanismos de seguridad, entre otros;

La lista anterior muestra que no es una tarea sencilla para un ingeniero de requerimientos identificarlos, evaluarlos y tipificarlos, si consideramos la falta de consenso en las definiciones y categorías expuestas en la sección anterior, más aún si el enfoque metodológico que se está utilizando para de desarrollo de la aplicación web carece de lineamientos, técnicas y/ o herramientas específicas de soporte a la etapa de ingeniería de requerimientos.

Para Cysneiros en [7] [30] los enfoques de desarrollo de software que tratan sobre Requerimientos No Funcionales, pueden clasificarse como “Orientados al Proceso” y “Orientados al Producto”.



Los orientados a productos, se enfocan en establecer el grado de cumplimiento del software con los Requerimientos No Funcionales detectados, mientras que los orientados al proceso, están enfocados en el proceso de desarrollo de software, ayudan a los ingenieros de software a buscar alternativas para cumplir los Requerimientos No Funcionales a medida que se desarrolla el software, lo cual permitirá además justificar decisiones de diseño.

A continuación, para asentar las bases conceptuales a aplicar en el desarrollo de este trabajo, se parte de las definiciones expuestas por Cysneiros en [7], descritas en la sección anterior en los ítems p) y q), que considera a los Requerimientos No Funcionales como requerimientos de calidad y como restricciones.

***Requerimientos No Funcionales como requerimientos de calidad:***

Para Cysneiros en p) los Requerimientos No Funcionales “*son requerimientos de calidad, que representan restricciones o las cualidades que el sistema debe tener tales como: precisión, usabilidad, seguridad, rendimiento, confiabilidad, performance entre otras*”...

Dentro de los tipos de requerimientos que consideran importantes para las aplicaciones web, Kappel, Pröll, Reich y Retschitzegger en [5]; mencionan a los *requerimientos de calidad*: como aquellos que describen el nivel de calidad del servicio y especifican propiedades relevantes que debe reunir el sistema en términos de las características del estándar de calidad internacional ISO/IEC 9126 [34], y aunque Kappel, Pröll, Reich y Retschitzegger en [5] hacen mención de la existencia de investigaciones que pretenden extender el modelo básico de ISO/IEC 9126 para las aplicaciones Web, presentan a la usabilidad, la performance y la seguridad como los aspectos críticos para las aplicaciones web.

La tabla 2.2 muestra los conceptos expuestos, con la finalidad de poner el marco de atributos de calidad comparables para aplicar en este trabajo.

	Cysneiros [7]	Kappel [5]	ISO/IEC 9126-1 [34]
Características o atributos de calidad	Precisión (*)	Usabilidad	Usabilidad
	Usabilidad	Performance (**)	Eficiencia
	Seguridad (*)	Seguridad (*)	Funcionalidad
	Rendimiento (**)		Confibilidad
	Confiabilidad		Mantenibilidad
	Performance (**)		Portabilidad

\* Subcaracterística de la Funcionalidad

\*\* Subcaracterística de la Eficiencia

TABLA 2.2: Requerimientos No Funcionales como requerimientos de Calidad.

En este trabajo se considera los Requerimientos No Funcionales como requerimientos de calidad y para ellos se contemplarán las seis características del estándar de calidad internacional ISO/IEC 9126 [34], descrito en la sección 2.1., ya que este, incluye los atributos mencionados en las definiciones de Cysneiros en [7] y Kappel en [5] y aporta dos características no consideradas por ellos como lo son: la Mantenibilidad y la Portabilidad.

***Requerimientos No Funcionales como restricciones:***

Para Cysneiros en q) *los Requerimientos No Funcionales son las restricciones globales sobre el sistema, un requerimiento funcional, el proceso de desarrollo o sobre el proceso de despliegue...*

De acuerdo a lo presentado en la sección 2.1, no hay ejemplos específicos expresados por Cysneiros o unicidad de criterio en la bibliografía existente de otros autores, para definir a las “restricciones globales del sistema”, las “restricciones sobre un requerimiento funcional”, las “restricciones de proceso de desarrollo” y las “restricciones del proceso de despliegue”; este trabajo se enfocará solamente en las “*restricciones del proceso de desarrollo*” para evaluar el nivel de completitud y de especificación de los procesos para desarrollar aplicaciones web, dado que la incompletitud de los mismos hace que se descuiden aspectos relevantes para el tratamiento de los Requerimientos No Funcionales.

Se proponen a continuación las restricciones de proceso de desarrollo a considerar en este trabajo:

*Alcance del proceso de desarrollo*: relacionada a las fases del ciclo de vida que presenta el proceso, el cual puede ser

*Parcial*: significa, que el proceso de desarrollo no contempla y describe fases para todo el ciclo de vida de desarrollo.

*Completo*: significa, que el proceso de desarrollo contempla y describe las fases para todo el ciclo de vida de desarrollo.

*Nivel de detalle de la especificación del proceso*: relacionada a la profundidad del detalle de la especificación del proceso, las actividades, las técnicas utilizadas en cada fase y los productos generados en las mismas. El cual puede ser:

*Parcial*: Significa, que el proceso de desarrollo describe solamente las actividades a ser aplicadas en cada fase, pero no describe técnicas a utilizar ni ofrece detalles o templates de los productos que deben ser producidos en cada fase.

*Completo*: Proceso especificado: significa, que el proceso de desarrollo describe las actividades, las técnicas a ser aplicadas y los productos que deben ser producidos en cada fase.

Los conceptos presentados en esta sección serán aplicados en los capítulos 4 y 5.

## Capítulo: 3

---

### 3. Técnicas utilizadas en la Ingeniería de Requerimientos

La Ingeniería de Requerimientos “es el proceso sistemático de desarrollar requerimientos a través de un proceso cooperativo e iterativo de analizar el problema, documentar las observaciones resultantes en una variedad de formatos de representación y chequear la precisión de la comprensión obtenida” [35].

La ingeniería de requerimientos comienza con la *elicitación* que es “el proceso de adquisición de todo el conocimiento relevante necesario para producir un modelo de requerimientos del dominio del problema” [35]. La esencia de la elicitación es la comprensión del dominio del problema, los ingenieros de software adquieren el conocimiento aplicando diferentes técnicas de elicitación a partir de los expertos de dominio o de diferentes fuentes (literatura sobre el dominio, software existente en el dominio, aplicaciones de software similares en otros dominios, estándares de desarrollo de software, entre otras). El modelo de requerimientos inicial, generado en la elicitación sirve de entrada a la *especificación* de requerimientos.

La especificación de requerimientos tiene como objetivo producir un modelo de especificación de requerimientos formal. Desarrollar una especificación consiste en la construcción de requerimientos en términos conceptuales (abstracciones, suposiciones, restricciones sobre el dominio de la aplicación); la especificación de requerimientos incluye los requerimientos funcionales, no funcionales y modelos de contexto, la cual será validada por el usuario en el proceso de *validación*.

La validación de requerimientos “es el proceso de certificar la corrección del modelo de requerimientos contra las intenciones de los usuarios” [35]. La actividad principal de este proceso es detectar y corregir errores del modelo de especificación de requerimientos, en la fase de requerimientos y no cuando el software este desarrollado. El modelo de requerimientos validado debe especificar una solución de software que satisface las necesidades del usuario.

A continuación se describen un grupo de técnicas utilizadas por la ingeniería de requerimientos, para la Elicitación, Especificación y Validación de requerimientos extraídas para su análisis a partir de la revisión de técnicas de ingeniería de requerimientos presentada por Escalona y Koch en [1].

**Técnicas de Elicitación:**

<b>Técnica</b>	<b>Entrevistas</b>
<b>Descripción</b>	Es una de las técnicas más utilizadas para tomar conocimiento del problema. Existen diversos tipos de entrevistas y son muchos los autores que han trabajado en definir su estructura para ofrecer guías para su correcta realización. Generalmente, la estructura de la entrevista consiste de tres pasos: la identificación de los entrevistados, la preparación de la entrevista, la realización de la entrevista y la documentación de los resultados.
<b>Comentarios</b>	La entrevista no es una técnica sencilla de aplicar, porque requiere que el entrevistador sea experimentado y tenga capacidad no solo para elegir bien a los entrevistados y obtener de ellos toda la información posible en un acotado período de tiempo, sino que además conozca técnicas o herramientas para la preparación de la entrevista y documentar los resultados.

<b>Técnica</b>	<b>JAD (Joint Application Development/Desarrollo conjunto de aplicaciones)</b>
<b>Descripción</b>	Es una alternativa a las entrevistas. Esta técnica consiste en una práctica de grupo que se desarrolla durante varios días y en la que participan varios roles del equipo de desarrollo (analistas, usuarios, administradores del sistema) y clientes. Se basa en cuatro principios: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación, mantener un proceso organizado y racional y por último una filosofía de documentación WYSIWYG ( <i>What You See Is What You Get, lo que ve es lo que obtiene</i> ). En otras palabras durante la entrevista se trabajará sobre lo que se generará. Después una fase de preparación para poder aplicar la técnica, el equipo de trabajo se reúne en varias oportunidades; en cada reunión se establecen los requerimientos de alto nivel, el contexto del problema y la documentación. Durante cada reunión se debate en grupo sobre estos temas llegando a una serie de conclusiones que se documentan, de esta forma en cada reunión se van concretando más las necesidades que deberá soportar el sistema.
<b>Comentarios</b>	Esta técnica presenta ventajas frente a las entrevistas tradicionales ya que permite un ahorro de tiempo al evitar que las opiniones de los clientes se tengan que contrastar por separado. Sus desventajas son por un lado, que requiere de grupos de participantes bien integrados y organizados y por otro requiere disponibilidad de tiempo de los mismos.

<b>Técnica</b>	<b>Brainstorming</b>
<b>Descripción</b>	Es una técnica para reunión de grupos de personas cuyo objetivo es que los participantes expongan sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. El grupo de personas a participar en estas reuniones no debe ser muy numeroso (máximo 10 personas) y una de ellas debe asumir el rol de moderador de la reunión, pero sin carácter de controlador.
<b>Comentarios</b>	La ventaja del Brainstorming como técnica de captura de requerimientos es que es sencilla de usar y de aplicar. La desventaja es que si bien ayuda a obtener una visión general de las necesidades del sistema, por lo general no sirve para identificar los detalles concretos del sistema, por lo que suele aplicarse solo en las primeras reuniones.

<b>Técnica</b>	<b>Concept Mapping (Mapas de Concepto)</b>
<b>Descripción</b>	Esta técnica se desarrolla con el equipo del proyecto y el usuario final, consiste en modelar los conceptos del sistema que está siendo construido mediante grafos; los vértices representan los conceptos y las aristas representan posibles relaciones entre dichos conceptos.
<b>Comentarios</b>	Es una técnica muy usada en la Ingeniería de Requerimientos porque que es de fácil comprensión para el usuario, más aún si es elaborada en el lenguaje de éste. Una desventaja, es que en algunos casos puede resultar ambigua si no se acompaña de una descripción textual adicional, sobre todo en sistemas complejos.

<b>Técnica</b>	<b>Sketches y Storyboards</b>
<b>Descripción</b>	Esta técnica consiste en representar sobre papel en forma muy esquemática las diferentes interfaces al usuario (sketches). Estos sketches pueden ser agrupados y unidos por enlaces dando idea de la estructura de navegación (storyboard). Es comúnmente usada por los diseñadores gráficos de aplicaciones web.
<b>Comentarios</b>	Es una técnica de fácil aplicación.

Técnica	Casos de Uso
Descripción	Aunque inicialmente se desarrollaron como técnica para la definición de requerimientos, algunos autores proponen casos de uso como técnica para la captura de requerimientos. Los casos de uso permiten mostrar el alcance del sistema (requerimientos funcionales expresados como casos de uso). Un caso de uso describe la secuencia de interacciones que se producen entre el sistema y los actores del mismo para realizar una función que produce un resultado de valor para el / los actores del mismo. Un actor es un rol, es un elemento externo (personas, otros sistemas, etc.) que interactúan con el sistema. Un actor puede participar en varios casos de uso y un caso de uso puede interactuar con varios actores.
Comentarios	La ventaja de los casos de uso, es que el modelo de casos de uso resulta de fácil comprensión para el usuario o cliente y para el equipo de desarrollo, la desventaja es que muchas veces carecen de la precisión necesaria, por lo que es necesario complementarlos con otras técnicas como diagramas de actividades o deben ser acompañados por información textual adicional.

Técnica	Cuestionarios y Checklists
Descripción	Esta técnica consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas o limitadas por opciones en el propio cuestionario o checklist. Este cuestionario puede ser completado durante una entrevista o puede ser utilizado para recoger/ capturar información de un grupo de personas en forma independiente a una entrevista.
Comentarios	Su aplicación requiere que el ingeniero de requerimientos conozca el dominio del problema en el que está trabajando y la aplicación a ser desarrollada a fin de preparar el cuestionario y checklist.

Técnica	Comparación de terminología
Descripción	Uno de los problemas que surge durante la elicitación de requerimientos es que los usuarios, expertos del dominio, no llegan a entenderse con los ingenieros de software debido a problemas de comunicación generados por no usar el mismo lenguaje. Esta técnica consiste en identificar el uso de palabras diferentes para referirse a los mismos conceptos (correspondencia) y palabras similares para referirse a diferentes conceptos (conflictos) o cuando no hay concordancia exacta en el vocabulario y los conceptos (contraste).
Comentarios	Esta técnica es utilizada en forma complementaria a otras para obtener consenso respecto de la terminología a ser usada en el proyecto de desarrollo.

**Especificación de requerimientos:** Dentro de las técnicas utilizadas se destacan las siguientes

<b>Técnica</b>	<b>Lenguaje natural</b>
<b>Descripción</b>	Consiste en definir los requerimientos en lenguaje natural sin usar reglas para ello. A pesar de que es criticada por su ambigüedad, es una de las técnicas más usadas para la especificación de requerimientos.
<b>Comentarios</b>	Es de fácil aplicación, pero resulta una técnica ambigua para la definición de los requerimientos.

<b>Técnica</b>	<b>Glosario y ontologías</b>
<b>Descripción</b>	El equipo de desarrollo de software puede ser muy diverso, esta diversidad trae aparejada la necesidad de establecer un marco de terminología común a ser utilizado en el proyecto. En las aplicaciones web el equipo de desarrollo suele ser más interdisciplinario que en los equipos para desarrollo de aplicaciones no web, por esta razón son muchas las metodologías que proponen desarrollar un glosario de términos en el que se recogen y definen los conceptos más relevantes y críticos para el sistema. En esta misma línea se propone también el uso de ontologías, en las que no sólo aparecen los términos, sino también las relaciones entre ellos.
<b>Comentarios</b>	Son de gran utilidad para establecer la terminología común y los conceptos relevantes a ser utilizado en el proyecto.

<b>Técnica</b>	<b>Plantillas o patrones</b>
<b>Descripción</b>	Esta técnica consiste en describir los objetivos y requerimientos usando el lenguaje natural pero de una forma estructurada. Una plantilla es una tabla con una serie de campos y una estructura predefinida, las secciones de esta plantilla son completadas por el equipo de desarrollo usando lenguaje natural y la terminología del usuario.
<b>Comentarios</b>	Las plantillas eliminan parte de la ambigüedad del lenguaje natural al estructurar la información; cuanto más estructurada sea ésta, menos ambigüedad ofrece, sin embargo si el nivel de detalle requerido es demasiado detallado, el trabajo de completar las plantillas y mantenerlas se dificulta.



Técnica	Escenarios
<b>Descripción</b>	La técnica de los escenarios consiste en describir las características del sistema a desarrollar mediante una secuencia de pasos. La representación de los escenarios puede variar de un autor a otro, pueden ser textuales o adoptar representaciones gráficas en forma de diagramas de flujo o casos de uso. La notación de escenarios está integrada en muchas técnicas de análisis orientados a objetos.
<b>Comentarios</b>	El análisis de los escenarios, puede ofrecer información importante sobre las necesidades funcionales de sistema. Cuando son representados en forma textual, pueden sumar ambigüedad a la definición de los requerimientos.

Técnica	Casos de uso
<b>Descripción</b>	Esta técnica tiene gran aceptación tanto en la elicitación (ver técnicas de Elicitación) como en la definición de requerimientos. Se ha propuesto como técnica básica del proceso RUP para describir las actividades que debe realizar el sistema para producir un resultado de valor para un actor. Dado que varios autores consideran que los casos de uso pueden resultar ambiguos a la hora de definir los requerimientos, las metodologías que lo utilizan proponen agregar a los mismos, descripciones textuales basadas en plantillas, diccionarios de datos o representaciones gráficas para eliminar su ambigüedad.
<b>Comentarios</b>	La desventaja de esta técnica es que resulta ambigua cuando definen requerimientos complejos; por lo que es necesario complementarlos con descripciones textuales basadas en plantillas, diccionarios de datos o representaciones gráficas.

Técnica	Lenguajes Formales
<b>Descripción</b>	En contraste a las descripciones en lenguaje natural, hay técnicas que utilizan lenguajes formales para describir los requerimientos del sistema. Las especificaciones algebraicas como ejemplo de técnicas de descripción formal, han sido aplicadas en la ingeniería de software desde hace años, sin embargo, utilizarlas resulta una actividad compleja y son de difícil comprensión para el usuario/cliente.
<b>Comentarios</b>	La ventaja es que es una de las formas menos ambigua para representar los requerimientos, lo cual permite la aplicación de técnicas de verificación automatizadas. La desventaja más notable de los Lenguajes Formales es que no favorecen la comunicación entre el usuario/cliente y el ingeniero de requerimientos.

**Validación de requerimientos:** El paso siguiente, una vez que los requerimientos fueron definidos es validarlos, dentro de las técnicas apropiadas para ello se encuentran las siguientes:

<b>Técnica</b>	<b>Reviews o Walk-throughs</b>
<b>Descripción</b>	Esta técnica consiste en la lectura y corrección de la documentación que contiene los requerimientos especificados y los modelos generados. Permite validar únicamente la correcta interpretación de la información transmitida. La verificación de consistencia de la documentación o información faltante demanda de métodos más sofisticados.
<b>Comentarios</b>	Debe complementarse con otras técnicas para la verificación de consistencia e identificar la información faltante.

<b>Técnica</b>	<b>Auditorías</b>
<b>Descripción</b>	La revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra checklists predefinidos o definidos al comienzo del proceso.
<b>Comentarios</b>	La revisión de la información y resultados puede ser parcial o total, esto depende de la preparación de la auditoria y el nivel de detalle de los checklists utilizados para aplicar esta técnica.

<b>Técnica</b>	<b>Matrices de trazabilidad</b>
<b>Descripción</b>	Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requerimientos del mismo. Es necesario ir viendo qué objetivos cubre cada requerimiento, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos.
<b>Comentarios</b>	Permite identificar la consistencia de los requerimientos especificados con los objetivos del sistema. Debe complementarse con otras técnicas para la verificación de consistencia e identificar la información faltante.

<b>Técnica</b>	<b>Prototipos</b>
<b>Descripción</b>	Es una técnica que consiste en construir herramientas basadas en la especificación de requerimientos. Los prototipos por lo general se construyen a partir de un conjunto parcial de los requerimientos funcionales del sistema, pero permiten al usuario tener una idea global de la estructura de la interfaz de usuario del sistema. Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.
<b>Comentarios</b>	Permite validar un subconjunto de requerimientos de los requerimientos especificados. Debe complementarse con otras técnicas para la verificación de consistencia e identificar la información faltante.

**Herramientas de Administración de requerimientos:** Las herramientas propuestas para la administración de requerimientos se presentarán en el capítulo siguiente, junto a la metodología a la que dan soporte.

## Capítulo: 4

---

### 4. Metodologías de desarrollo de aplicaciones Web y los requerimientos.

#### 4.1 Aplicaciones Web

Una aplicación Web es un sistema de software basado en tecnologías y estándares del World Wide Web Consortium (W3C), que proporciona recursos web específicos, tales como los contenidos y servicios a través de una interfaz de usuario, el navegador Web [5]. Dicho en otras palabras una aplicación web en la actualidad es un complejo sistema de software que se codifica en un lenguaje soportado por navegadores web, donde la característica distintiva más relevante comparada con aplicaciones de software tradicionales, es la forma en la cual la aplicación web es usada.

Existen diferentes tipos de aplicaciones web, unas pueden ser de carácter puramente informativas otras pueden manejar aplicaciones de comercio electrónico, entre otras, Kappel, Pröll, Reich y Retschitzegger en [5] presentan una categorización de aplicaciones web organizada en función a la historia y evolución del grado de complejidad de las mismas:

1. Sitios Web centrados en documentación.
2. Aplicaciones Web interactivas.
3. Aplicaciones web transaccionales.
4. Aplicaciones web basadas en workflow
5. Aplicaciones web colaborativas
6. Web sociales
7. Aplicaciones web orientadas a portales
8. Aplicaciones web ubiquitous, basados en la localización
9. Web semánticas

Desde el punto de vista de la ingeniería del software el desarrollo de las aplicaciones web representan un nuevo dominio de aplicación [5] si bien tiene similitud con el desarrollo de las aplicaciones tradicionales, la característica especial de las aplicaciones web

hace surgir la necesidad de una adaptación de los enfoques de desarrollo existentes o quizás el desarrollo de enfoques metodológicos nuevos. El desarrollo de aplicaciones web tiene que contemplar varias características que difieren respecto del desarrollo de otros tipos de aplicaciones (no web). Por ejemplo:

- Los diferentes tipos de stakeholders involucrados en el proceso de desarrollo (cliente, usuarios, analistas, diseñadores gráficos, desarrolladores, expertos en multimedia y seguridad, marketing, etc.)
- La estructura de navegación, en la cual debe ser clara e intuitiva para el usuario.
- La interfaz gráfica de usuario, cuyo diseño debe tener en cuenta, aspectos de marketing y multimedia,

Los dos últimos si bien no solo deben ser tratados en forma diferente en la etapa de diseño, sino que deberían considerarse como parte de la aplicación de la ingeniería de requerimientos.

Si bien actualmente existen varias metodologías de desarrollo web en vigencia, en la sección siguiente, se analizarán un subconjunto de enfoques metodológicos considerados a partir del estudio comparativo realizado por Escalona y Koch en [1], a los efectos de identificar como estos enfoques, están aplicando la ingeniería de requerimientos para el tratamiento de los Requerimientos No Funcionales que son el objetivo de este trabajo.

## **4.2 Metodologías de desarrollo Web y los RNF**

El trabajo de Escalona y Koch [1], aporta una comparación de diez metodologías de desarrollo web, destacando entre otros aspectos, técnicas de ingeniería de requerimientos y tipos de requerimientos que manejan. Respecto a los Requerimientos No Funcionales se limita a mencionar cuales metodologías detectan los mismos.

En este capítulo se analizan puntualmente los enfoques que mencionan Requerimientos No Funcionales en [1] a efectos de determinar cuál es el concepto que estas

metodologías tienen de los distintos tipos de Requerimientos No Funcionales y las técnicas de detección, especificación, validación y gestión de requerimientos que aplican las mismas.

La mayoría de las metodologías comparadas en el trabajo de Escalona y Koch [1] tienen una clasificación de requerimientos que difiere entre una y otra, los autores propusieron una clasificación general de requerimientos, a fin de poder realizar una comparación sobre cuales tipos de requerimientos utiliza cada enfoque metodológico.

A continuación se describe la clasificación realizada en [1] para los requerimientos de las aplicaciones Web:

- Se identifica por un lado, los Requerimientos Funcionales: *como capacidades que el sistema debe exhibir para resolver un problema*. Estos a su vez fueron subclasificados en:
  - Requerimientos de datos, *los cuales establecen “como” la información es almacenada y administrada por la aplicación. También son denominados requerimientos de contenido, requerimientos conceptuales o requerimientos de almacenamiento de información.*
  - Requerimientos de Interface de usuario, *que describen “como” el usuario va a interactuar con la aplicación Web. También conocidos en algunas propuestas como requerimientos de interacción o requerimientos de usuario.*
  - Requerimientos de Navegación, *que representan la navegación que los usuarios requieren a través de la aplicación Web.*
  - Requerimientos de Personalización, *describen “como” la aplicación Web debe adaptarse dinámicamente, dependiendo del usuario que interactúe con ella.*
  - Requerimientos Transaccionales, *describen “que” es lo que la aplicación Web tiene que realizar internamente sin considerar interfaces y aspectos de interacción. Son conocidos además como requerimientos de servicios.*
  
- Y los Requerimientos No Funcionales, *los cuales son considerados como restricciones a la solución, los autores mencionan como ejemplos: los requerimientos de portabilidad, los requerimientos de reutilización, requerimientos de usabilidad, requerimientos de disponibilidad, requerimientos de performance etc.*

La tabla 4.1, fue extraída del análisis comparativo realizado en [1] por Escalona y Koch. Presenta un resumen de los diferentes tipos de requerimientos que son tratados por cada una de las metodologías estudiadas. Los enfoques metodológicos están ordenados cronológicamente y de esta forma puede verse la evolución e importancia que fueron adquiriendo los requerimientos en el entorno de la Web.

	Req. Datos	Req. Interfaz usuario	Req. Navegacionales	Req. Personalización	Req. Transaccionales	Req. No Funcionales
WSDM	✓			✓		✓
SOHDM	✓	✓			✓	
RNA	✓	✓	✓		✓	
HFPM	✓	✓	✓			✓
OOHDM	✓	✓	✓			
UWE	✓	✓	✓	✓		✓
W2000			✓	✓	✓	
WebML	✓	✓		✓		✓
NDT	✓	✓	✓	✓	✓	✓
DDDP	✓	✓	✓	✓	✓	✓

TABLA 4.1: Requerimientos tratados por cada metodología

La columna titulada “Req. No funcionales” de la tabla 4.1, muestra cuales de las metodologías para desarrollo de aplicaciones web estudiadas, identifican Requerimientos No Funcionales, de ahí que: WSDM, HFPM, UWE, WebML, NDT, DDDP serán a partir de ahora el foco de análisis.

La figura 4.1, es una representación gráfica que surge a partir de la tabla 4.1; ilustra la relación entre las metodologías Web y los requerimientos. El eje vertical representa el tiempo y permite identificar el año en el que surgieron los diferentes enfoques metodológicos; el eje horizontal representa a los requerimientos clasificados como funcionales y no funcionales. En el espacio se destaca la relación de las metodologías y los requerimientos manejados por cada una, se han señalado con azul aquellas que son el foco de nuestro estudio.

Durante el desarrollo de este trabajo se pretende evolucionar la tabla 4.1 detallando los tipos de Requerimientos No Funcionales mencionados o tratados en cada enfoque.

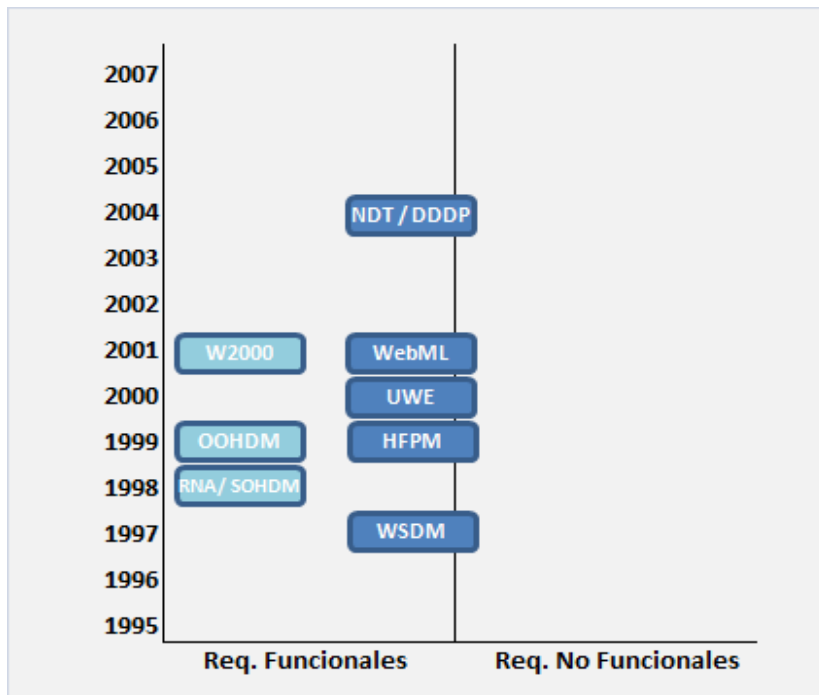


FIGURA 4.1: Metodologías para aplicaciones Web y los Requerimientos

### Análisis de los enfoques metodológicos para desarrollo de aplicaciones Web

El análisis de las metodologías WSDM, HFPM, UWE, WebML, NDT, DDDP se realizará en base a la siguiente estructura de presentación:

- ¿Cuáles son las características relevantes?: *En esta sección se enumeran los aspectos y o cualidades significativas y representativas del enfoque metodológico.*
- Descripción de las fases del proceso: *En esta sección se presenta un grafico que ilustra las fases del ciclo de vida del enfoque metodológico, se describen las actividades relevantes y/o productos generados en cada una de ellas.*



*Dado que los diferentes enfoques metodológicos presentados en este trabajo utilizan los términos fases o etapas en la descripción de su ciclo de desarrollo, proponemos el término “Fase” a fin de unificar términos comparables.*

- *¿Qué tipos de Requerimientos No Funcionales identifica?: En esta sección se detallan los diferentes tipos de Requerimientos No Funcionales que se identifican en las diferentes fases del ciclo de vida del enfoque metodológico.*
- *¿Qué técnicas de análisis y administración de requerimientos emplea?: En esta sección se detallan las técnicas de elicitación, especificación, validación y administración de requerimientos que menciona y emplea cada enfoque en la fase de requerimientos. Se pone foco en evaluar si el enfoque utiliza técnicas específicas para elicitación, especificación y validación de Requerimientos No Funcionales.*

La presentación de cada enfoque sigue el orden cronológico de publicación, lo cual permite visualizar y analizar cómo fue evolucionando la ingeniería de requerimientos aplicada a metodologías web.

#### **4.2.1 WSDM: Web Site Design Method**

Este enfoque fue propuesto en el año 1997 por De Troyer y Leune [9], presenta un tratamiento interesante de la visión de los usuarios; plantea el diseño de una aplicación Web de tipo Kiosco” a partir de los potenciales grupos de usuarios.

##### ***¿Cuáles son las características relevantes?***

- El proceso consiste en cuatro fases.
- Sirve para desarrollar kioscos web, es decir, aplicaciones que muestran una determinada información al usuario y les permiten navegar hacia ella.

- El proceso de definición de requerimientos se centra en detectar los perfiles de usuarios potenciales mediante dos tareas:
  - Identificar y clasificar usuarios mediante el estudio del entorno
  - Describir las características de los grupos de usuarios detectados y sus necesidades de información, lo cual se denomina “perspectivas”.
- El diseño conceptual comienza a partir de las perspectivas detectadas.
- La implementación puede ser generada una vez que el diseño conceptual está completo.
- “Es un enfoque centrado en el usuario esto no es lo mismo que a un enfoque conducido por el usuario”... [9].

La figura 4.2 muestra un diagrama de las fases y subfases que propone WSDM.

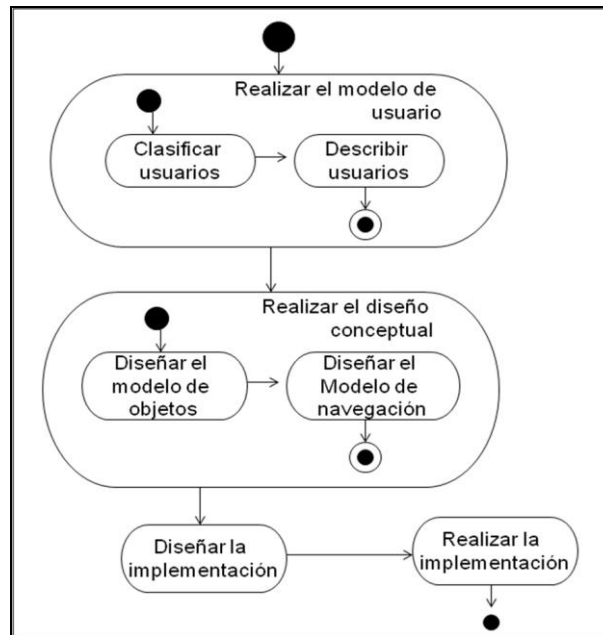


FIGURA 4.2: Diagrama de las fases de WSDM

A continuación se presentan y describen en forma general las actividades desarrolladas en cada una de las fases del proceso WSDM.

**Realizar el modelo de usuario:** Esta Fase se concentra en los potenciales usuarios de la aplicación web, a fin identificarlos, clasificarlos y luego describir sus características y los requerimientos de información, para lo cual cuenta con dos subfases:

- Clasificar usuarios: WSDM propone estudiar del entorno de la organización o los procesos de negocio para los cuales se implantará la aplicación web a fin de identificar las actividades y las personas (potenciales usuarios) involucradas en cada una. Luego describe las relaciones entre usuarios y las actividades que realizan los mismos; para la representación gráfica de estas relaciones WSDM propone mapas de conceptos de roles y actividades.
- Describir usuarios: Los grupos de usuarios detectados en la subfase anterior son descritos con más detalle. Mediante un diccionario de datos, se describen los requerimientos de información, requerimientos funcionales y requerimientos de seguridad, requerimientos de usabilidad para cada grupo de usuarios.

En el trabajo de [9] algunos ejemplos de las características que son analizadas para cada grupo de usuarios son: nivel de experiencia con sitios web en general, frecuencia de uso de esta la aplicación web, mandatorio/ discreto uso del sitio web, Motivación para el uso de la aplicación web, lenguajes, nivel de educación, habilidades, edad.

Se hace hincapié en este punto el cual es relevante para este trabajo.

**Realizar el diseño conceptual:** Durante esta fase se realiza el modelo conceptual de los requerimientos detectados en la fase anterior y se realiza el diseño de la navegación que describe como los diferentes grupos de usuarios podrán navegar a través de la aplicación web. Para ello plantea las siguientes subfases:

- Diseñar modelo de objetos: El propósito de esta fase es desarrollar un modelo de objetos conceptual para los diferentes grupos de usuarios, los mismos son llamados MOU (Modelos de objetos usuario). El modelo conceptual describe diferentes tipos de objetos, las relaciones, restricciones entre ellos y describe el comportamiento de los objetos.

Un punto a destacar es que un grupo de usuarios puede tener diferentes perspectivas generando así diferentes requerimientos de usabilidad. Para este caso se pueden diseñar MPO (Modelos de Perspectiva de Objetos).

- Diseñar modelo de navegación: El modelo de navegación que se genera en esta fase muestra un número de pistas de navegación por cada MPO. Dicho en otras palabras, expresa como un usuario de un grupo en particular puede navegar hacia la información que necesita. Se modela a través de componentes (de información, de navegación y externos) y links o enlaces que conectan los mismos. Un modelo de navegación cuenta de tres capas: Capa de contexto es el nivel más alto de las pistas de navegación, la capa de navegación que proporciona las diferentes formas para acceder a la información (conecta la capa de contexto con la información) y la capa de Información contiene los componentes de información.

Puede existir más de un modelo de navegación, dependiendo de los grupos de usuarios detectados durante la primera fase.

***Diseñar la implementación:*** Durante esta fase se realiza el diseño de interface (look and feel) de la aplicación web para cada uno de los grupos de usuarios definidos. El resultado es el Modelo de implementación; el cual ayuda a que los usuarios tengan un modelo mental de la aplicación.

Una aplicación Web construida con WSDM consiste de un número de pistas de navegación para cada perspectiva de usuario identificada.

Es importante destacar que el diseño de la implementación puede depender de las limitaciones del lenguaje de programación seleccionado.

***Realizar la implementación:*** Durante esta fase se realiza la codificación del modelo obtenido en la fase anterior.

Se deben tener en cuenta las consideraciones y buenas prácticas para mejorar la mantenibilidad de la información que la aplicación web brindará a los diferentes usuarios, a fin de que conservarla actualizada.

*¿Qué tipos de Requerimientos No Funcionales identifica?*

- Usabilidad - (Relevamiento)
- Seguridad - (Relevamiento)
- Mantenibilidad - (Diseño - Implementación)
- Restricciones: De proceso, Lenguaje de programación.

Las restricciones y/o limitaciones del lenguaje de programación y necesidades de Mantenibilidad surgen en el momento de diseñar la implementación e implementación propiamente dicha.

El Modelo de Implementación puede depender de las limitaciones del lenguaje de programación.

*¿Qué técnicas de Análisis y administración de requerimientos emplea?*

***Requerimientos Funcionales***

- Elicitación/ Captura de requerimientos: Entrevistas.
- Especificación/Definición de requerimientos: Utiliza lenguaje natural.
- Validación: No menciona el empleo de técnicas para la validación de requerimientos.

***Requerimientos No Funcionales***

- Elicitación/Captura: No especifica
- Especificación/ Definición: No especifica.
- Validación: No especifica.

***Administración de requerimientos:*** No menciona técnicas o herramientas.

## **4.2.2 HFPM: Hypermedia Flexible Process Modeling**

Propuesto por Olsina [10] en año 1998, este enfoque metodológico presenta fases para todo el proceso de desarrollo; define un conjunto de actividades que van desde el análisis hasta el desarrollo de documentación y el mantenimiento.

**¿Cuáles son las características relevantes?**

- El proceso detallado cubre todo el ciclo de vida de desarrollo de la aplicación.
- Está compuesto por 13 fases.
- Describe las tareas y subtareas a realizar en cada fase pero, no ofrece modelos o técnicas de modelado.
- Incluye al usuario desde el principio del proceso de desarrollo.
- Si bien no define lineamientos de modelado, presenta la necesidad de definir modelos específicos para el usuario.
- Presenta la necesidad de elaborar la documentación específica a modelos y actividades desarrolladas en el ciclo de vida.
- Aunque la presentación de las fases es secuencial, HFPM permite la realización de iteraciones en el proceso de desarrollo.

La figura 4.3 muestra un diagrama de las fases que propone HFPM,

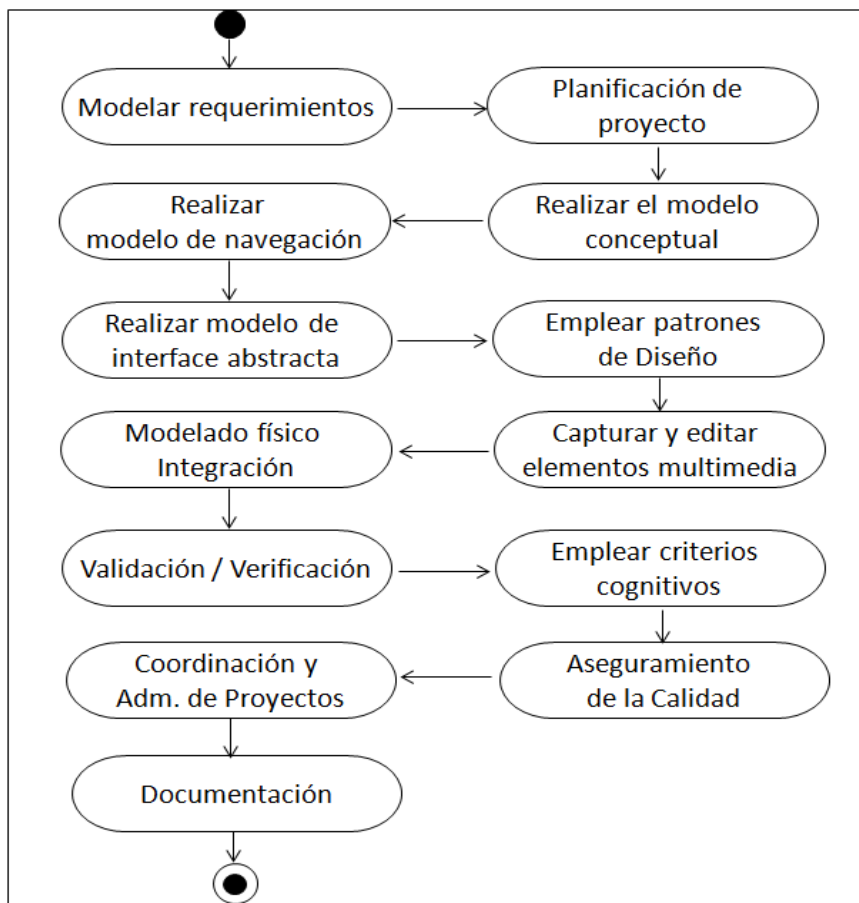


FIGURA 4.3: Diagrama de las fases de HFPM

A continuación se presentan y describen en forma general las actividades propuestas para cada una.

**Modelar los requerimientos:** En esta fase se proponen las siguientes tareas:

- Realizar encuestas iniciales.
- Describir los requerimientos funcionales mediante casos de usos.
- Modelar un glosario.
- Modelar la interface de usuario.
- Modelar los Requerimientos No Funcionales. En éstos incluyen la performance, la navegación, la reutilización, etc.

En principio no da ninguna norma a seguir para realizar estas tareas, dejando flexible las representaciones.

**Planificación de proyecto:** En esta fase se analizan y se especifican las actividades del plan de proyecto; no presenta ningún lineamiento o proceso a seguir para realizar el mismo.

**Realizar el modelado conceptual:** En esta fase se genera el modelo de clases que representa la aplicación. No se establecen los lineamientos a seguir para el realizar el modelo, se menciona que el objetivo de esta fase es similar a la fase de modelado conceptual de OOHDM (Object Oriented Hypermedia Design Model) en el cual, el modelo conceptual es representado como un modelo de clases para mostrar el aspecto estático del sistema. El enfoque OOHDM [11] no será descrito en este trabajo, dado que en el estudio realizado por Escalona y Koch no se presenta que OOHDM identifique los Requerimientos No Funcionales.

Las tareas a que propone HFPM en esta fase son:

- Analizar el dominio del problema.
- Realizar el modelo de dominio, mediante un modelo de clases.

**Realizar el modelado de navegación:** Esta fase propone realizar el modelo de navegación, el cual representará las posibilidades de navegación de la aplicación. Para realizar el modelo se menciona que los enfoques OOHDM, EORM (Enhanced Object Relationship Methodology) [12] y RMM (Relationship Management Methodology) [13] ofrecen procesos que pueden dar

soporte a las tareas de esta fase. Los procesos EORM [12] y RMM [13] no serán descriptos en este trabajo.

Las actividades de esta fase son:

- Analizar las necesidades de los usuarios.
- Identificar las clases de navegación.
- Especificar las clases de navegación y de esquema.
- Analizar las transformaciones de navegación.
- Especificar las transformaciones de navegación.

***Realizar el modelo de interfaz abstracta:*** El objetivo de esta fase es realizar un diseño de la interface. Las tareas que se proponen son:

- Analizar las necesidades de interfaz de usuario.
- Analizar y especificar los objetos de interfaz, los eventos y las transformaciones mediante el diseño de prototipos.
- Realizar el modelo de interfaz

***Emplear patrones de diseño:*** El objetivo de esta fase es emplear patrones de diseño a fin de enriquecer los modelos obtenidos en las fases anteriores. En esta fase se decide y diseña la arquitectura que tendrá la aplicación. Las tareas son las siguientes:

- Aplicar patrones de diseño para mejorar el modelo de interfaz de usuario.
- Emplear patrones de navegación y patrones de arquitectura
- Diseñar la arquitectura.

***Capturar y editar elementos multimedia:*** En esta fase se plantean los múltiples medios con los que se va a trabajar la aplicación, tales como medios de almacenamiento que se usarán en la misma. No describe tareas.

***Modelado físico / Integración:*** Esta fase si bien no especifica tareas, implica la creación de prototipos funcionales, prototipos evolutivos, bocetos o storyboarding e Integración de componentes.



**Verificación y validación:** En esta fase se va a analizar si el resultado es adecuado en base a los requerimientos detectados en la primera fase del proceso. No se establecen lineamientos a seguir dentro de la propuesta HFPM.

**Emplear de criterios cognitivos:** Esta fase propone emplear criterios de coherencia, cohesión y navegación. No especifica lineamientos para ello.

**Aseguramiento de la calidad:** Esta fase si bien HFPM no especifica qué entiende por calidad, ni como evaluar la calidad del producto propone estas actividades:

- Analizar y determinar las estrategias de calidad y mejora
- Especificar el plan de calidad

**Coordinación y gestión de proyectos:** En general, esta fase involucra las actividades de administrar y controlar los procesos, artefactos y Recursos. No se especifican pautas para ello.

**Documentación:** Esta fase involucra las tareas para generar la documentación de la aplicación. Las cuales implican:

- La documentación de los modelos generados en cada fase.
- La documentación del resultado de las pruebas y las valoraciones realizadas en
- las fases: Verificación/Validación y Aseguramiento de la calidad.

### ***¿Qué tipos de Requerimientos No Funcionales identifica?***

- Performance – (Relevamiento)
- Usabilidad – (Relevamiento - Diseño) si bien no es mencionado expresamente, se infieren cuando se plantea la actividad de Modelar la interfaz de usuario y considerar las necesidades del usuario para realizar el modelo de navegación.
- Reutilización – (Requerimientos)

Para realizar el modelo de navegación propone utilizar las bases de otros enfoques OOHDM, EORM (Enhanced Object Relationship Methodology) [12] y RMM (Relationship Management Methodology).

### *¿Qué técnicas de análisis y administración de requerimientos emplea?*

#### ***Requerimientos Funcionales***

- Elicitación/ Captura de requerimientos: Encuesta,
- Especificación/ Definición de requerimientos: Utiliza, Glosario, Casos de uso de análisis, Interfaces sketches.
- Validación: Utiliza prototipos para la validación de requerimientos.

#### ***Requerimientos No Funcionales***

- Elicitación/Captura: No especifica
- Especificación/ Definición: No especifica.
- Validación: Utiliza Prototipos.

***Administración de requerimientos:*** No menciona técnicas o herramientas.

## **4.2.3 UWE: UML-based Web Engineering**

Propuesto por Koch [14], Koch y Wirsing [15] años 2000-2001, este enfoque metodológico está basado en el proceso de desarrollo unificado [17] y el lenguaje de modelado UML (Unified Modeling Language) [18] ambos, adaptados para aplicaciones Web.

Este proceso, además ha evolucionado hacia el desarrollo MDD (Model-driven software development) [16]<sup>1</sup>, donde define los conceptos en base a un conjunto modelos.

### *¿Cuáles son las características relevantes?*

- El proceso de UWE alcanza todo el ciclo de vida de desarrollo de aplicaciones web.
- Está compuesto por 3 fases o etapas.
- Se centra en el usuario en la etapa de la captura de requerimientos [20].
- Centra su atención en aplicaciones Web personalizadas o adaptativas.

---

<sup>1</sup> En [16] This article is an extended version of “Transformation Techniques in the Model-Driven Development Process of UWE” presented at the 2<sup>nd</sup> Workshop on Model-Driven Web Engineering (MDWE’06) at the 6<sup>th</sup> International Conference on Web Engineering (ICWE 2006).

- En la fase de captura de requerimientos, hace la separación entre actividades de elicitación, especificación y validación de requerimientos.
- Utiliza UML extendido denominado UML Profile, para generar los modelos propuestos en el proceso.
- El proceso está más enfocado a actividades de análisis y de diseño utiliza el paradigma de orientación a objetos.
- Los modelos son refinados en las iteraciones que plantea el proceso de desarrollo.
- Existe una herramienta Case, denominada ArgoUML [19] que soporta UWE.

La figura 4.4 muestra un diagrama de las etapas de UWE.

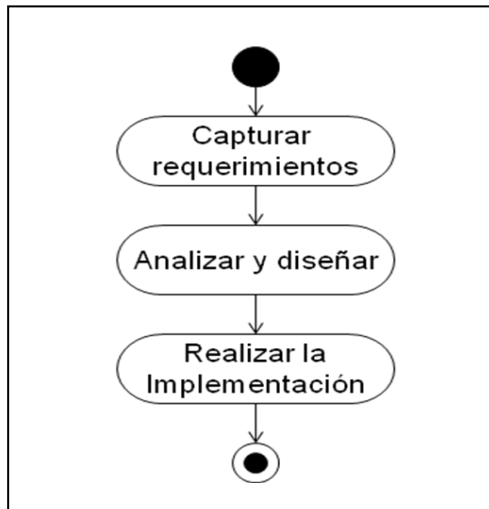


FIGURA 4.4: Diagrama de Proceso de UWE

A continuación se describen en forma general las fases/ etapas del proceso.

**Capturar requerimientos:** Esta etapa tiene por objetivo encontrar los requerimientos que la aplicación Web debe satisfacer y representarlos a través de un modelo de casos de uso, dado que es una técnica centrada en el usuario que permite definir quienes serán los *actores* que van a interactuar con la aplicación y representa la funcionalidad que la aplicación Web deberá cumplir para cada actor.

Como técnicas de captura, especificación y validación de requerimientos UWE propone [1]: para captura *entrevistas, cuestionarios y checklists*; para la especificación *glosarios, escenarios y casos de usos*, para validación *Walk-Through, auditorias y prototipos*.

UWE identifica requerimientos funcionales (contenido, presentación, navegación, adaptación) y no funcionales, si bien no establece lineamientos para representar los Requerimientos No Funcionales, los mismos serán considerados en las etapas siguientes.

**Analizar y diseñar:** Esta etapa toma como entrada el modelo de casos de uso a fin de generar: El modelo conceptual del dominio: Debe tratar de no tomar en cuenta los aspectos de navegación, presentación y adaptación. El resultado es un modelo de clases UML que contiene los atributos, operaciones de cada clase, además muestra gráficamente las relaciones de asociación, dependencia, jerarquía entre las clases del modelo; identifica interfaces y restricciones.

El modelo de navegación: Este se diseña a partir del modelo conceptual, las decisiones de diseño están basadas en modelo conceptual, modelo de casos y los requerimientos de navegación. El resultado es un modelo que muestra un diagrama de clases UML que representa las clases de navegación y las asociaciones entre estas.

El modelo de estructura de navegación: Describe como es la navegación a través del espacio de navegación. Dicho en otras palabras muestra como el usuario alcanzará las clases definidas en el modelo de navegación. Este modelo sirve de guía para construir el modelo de Presentación.

El modelo de presentación: Se enfoca en visualizar como es la organización estructural no en la apariencia física (formato, color etc.). Hay muchas posibilidades de construir un Modelo de presentación en base a Modelo de estructura de navegación dado. Los autores [20] proponen dividir el modelo en dos. Por un lado presentar el usuario el árbol de navegación y por otro el contenido. La profundidad del árbol de navegación en esta etapa de modelado debe ser limitada. El resultado es un Diagrama UML de clases construidas con estereotipos UML para elementos de presentación.

**Realizar la implementación:** Esta consiste en implementar los modelos validados e integrarlos. Se implementan desde los prototipos de interfaz de usuario hasta el código.

***¿Qué tipos de Requerimientos No Funcionales identifica?***

- Seguridad - (Relevamiento).
- Usabilidad – (Relevamiento - Diseño).

UWE identifica requerimientos de presentación, navegación, como requerimientos funcionales.

***¿Qué técnicas de análisis y administración de requerimientos emplea?***

***Requerimientos Funcionales***

- Elicitación/ Captura de requerimientos: entrevistas, cuestionarios y checklists;
- Especificación/ Definición: Glosarios, escenarios y casos de usos,
- Validación: Walk-Through, auditorias y prototipos.

***Requerimientos No Funcionales***

- Elicitación/ Captura de requerimientos: No especifica.
- Especificación/ Definición: No especifica.
- Validación: Utiliza Prototipos.

***Administración de requerimientos:*** UWE utiliza ArgoUML [19] herramienta case (No comercial) la cual:

- Permite la administración de los requerimientos y modelos generados a partir de los mismos.
- Soporta la notación de UWE para diseñar aplicaciones Web.
- Implementa el proceso de desarrollo de UWE
- Permite modelar los conceptos de (Contexto, navegación y presentación)
- La desventaja es que esta herramienta no ha evolucionado a UML 2.0

#### 4.2.4 WebML: Web Modeling Language

Propuesto en el año 2000, por Ceri, Fraternali y Bongio [21], quienes lo presentan como una notación para especificar sitios web complejos a un nivel conceptual.

Los elementos de WebML están asociados a una notación gráfica y utiliza sintáxis de XML (eXtensible Markup Language).

WebML presenta un proceso de desarrollo de aplicaciones Web dirigido por modelos.

##### *¿Cuáles son las características relevantes?*

- El proceso de WebML alcanza todo el ciclo de vida de desarrollo de aplicaciones Web.
- Permite la descripción en alto nivel de una aplicación web, bajo perspectivas ortogonales, tales como: el contexto de datos, las páginas que la componen, los link o enlaces entre las páginas, la presentación gráfica por página y la personalización por grupo de usuarios.
- Cubre aspectos avanzados del modelado de aplicaciones Web incluyendo presentación, modelado de usuarios y personalización.
- Es compatible con la notación clásica del modelo de E/R y UML [18].
- El proceso de diseño consta de 4 fases.
- Considera por un lado la captura de requerimientos y por otro la especificación a la cual denomina *Análisis de requerimientos*, donde utiliza casos de usos y diagrama de actividades para expresar el workflow de casos de uso complejos.
- Propone test de aceptación para validar los requerimientos funcionales y no funcionales.
- Esta soportado por dos herramientas CASE TotiiSoft [23] y WebRatio [24].

La figura 4.5 muestra un diagrama de las fases de WebML.

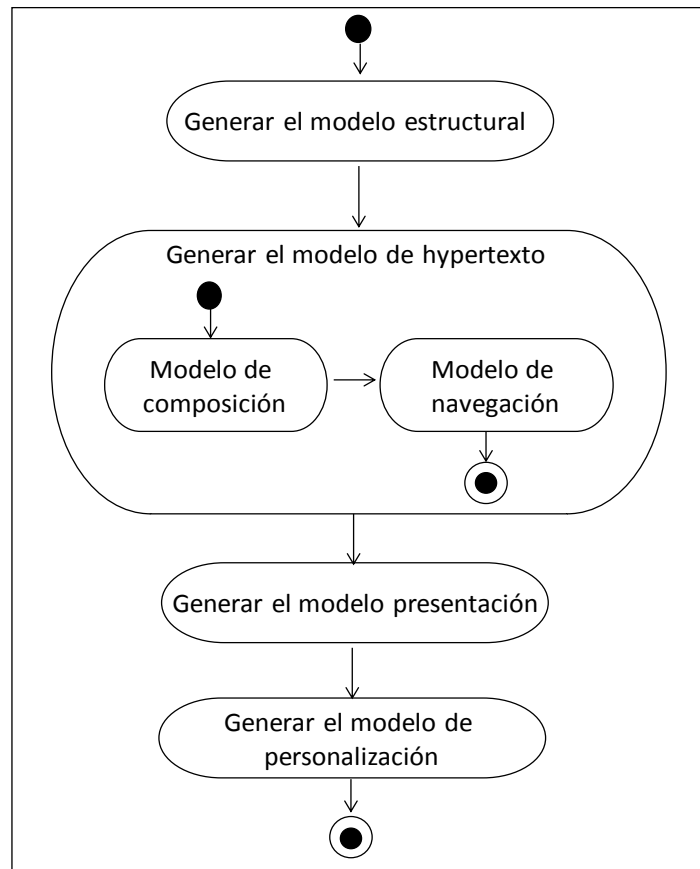


FIGURA 4.5: Diagrama de Proceso de WebML

A continuación se describen las cuatro fases o perspectivas de diseño de que propone WebML, las mismas se complementan con las actividades de captura de Requerimientos y validación propuestas en [22].

*Captura y Análisis de requerimientos:* Comienza con la identificación de usuarios y grupos y personalización de sus necesidades, para realizar la recolección de requerimientos propone técnicas como entrevistas, análisis de documentación y checklists de captura de requerimientos. Identifica requerimientos funcionales, de estructura o navegación y Requerimientos No Funcionales tales como: Performance, Disponibilidad, Escalabilidad, Mantenibilidad y Seguridad.

La especificación de los requerimientos (es denominada análisis) en esta etapa se obtiene un conjunto de especificaciones semi-formales, que incluyen, la lista de grupos de usuarios con sus derechos de acceso, un diccionario de datos de los objetos más relevantes del dominio de

la aplicación, una especificación informal de las vistas del sitio, una lista de Requerimientos No Funcionales, un conjunto de guías de presentación que dan indicaciones para el look and feel de la interface a ser desarrollada. Utiliza casos de usos para expresar las interacciones entre los grupos de usuarios y la aplicación, dependiendo de la complejidad del caso de uso, se modelan diagramas de actividad del workflow de proceso. Se utiliza notación UML.

*Validación de requerimientos:* propone elaborar test de aceptación para validar los Requerimientos No Funcionales.

***Generar el modelo estructural:*** Pone foco en las entidades relevantes del contexto de datos de la aplicación Web. Representa mediante notación del modelo de E/R y clases de UML las entidades y relaciones entre estas.

***Generar el modelo de hipertexto:*** Describe los hipertextos que pueden ser publicados en la aplicación. Está compuesto de dos submodelos.

***Modelo de composición:*** Muestra cuales son las páginas que pueden ser accedidas y las unidades de contenido que componen cada página.

***Modelo de Navegación:*** Muestra como las páginas y las unidades de contenido de cada páginas son accedidas o están linkeadas.

***Generar el modelo de presentación:*** Expresa mediante sintaxis XML, la apariencia de las páginas y los objetos incluidos en cada página, independientemente del dispositivo de salida. Se pueden generar páginas específicas o genéricas.

***Generar el modelo de personalización:*** En el modelo de estructura los usuarios y grupos han sido modelados como entidades llamadas Usuarios y Grupos, las cuales tiene características almacenadas que pueden ser usadas para guardar grupos específicos o contenidos individuales. Mediante sentencias declarativas con OQL se pueden agregar al modelo de estructura los contenidos basados en los perfiles de usuarios y grupos.



### ***Que tipos de Requerimientos No Funcionales identifica?***

- Usabilidad - Relevamiento - Diseño
- Seguridad - Relevamiento - Diseño
- Performance - Relevamiento
- Disponibilidad - Relevamiento
- Escalabilidad - Relevamiento
- Mantenibilidad - Relevamiento

WebML propone utilizar sentencias declarativas con OQL para agregar al modelo de estructura los contenidos basados en los perfiles de usuarios y grupos.

### ***¿Qué técnicas de análisis y administración de requerimientos emplea?***

#### ***Requerimientos Funcionales***

- Elicitación/captura de requerimientos: Entrevistas, análisis de documentación y checklists de captura de requerimientos.
- Especificación/ Definición de requerimientos: Utiliza Casos de uso de análisis, Diagramas de Actividad para expresar workflow de casos de uso complejos. Diccionario de datos.

Además menciona:

- Lista de grupos de usuarios con sus derechos de acceso.
- Diccionario de datos de los objetos más relevantes del dominio de la aplicación.
- Una especificación informal de las vistas del sitio.
- Un conjunto de guías de presentación que dan indicaciones para el look and feel de la interface a ser desarrollada.

#### ***Requerimientos No Funcionales***

- Elicitación/captura de requerimientos: No especifica
- Especificación/ Definición de requerimientos: Lista de Requerimientos No Funcionales.

- Validación: Test de aceptación para validar Requerimientos No Funcionales tales como Performance, Disponibilidad, Escalabilidad, Mantenibilidad y Seguridad.

**Administración de requerimientos:** WebML esta soportado por dos herramientas CASE TotiiSoft [23] y WebRatio [24].

*TotiiSoft*

- Cubre las actividades del ciclo de vida de desarrollo de aplicaciones web.
- Esta centrado en el uso de WebML.
- Tiene un módulo de diseño para edición de especificaciones de estructura, hipertextos y de modelos de personalización.
- Módulo de Manager para la administración y evolución de la aplicación en el tiempo.

*WebRatio*

- Entorno de desarrollo dirigido por modelos
- Para el desarrollo de aplicaciones complejas Web Java basadas en lenguajes BPMN y WebML.
- Desarrollo DDM de aplicaciones personalizadas BPM/BAM basado en BPMN y WebML
- Desarrollo DDM de aplicaciones personalizadas Web/SOA basado en WebML

#### **4.2.5 NTD: Navigational Development Techniques**

Propuesto en el año 2004, por Escalona [27] y Escalona, Torres, Mejias, [25][26], como sus autores lo indican, es una metodología centrada en la primera fase del ciclo de vida, su proceso de desarrollo se enfoca en la ingeniería de requerimientos, para lo cual presenta las actividades y técnicas para elicitar, analizar, especificar y validar los requerimientos de aplicaciones Web.

### *¿Cuáles son las características relevantes?*

- El proceso no contempla las fases de implementación o mantenimiento del ciclo de vida de desarrollo de aplicaciones Web, está centrado en la fase de definición de requerimientos.
- El proceso de definición de requerimientos se basa los diferentes roles de usuarios que pueden aparecen en el sistema.
- Es un enfoque orientado al *proceso* debido a que para cada actividad describe concretamente: los pasos a seguir para obtener los requerimientos y en base a ellos generar los modelos.
- Es un enfoque orientado a la *técnica*, debido a que describe las técnicas que se deben aplicar, la forma de aplicarlas y los resultados que debe ser obtenidos.
- Es un enfoque orientado al *resultado*, puesto que NDT describe los modelos de estructura y nomenclatura para los resultados obtenidos tras la aplicación de cada técnica y los resultados finales del proceso que son: la especificación de requerimientos, la especificación de modelos de análisis y los prototipos de navegación.
- Propone la validación de las especificaciones y modelos conceptuales obtenidos.
- Ofrece una guía sistemática para el tratamiento de la navegación.
- NDT no propone lenguajes de modelado, sino que utiliza lenguajes de modelado estándares como UML y UWE definido en otro enfoque metodológico.
- Define un proceso iterativo para generar la especificación de requerimientos y especificación de modelos de análisis, dado que estos modelos son estándares, pueden ser tomados por otra metodología basada en esos estándares y continuar el ciclo de vida de desarrollo.
- Propone 2 fases: Captura de Requerimientos la cual se divide en 7 subfases y por otro lado la fase de Análisis que comprende 3 subfases.
- Es soportado por una herramienta CASE NT-Tool [25][27].

La figura 4.6 muestra un diagrama de las fases y subfases de NDT.

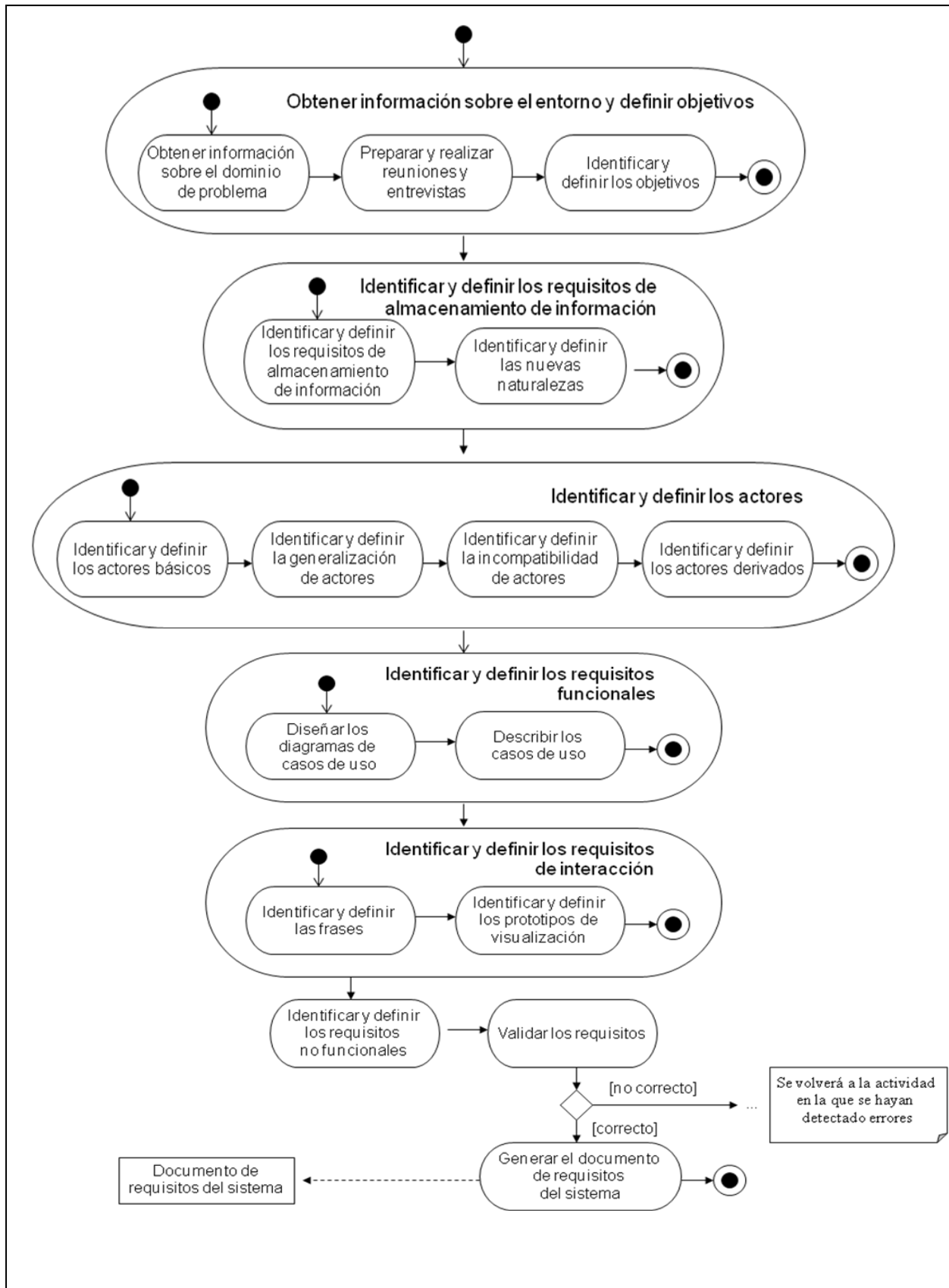


FIGURA 4.6: Diagrama de Proceso de NDT

A continuación se describen brevemente las fases y subfases que propone NDT.

***Fase de Captura de requerimientos:*** El objetivo de esta fase es obtener como resultado el *Documento de especificación de requerimientos*. NDT clasifica a los requerimientos en: de Almacenamiento de información, de actores, de interacción, funcionales y no funcionales.

El proceso comienza definiendo los objetivos del sistema para lo cual plantea las siguientes actividades:

- Obtener información sobre el entorno y definir objetivos: Las actividades principales consisten en:
  - Obtener información sobre el dominio de problema.
  - Preparar y realizar reuniones y entrevistas.
  - Identificar y definir los objetivos.

Las técnicas propuestas son entrevistas, brainstorming, cuestionarios, concept mapping, patrón para la definición de objetivos.

- Identificar y definir los requerimientos de almacenamiento de información: se propone realizar las actividades de
  - Identificar y definir los requerimientos de almacenamiento de información
  - Identificar y definir las nuevas naturalezas

Las técnicas propuestas son técnicas de estudios de sistemas anteriores, patrón para la definición de requerimientos de información, patrón para la definición de las nuevas naturalezas.

- Identificar y definir los actores: Consiste en las siguientes actividades
  - Identificar y definir los actores básicos
  - Identificar y definir la generalización de actores
  - Identificar y definir la incompatibilidad de actores
  - Identificar y definir los actores derivados

Las técnicas propuestas para cada actividad son: Patrón para la definición de actores básicos, diagrama de representación de actores generalizados, matriz

para la definición de incompatibilidad de actores, matriz para la definición de actores derivados.

- Identificar y definir los requerimientos funcionales: en base a las siguientes actividades,
  - Diseñar los diagramas de casos de uso
  - Describir los casos de uso

Las técnicas propuestas son: Diagramas de casos de uso y un patrón para la definición de requerimientos funcionales.

- Identificar y definir los requerimientos de interacción: las actividades que se realizan son
  - Identificar y definir las frases
  - Identificar y definir los prototipos de visualización

Las técnicas propuestas son: BNL (Bounded natural language), un patrón para definición de frases y un patrón para la definición de prototipos de visualización.

- Identificar y definir los Requerimientos No Funcionales: Se enumeran todas las necesidades que no están involucradas en ninguna de las categorías anteriores. NDT menciona que no es posible detectar todos los Requerimientos No Funcionales del sistema, debido a la diversidad que existe, solamente propone un patrón para describirlos. Dentro de los Requerimientos No Funcionales NDT identifica a los requerimientos de comunicación, de fiabilidad, de entorno de desarrollo, de portabilidad. Como técnica propone un patrón para describir Requerimientos No Funcionales.

- Validar los requerimientos: una vez que se identificado y definido los requerimientos, se hace necesaria su validación.

Las técnicas propuestas para esta fase son: revisiones, auditorias, glosarios, tesauros y ontologías y matriz de trazabilidad.

Si en la validación se detectaron errores, se volverá a la actividad en la que se hayan detectado los mismos para corregirlos.

- Generar el documento de requerimientos del sistema: NDT propone la estructura que debe tener este documento; el cual es la base para la fase de análisis.

***Fase de Captura de Análisis:*** El objetivo de esta fase es obtener como resultado el documento de Análisis del Sistema y los Prototipos evaluables, para ello plantea las siguientes actividades:

- Realizar el Modelo Conceptual: la cual consiste en
  - Realizar el Modelo Conceptual Básico
  - Realizar el Modelo Conceptual Final

Las técnicas propuestas son diagrama de clases, patrón para la definición de clases, patrón para la definición de asociaciones, patrón para la definición de paquetes, proceso de generación de modelo básico, proceso de revisión de modelo básico.

- Realizar el Modelo de Navegación: esta actividad tiene dos tareas
  - Realizar el Modelo de Navegación Básico
  - Realizar el Modelo de Navegación Final

Las técnicas propuestas son diagrama de navegación de UWE, patrón para la definición de nodos, enlaces, menús, índices, queries, proceso de generación de modelo básico, proceso de revisión de modelo básico algoritmos de Kruskal, y algoritmo de Warshall.

- Realizar y validar el conjunto de prototipos: que consiste en
  - Realizar el Modelo de Prototipos básicos:
  - Realizar el Modelo de Prototipos Final

Las técnicas son procesos de generación de prototipos, revisión y auditorías.

- Generar el documento de análisis del sistema: NDT propone la estructura que debe tener este documento; la cual es similar a la del documento de requerimientos. En este documento se incluyen y describen los modelos generados durante la fase de análisis mencionados anteriormente.

En caso de detectarse errores en los modelos, se volverá a la actividad en la que se hayan detectado los mismos para corregirlos.

Es importante destacar que los documentos de especificación de requerimientos, de modelo de análisis y los prototipos obtenidos por NDT, sirven de base para el diseño del sistema.

### ***¿Qué tipos de Requerimientos No Funcionales identifica?***

- Usabilidad - Relevamiento
- Fiabilidad - Relevamiento
- Portabilidad - Relevamiento
- Entorno de desarrollo - Relevamiento
- Comunicación - Relevamiento

NDT menciona que no es posible detectar todos los Requerimientos No Funcionales del sistema, debido a la diversidad que existe.

### ***¿Qué técnicas de análisis y administración de requerimientos emplea?***

#### ***Requerimientos Funcionales***

- Elicitación/Captura: Las técnicas propuestas son entrevistas, brainstorming, cuestionarios, concept mapping, patrón para la definición de objetivos.
- Especificación/Definición: utiliza técnicas estándares conocidas para especificar los diferentes tipos de requerimientos tales como:



- Estudios de sistemas anteriores, patrón para la definición de requerimientos de información,
  - Patrón para la definición de actores básicos, Diagramas de casos de uso y un patrón para la definición de requerimientos funcionales, BNL (Bounded natural language), diagrama de clases, diagrama de navegación de UWE, algoritmos de Kruskal y algoritmo de Warshall.
  - Y técnicas propuestas por NDT tales como: Patrón para la definición de las nuevas naturalezas, diagrama de representación de actores generalizados, matriz para la definición de incompatibilidad de actores, matriz para la definición de actores derivados, patrón para definición de frases, patrón para la definición de prototipos de visualización, patrón para la definición de clases, asociaciones y paquetes, patrón para la definición de nodos, enlaces, menús, índices y queries, proceso de generación de modelo básico conceptual, proceso de revisión de modelo básico conceptual, proceso de revisión de modelo básico navegacional, proceso de generación de modelo básico navegacional, procesos de generación de prototipos.
- Validación: Las técnicas propuestas para esta fase son: revisiones, auditorias, glosarios, tesauros y ontologías y matriz de trazabilidad.

### ***Requerimientos No Funcionales***

- Elicitación/captura de requerimientos: No especifica.
- Especificación/ Definición de requerimientos: Patrón para describir los Requerimientos No Funcionales.
- Validación: Revisiones, Auditorias, Matriz de trazabilidad, Prototipos.

***Administración de requerimientos:*** NDT está soportado por la herramienta NDT-Tool [25] [27] la cual:

- Sirve de soporte en la aplicación de las técnicas de captura o elicitación y especificación de requerimientos propuestas por NDT.

- Automatiza los procedimientos para generar modelos básicos conceptual y de navegación propuestos por NDT.
- Permite generar automáticamente el documento de especificación de requerimientos, de análisis y los prototipos.
- Controla las inconsistencias y errores producidos en la implementación de los patrones.
- Utiliza rational rose para visualizar los modelos (conceptual y de navegación) generados.
- Permite administrar varios proyectos en forma paralela.
- Permite a varios desarrolladores trabajar en forma concurrente.

A diferencia de otros enfoques metodológicos como UWE y WebML que tienen herramientas que soportan el proceso. NDT tool es una herramienta que soporta la primera fase del ciclo de vida y ayuda a elaborar los modelos y resultados de la aplicación de ingeniería de requerimientos propuestos por NDT.

#### **4.2.6 DDDP: Design-Driven Requirements Elicitation**

Propuesto por Lowe [28], Lowe, D., Eklund, J. [29], la denominación DDDP fue realizada por Escalona en [1].

Este proceso, propone un modelo iterativo para el desarrollo de aplicaciones Web que incorpora el uso de prototipos de diseño parcial como un elemento crucial para identificar, entender las necesidades del cliente y conducir a la formulación de los requerimientos.

##### *¿Cuáles son las características relevantes?*

- El proceso de DDDP es derivado de un resultado de un análisis de best practices en el desarrollo de aplicaciones comerciales para la web.

- La propuesta para el tratamiento de requerimientos es parte del proceso design-Driven que consiste en realizar la captura, la definición y la validación de requerimientos durante el proceso de diseño.
- Se enfoca en establecer el cómo y cuando los requerimientos son identificados.
- El proceso iterativo incorpora al cliente-desarrollador explorando en forma conjunta los diseños parciales, como parte del desarrollo de la especificación del sistema.
- Está compuesto por 2 fases o ciclos.

La figura 4.7 muestra un diagrama de las fases de DDDP.

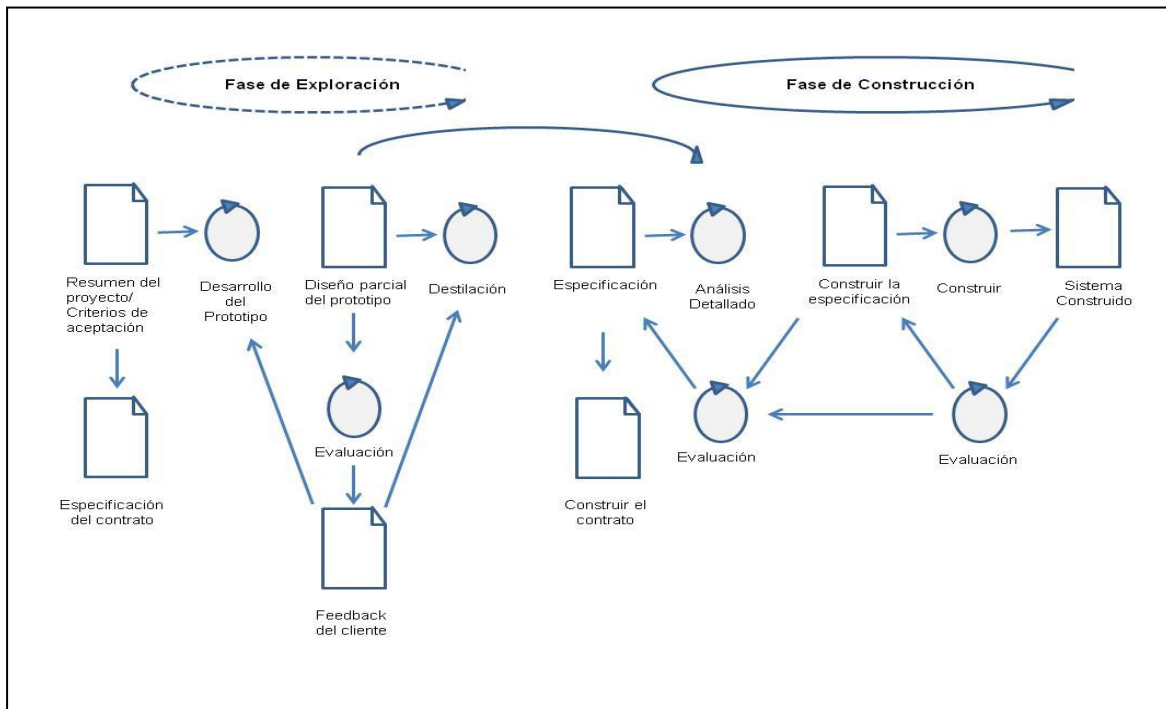


FIGURA 4.7: Diagrama de Proceso de DDDP

A continuación se describen brevemente las fases que propone DDRE o DDDP.

**Fase de Exploración:** Esta fase parte de un resumen de las necesidades del proyecto y una especificación de contrato, en base a los cuales, los desarrolladores diseñan prototipos

parciales, luego exploran y evalúan los mismos con el cliente a fin de obtener feedback e ir generando la especificación de requerimientos en forma iterativa.

Este proceso identifica requerimientos de contenido, protocolos de interface, estructura de navegación, look and feel, representación interna de datos, versiones, control de cambio, seguridad, eficiencia, adaptación del sistema, performance.

Este enfoque no define el template para los artefactos resultantes de cada etapa. Una vez que se obtuvo la especificación de requerimientos se realiza un contrato.

***Fase de Construcción:*** Esta fase comienza después que se obtuvo el contrato. Toma como entrada, el/ los diseños de prototipos parciales y la especificación de requerimientos generados en la fase anterior y plantea actividades para construir la aplicación en forma iterativa, refinando los diseños de prototipos, codificando, evaluando los resultados con el cliente a fin de actualizar la especificación de requerimientos y construir la aplicación en forma incremental.

Este enfoque plantea que muchos requerimientos son expresados como parte del diseño. Lo cual permite que ciertas características o necesidades del cliente que surgen al inicio del proceso, tales como, look and feel, estructura de contenido sean realizadas y exploradas a nivel de arquitectura, antes de que las funcionalidades del sistema sean finalizadas.

Las restricciones de performance, protocolos de interface, decisiones sobre la plataforma técnica, que con frecuencia aparecen al momento de construir la aplicación, pueden ser analizadas utilizando diseños de prototipos que ayudan al cliente a explorar posibles soluciones y de esa forma, formular los requerimientos.

### ***¿Qué tipos de Requerimientos No Funcionales identifica?***

- Estructura de navegación – Relevamiento.
- look and feel - Relevamiento.
- Representación interna de datos: Relevamiento.
- Seguridad - Relevamiento.
- Versiones, control de cambio: Implementación

- Eficiencia: Implementación
- Adaptación del sistema: Implementación
- Protocolos de interface: Implementación
- Performance: Implementación

*¿Qué técnicas de análisis y administración de requerimientos emplea?*

***Requerimientos Funcionales***

- Elicitación/Captura: Entrevistas.
- Especificación/ Definición: No especifica.
- Validación: Prototipos.

***Requerimientos No Funcionales***

- Elicitación/Captura: No especifica
- Especificación/ Definición: No especifica.
- Validación, propone prototipos para validar restricciones durante la etapa de diseño.

***Administración de requerimientos:*** No menciona técnicas o herramientas.

## Capítulo: 5

---

### 5. Tratamiento de los RNF en las metodologías de desarrollo Web.

En este capítulo se presentan los resultados del estudio comparativo de las metodologías de desarrollo de aplicaciones web con tres puntos de vista:

- Qué tipos de Requerimientos No Funcionales identifica cada enfoque.
- Qué técnicas para tratamiento de los Requerimientos No Funcionales utiliza cada enfoque.
- En qué fase del ciclo de vida se identifican los Requerimientos No Funcionales en cada enfoque.

En las secciones, 5.1, 5.2 y 5.3 se desarrollan cada uno de estos puntos.

#### 5.1 Tipos de Requerimientos No Funcionales en cada metodología

En el capítulo 2 se presentaron las bases conceptuales a ser aplicadas en este trabajo. A partir de lo expuesto en Cysneiros [7] se considera que los Requerimientos No Funcionales son *requerimientos de calidad* y son *restricciones*. Esta sección presenta los resultados de la aplicación de estos conceptos en la revisión y análisis de los enfoques metodológicos de desarrollo de aplicaciones web descritos en el capítulo 4, a fin de determinar los tipos de Requerimientos No Funcionales que son identificados y tratados en cada enfoque. En las tablas resultantes los enfoques metodológicos siguen un orden cronológico, que permite ver la evolución e importancia que fueron adquiriendo los diferentes tipos de Requerimientos No Funcionales en el entorno de la Web.

Finalmente se presenta la evolución de la tabla 4.1 presentada por Escalona y Koch en [1], citada en el capítulo 4, a fin de realizar una extensión de la columna “Req. No funcionales” con los conceptos expuestos en este trabajo.

***Los Requerimientos No Funcionales como requerimientos de calidad:***

En el análisis de las metodologías surgieron atributos de calidad adicionales a las características del estándar ISO/IEC 9126 [34], se decidió incluirlos a efectos de establecer todos los atributos que son considerados por cada enfoque. La tabla 5.1 ilustra el resultado de este análisis, presenta las seis características de calidad del estándar ISO/IEC 9126 y otras características, mostrando cuales de ellas son consideradas por cada proceso metodológico.

	Atributos de calidad ISO/IEC 9126-1						Otros				
	Usabilidad	Performance (1)	Seguridad (2)	Confidencialidad	Mantenibilidad	Portabilidad	Adaptabilidad (3)	Reutilización	Fiabilidad	Disponibilidad	Escalabilidad
WSDM	✓		✓		✓						
HFBM	✓	✓						✓			
UWE	✓		✓								
WebML	✓	✓	✓		✓					✓	✓
NDT	✓					✓			✓		
DDDP	✓	✓	✓				✓				

(1) Subcaracterística de la Eficiencia

(2) Subcaracterística de la Funcionalidad

(3) Subcaracterística de la Portabilidad

TABLA 5.1: Requerimientos No Funcionales de calidad y las metodologías Web.

A partir de la tabla 5.1 se pueden extraer las siguientes conclusiones:

- WebML. Es en cierta forma el enfoque más preocupado por *requerimientos de calidad*, considera cuatro características del estándar ISO/IEC 9126 y dos características adicionales. En contraposición UWE es el que menos atributos de calidad considera en su proceso.

Después de WebML, es DDDP el segundo interesado en los requerimientos de calidad, pero solo se limita a las características de modelo ISO/IEC 9126.

- La *usabilidad* es la característica del modelo ISO/IEC 9126 considerada por todos procesos.
- La *confiabilidad* no es tratada por ninguna de las metodologías estudiadas.
- La *usabilidad*, *performance* y *seguridad*, son las características más relevantes.
- La *mantenibilidad* y la *portabilidad* son consideradas por dos enfoques cada una.
- La *portabilidad* es relevante para los enfoques más recientes, de acuerdo al orden cronológico de presentación.
- La *reutilización*, la *fiabilidad*, la *disponibilidad*, la *escalabilidad* son las características no mencionadas explícitamente en el estándar ISO/IEC 9126; es importante destacarlas para su futuro análisis.

#### ***Los Requerimientos No Funcionales como restricciones:***

Esta sección está enfocada principalmente en las *restricciones del proceso de desarrollo* que fueron propuestas y definidas en el capítulo 2 tales como: *Alcance del proceso* (Parcial, Completo) y *Nivel de detalle del proceso* (Parcial o completo). La importancia de considerarlas radica en que un proceso de desarrollo incompleto, tendrá impacto directo en el tratamiento de los Requerimientos No Funcionales, debido a que carece de actividades, técnicas o productos para capturarlos, documentarlos y validarlos. Por consiguiente impactará en la calidad de las aplicaciones web que se desarrollen usando alguno de ellos.

También se identificaron y se agruparon otras restricciones a fin de obtener la visión global de las todas restricciones tratadas por cada enfoque. La tabla 5.2, presenta el resumen de este análisis.



	Restricciones										
	Restricciones de proceso de desarrollo				Otras						
	Alcance del proceso		Detalle de proceso		Lenguajes		Técnicas				
	Parcial	Completo	Parcial	Completo	Lenguajes de modelado	Lenguaje de implementación	Representación interna de datos	Control de versiones,	Protocolos de interface	Comunicación	De ambiente de desarrollo
WSDM		✓	✓			✓					
HFPM		✓	✓		✓						
UWE		✓	✓		✓						
WebML		✓		✓	✓	✓					
NDT	✓			✓	✓					✓	✓
DDDP		✓	✓		✓	✓	✓	✓	✓		

TABLA 5.2: Requerimientos No Funcionales – Restricciones y las metodologías Web.

Del análisis de la tabla 5.2 se pueden extraer las siguientes conclusiones

- Las *restricciones de proceso* aplicadas en las metodologías estudiadas permiten observar que:
  - WebML ofrece un enfoque que cubre todas las fases del ciclo de vida de desarrollo y además el brinda un nivel de detalle completo para cada fase, propone las actividades, técnicas y ofrece detalles de los productos a producir en cada fase.
  - WSDM, HFPM, UWE, DDDP si bien presentan un proceso de desarrollo completo, el nivel de definición de cada proceso es parcial, esto implica que carecen de lineamientos específicos de las actividades o la aplicación de técnicas, o el detalle de los productos esperados como resultado de cada fase. A primera vista estos enfoques demandan la formación de un equipo de trabajo con mucha experiencia en el desarrollo de aplicaciones web para su aplicación.
  - NDT tiene un proceso de desarrollo parcial. Aunque el nivel de detalle de la definición del proceso es completa porque detalla las actividades, las técnicas y ofrece detalles de los productos a producir en las fases que propone. El proceso está acotado solamente a las fases de relevamiento y análisis de requerimientos.

- DDDP el proceso plantea la captura, definición y validación de requerimientos durante el proceso de diseño. Propone analizar las restricciones mediante prototipos; el nivel de detalle del proceso es parcial, menciona las actividades para cada fase, no especifica las técnicas, ni ofrece detalles de los productos a generar en cada fase.

Las otras restricciones encontradas en el análisis de las metodologías fueron agrupadas como: restricciones de *lenguaje* (son las *restricciones de lenguaje de Modelado* y las *restricciones de lenguajes de implementación*), y restricciones *técnicas* (estas comprenden a *Restricciones de representación interna de datos, control de versiones, protocolos de interface, comunicación y las restricciones de ambiente de desarrollo*).

- *Restricciones de Lenguaje de modelado:*
  - WSDM, es el único enfoque que no menciona restricciones de lenguaje de modelado.
  - HFPM, demanda el conocimiento adicional de otros enfoques OOHDM [11], EORM [12] y RMM [13] para generar modelos de navegación.
  - UWE, requiere de conocimiento de enfoques UML profile y paradigma de orientación a objetos.
  - WebML, demanda conocimiento de lenguajes de modelado UML para generar el modelo estructural y XML para mostrar la apariencia de las páginas y los objetos incluidos en cada página.
  - NDT, requiere conocimiento de lenguajes de modelado UML y los diagramas de navegación de UWE.
- *Restricciones de Lenguaje de implementación:* WSDM, WebML y DDDP mencionan restricciones de este tipo:
  - En WSDM el modelo de implementación, puede depender de las limitaciones del lenguaje de programación.
  - WebML propone utilizar sentencias declarativas con OQL para agregar al modelo de estructura los contenidos basados en los perfiles de usuarios y grupos.

- DDDP, demanda conocimiento de lenguajes de programación que permitan la generación de prototipos desde la fase de exploración.
- *Las restricciones de representación interna de datos, control de versiones, protocolos de interface*, son detectadas solo por DDDP.
- *Comunicación y las restricciones de ambiente de desarrollo* son detectadas únicamente por NDT.

Hay dos puntos de vistas significativos que se desprenden de las restricciones de proceso de desarrollo analizadas en la tabla 5.2:

- El primero está relacionado al nivel de detalle de especificación parcial, de un proceso de desarrollo completo: *Un proceso de desarrollo que contempla todas las fases del ciclo de vida, pero no brinda un nivel de detalle completo en sus fases, dará lugar a una forma de trabajo no sistemática. La calidad de los productos intermedios y la aplicación web resultante estarán ligados a la idoneidad y experiencia del ingeniero de requerimientos.*
- El segundo, está vinculado al nivel de detalle de especificación completa de un proceso de desarrollo de alcance parcial: *Un proceso de desarrollo cuyo nivel de especificación de proceso es completa solo para las primeras fases del ciclo de vida, permitirá, en cierta forma, dar garantía de la calidad de los productos intermedios, pero no dará visibilidad de la calidad la aplicación web resultante, ya que las fases del ciclo de desarrollo ausentes estarán ligadas a la experiencia del equipo de trabajo.*

En ambos casos, se puede ver que contar con un proceso de desarrollo incompleto o con un nivel de detalle de especificación parcial, impactará directamente en la calidad de los productos intermedios y en la aplicación web resultante. Por consiguiente un proceso que no contempla la identificación de Requerimientos No Funcionales es incompleto y un proceso que si identifica Requerimientos No Funcionales a lo largo de su ciclo de vida, pero no ofrece actividades, técnicas, herramientas para su tratamiento, también lo es.

***Extensión de la tabla de Escalona y Koch con los diferentes tipos de Requerimientos No Funcionales***

La tabla 5.3 “Requerimientos identificados por cada metodología”, extraída del estudio realizado por Escalona y Koch, resume los tipos de requerimientos tratados por cada enfoque metodológico estudiado por los autores, en esta tabla la columna “Req. No Funcionales”, señala simplemente cual de los enfoques los identifica, sin discriminarlos.

	Req. Datos	Req. Interfaz usuario	Req. Navegacionales	Req. Personalización	Req. Transaccionales	Req. No Funcionales
WSDM	✓			✓		✓
SOHDM	✓	✓			✓	
RNA	✓	✓	✓		✓	
HFPM	✓	✓	✓			✓
OOHDM	✓	✓	✓			
UWE	✓	✓	✓	✓		✓
W2000			✓	✓	✓	
WebML	✓	✓		✓		✓
NDT	✓	✓	✓	✓	✓	✓
DDDP	✓	✓	✓	✓	✓	✓

TABLA 5.3: Requerimientos identificados por cada metodología

A continuación, partiendo del análisis realizado en este trabajo, se presentan tablas y gráficos que evolucionan la tabla 5.3 de Escalona y Koch [1], agregando los tipos de Requerimientos No Funcionales identificados y tratados en cada enfoque.

La apertura de la columna “Req. No Funcionales” en las tablas 5.4 y 5.5 se realizó considerando a los enfoques WSDM, HFPM, UWE, WebML, NDT, DDDP analizados en el capítulo 4.

En primer lugar se realizó la tabla 5.4, en la cual se agregaron los requerimientos de calidad y las restricciones de proceso, dejando a un lado requerimientos calidad y restricciones adicionales a los conceptos introducidos en el capítulo 2. El gráfico 5.1 surgió a partir de esta tabla.

La tabla 5.5 suma a la tabla anterior los *requerimientos de calidad adicionales* y *otras restricciones* detectadas a partir del análisis de los enfoques, a fin de tener una visión global de todos los requerimientos identificados y tratados por cada metodología de desarrollo estudiada. La visión gráfica de estos conceptos se muestra en el gráfico 5.2.

	Atributos de calidad							Restricciones			
	Atributos de calidad ISO/IEC 9126-1							Restricciones de proceso de desarrollo			
								Alcance del proceso		Detalle de proceso	
	Usabilidad	Performance (1)	Seguridad (2)	Confibilidad	Mantenibilidad	Portabilidad	Adaptabilidad (3)	Parcial	Completo	Parcial	Completo
WSDM	✓		✓		✓				✓	✓	
HFBM	✓	✓							✓	✓	
UWE	✓		✓						✓	✓	
WebML	✓	✓	✓		✓				✓		✓
NDT	✓					✓		✓			✓
DDDP	✓	✓	✓			✓			✓	✓	

(1) Subcaracterística de la Eficiencia  
 (2) Subcaracterística de la Funcionalidad  
 (3) Subcaracterística de la Portabilidad

TABLA 5.4: RNF identificados y tratados por cada metodología de desarrollo Web - Parcial

La tabla 5.4 muestra que:

- WebML y DDDP son los enfoques que más Requerimientos No Funcionales de calidad identifican.
- WebML a diferencia de DDDP ofrece un proceso de desarrollo completo en cuanto a alcance y el nivel de especificación del proceso.

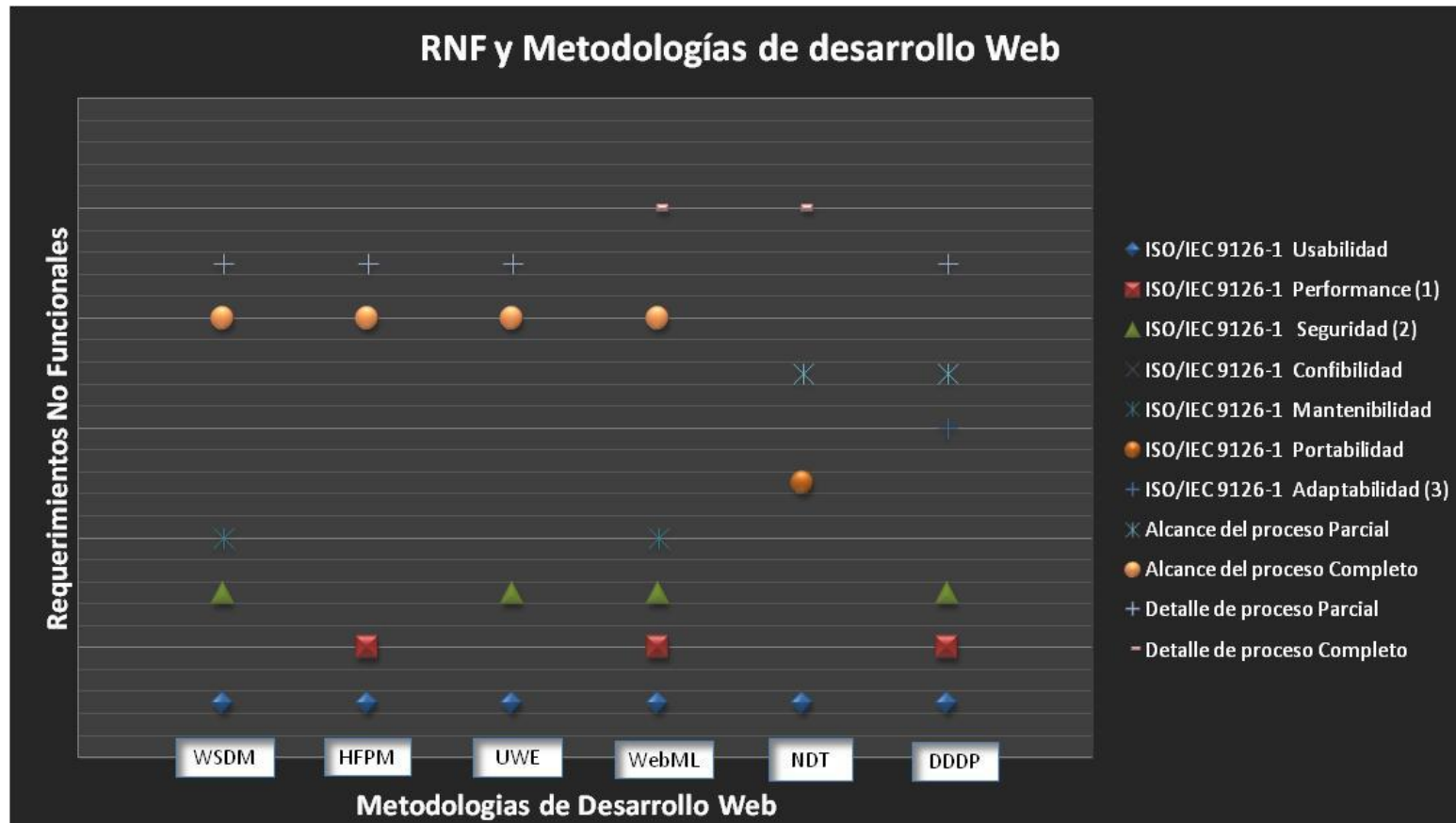


FIGURA 5.1: Requerimientos No Funcionales y Metodologías de desarrollo Web - Parcial.

	Atributos de calidad											Restricciones												
	Atributos de calidad ISO/IEC 9126-1							Otros				Restricciones de proceso de desarrollo				Otras								
	Usabilidad	Performance (1)	Seguridad (2)	Confidabilidad	Mantenibilidad	Portabilidad	Adaptabilidad (3)	Reutilización	Fiabilidad	Disponibilidad	Escalabilidad	Alcance del proceso		Detalle de proceso		Lenguajes		Técnicas						
												Parcial	Completo	Parcial	Completo	Lenguajes de modelado	Lenguaje de implementación	Representación interna de datos	Control de versiones,	Protocolos de interface	Comunicación	De ambiente de desarrollo		
WSDM	✓		✓		✓								✓		✓									
HFPM	✓	✓					✓						✓		✓		✓							
UWE	✓		✓										✓		✓		✓							
WebML	✓	✓	✓		✓				✓	✓			✓		✓	✓								
NDT	✓					✓		✓				✓		✓		✓					✓		✓	
DDDP	✓	✓	✓			✓							✓		✓		✓	✓	✓	✓	✓			

(1) Subcaracterística de la Eficiencia

(2) Subcaracterística de la Funcionalidad

(3) Subcaracterística de la Portabilidad

TABLA 5.5: RNF identificados y tratados por cada metodología de desarrollo Web.

Una mirada integradora posiciona a DDDP como el enfoque que más Requerimientos No Funcionales identifica, a pesar que ofrece un proceso de desarrollo parcial en cuanto a alcance y el nivel de especificación del proceso.

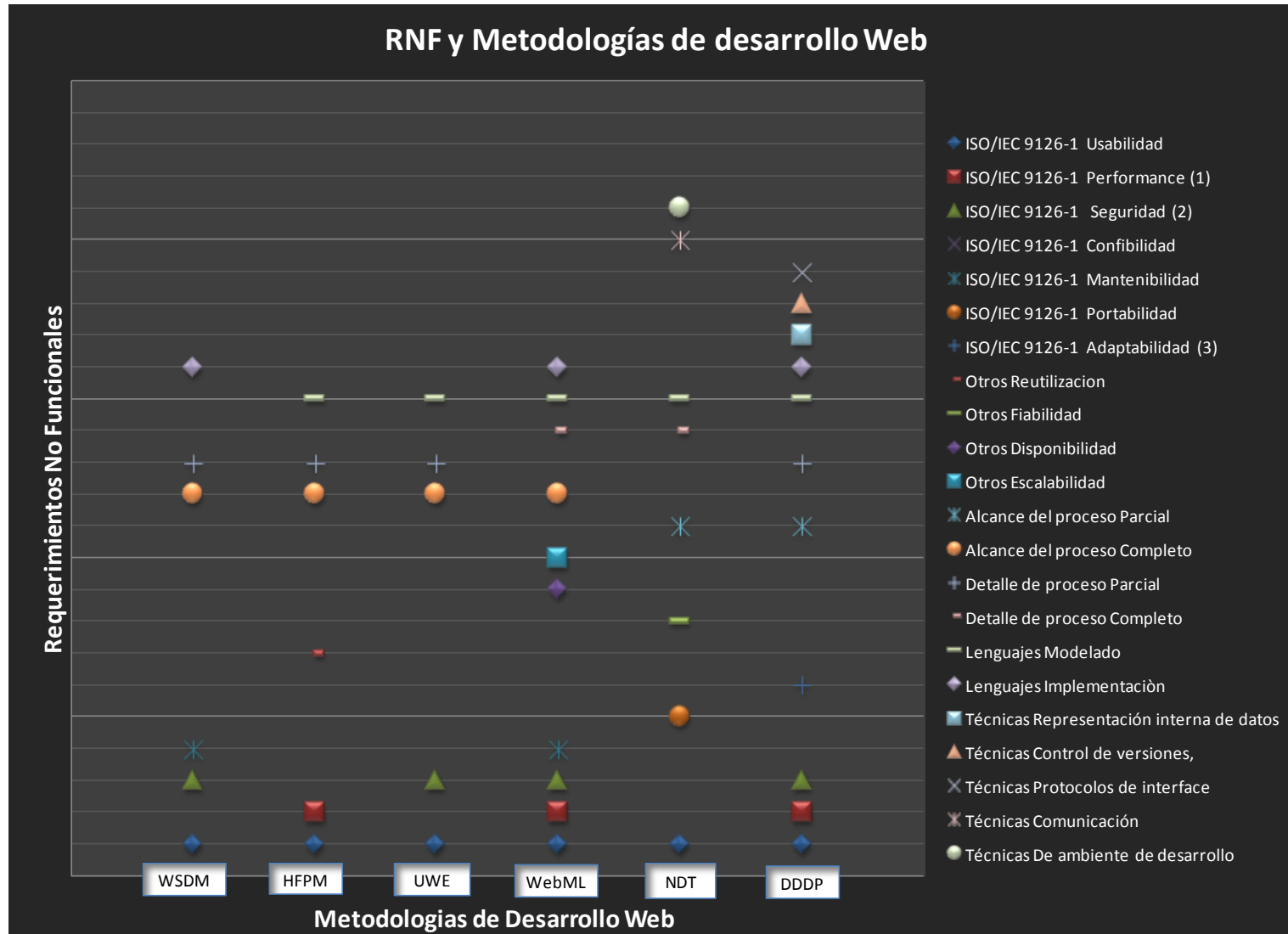


FIGURA 5.2: Requerimientos No Funcionales y Metodologías de desarrollo Web.



## 5.2 Técnicas de Ingeniería de Requerimientos para el tratamiento de RNF.

Esta sección presenta el resultado del análisis enfocado a establecer que técnicas de elicitación, especificación y validación, mencionadas en el capítulo 3, se aplican al tratamiento de los Requerimientos No Funcionales y que herramientas se proponen para su administración, de acuerdo a las herramientas presentadas en el capítulo 4.

*Elicitación:* Ninguno de los procesos de desarrollo analizados, mencionan con exactitud cual/es técnicas de elicitación es/ son utilizadas para identificar Requerimientos No Funcionales. Si bien algunos de ellos como UWE, WebML, mencionan el uso de Checklist durante la captura de requerimientos, no señalan que estos se apliquen puntualmente para capturar los Requerimientos No Funcionales.

*Especificación:* WebML y NDT hacen referencia a técnicas para la especificación de Requerimientos No Funcionales.

WebML, propone generar una “Lista de Requerimientos No Funcionales”, pero no ofrece lineamientos, ni una estructura de organización para los mismos.

NDT, ofrece “Patrón para describir los Requerimientos No Funcionales”, el cual es un template que representa un producto dentro de su proceso.

*Validación:* Cinco de los enfoques analizados consideran la validación de requerimientos como una actividad relevante dentro del marco de la ingeniería de requerimientos. Sobre los Requerimientos No Funcionales puntualmente se puede observar lo siguiente:

HFPM, propone en la primera fase de su proceso modelar los Requerimientos No Funcionales, si bien menciona el uso de prototipo, no restringe su aplicación a la fase de Requerimientos, sino que ofrece libertad para aplicarlo durante las fases de diseño e implementación.

UWE, si bien no propone lineamientos para documentar Requerimientos No Funcionales, hace alusión al modelado de Interfaz de usuario a través de prototipos, durante la fase de análisis y diseño.

WebML, propone realizar test de aceptación para validar los Requerimientos No Funcionales, detallados en una “Lista de Requerimientos No Funcionales”.

NDT, menciona el uso de Revisiones, Auditorias, Matriz de trazabilidad, Prototipos, para validar requerimientos, pero no dice puntualmente cuál de ellas es aplicada para validar los Requerimientos No Funcionales, detallados en el “Patrón para describir los Requerimientos No Funcionales” que ofrece su proceso.

DDDP, este enfoque propone analizar las restricciones mediante diseño de prototipos durante la fase de exploración de su proceso, la cual contempla actividades de identificación, análisis y diseño de necesidades.

El enfoque que no incluye a la validación como parte del proceso de Ingeniería de requerimiento es: WSDM.

*Administración de requerimientos:* Tres de los seis enfoques analizados, son soportados por herramientas CASE, que permiten entre otras actividades, la administración y trazabilidad de requerimientos.

ArgoUML, da soporte al proceso de UWE y la administración de Requerimientos. Esta herramienta presenta una limitación, no ha evolucionado a UML 2.0.

TotiiSoft y WebRatio dan soporte al proceso de WebML. TotiiSoft permite la administración de requerimientos mientras que WebRatio no.

NDT-Tool, brinda soporte y permite la administración de los requerimientos identificados con el proceso de NDT. Con lo cual, solo se limita a la fase de relevamiento y análisis.

Puesto que WebML y NDT ofrecen elementos para especificar los Requerimientos No Funcionales, las herramientas que dan soporte a estos procesos TotiiSoft y NDT-Tool, permiten la administración de Requerimientos No Funcionales.

La tabla 5.6 resume lo expuesto en esta sección.

	Técnicas			Herramienta CASE	
	Elicitación/ Captura	Especificación/ Definición	Validación	Soporta el proceso de desarrollo	Soporta la Administración de Requerimientos
WSDM					
HFPM			Prototipos		
UWE			Prototipos	ArgoUML	ArgoUML
WebML		Lista de requerimientos no funcionales	Test de aceptación	TotiiSoft WebRatio	TotiiSoft
NDT		Patrón para describir los requerimientos no funcionales	- Revisiones - Auditorias - Matriz de trazabilidad - Prototipos	NDT-Tool	NDT-Tool
DDDP			Prototipos		

TABLA 5.6: Técnicas de Ingeniería de requerimientos y herramientas CASE aplicadas a RNF

### 5.3 Ciclo de vida y los Requerimientos No Funcionales

En la sección 5.1 se presentaron los tipos de Requerimientos No Funcionales que son identificados por las metodologías WSDM, HFPM, UWE, WebML, NDT, DDDP. En esta sección el análisis de cada enfoque está orientado a establecer: (a) en qué fase del Ciclo de Vida se identifica y es tratado cada Requerimiento No Funcional (ver Tabla 5.7), y (b) para cada enfoque metodológico se analiza la influencia en el Ciclo de Vida de las restricciones de proceso (ver tabla 5.8.)

Debido a que la estructura de fases que proponen los enfoques metodológicos estudiados difieren notablemente, para poder hacer la comparación en este trabajo se contemplan las fases del ciclo de vida en forma general: Relevamiento, Análisis, Diseño, Implementación y Prueba.

	Atributos de calidad											Restricciones											
	Atributos de calidad ISO/IEC 9126-1							Otros				Restricciones de proceso de desarrollo				Otras							
	Usabilidad	Performance (1)	Seguridad (2)	Confiabilidad	Mantenibilidad	Portabilidad	Adaptabilidad (3)	Reutilización	Fiabilidad	Disponibilidad	Escalabilidad	Alcance del proceso		Detalle de proceso		Lenguajes		Técnicas					
												Parcial	Completo	Parcial	Completo	Lenguajes de modelado	Lenguaje de implementación	Representación interna de datos	Control de versiones, protocolos de interface	Comunicación	De ambiente de desarrollo		
WSDM	R/D		R/D		D/I								R	R			D/I						
HFPM	R/D	R/D/I					R						R	R		D							
UWE	R/D		R										R	R		D							
WebML	R/D	R	R/D		R				R	R			R		R	D	D/I						
NDT	R/D					R		R				R			R	D						R	R
DDDP	R/D	R/D	R/D				I						R	R		D	D/I	I	I	I			

(1) Subcaracterística de la Eficiencia R: Relevamiento  
 (2) Subcaracterística de la Funcionalidad D: Diseño  
 (3) Subcaracterística de la Portabilidad I: Implementación

TABLA 5.7: Los Requerimientos No Funcionales y el Ciclo de Vida.

La tabla 5.7 muestra en que fase del ciclo de vida de desarrollo, los procesos identifican y tratan cada tipo de requerimiento no funcional; a partir de la cual se puede decir que:

- La mayoría de los atributos de calidad, son identificados durante el relevamiento.
- La *usabilidad*, es identificada por todos los enfoques durante el relevamiento y el diseño.
- La *performance* y la *seguridad*, los procesos que las identifican lo hacen durante el relevamiento y el diseño.

- La *mantenibilidad* es identificada por dos enfoques, uno lo hace en el relevamiento y el otro durante el diseño e implementación.
- La *portabilidad* es identificada por uno de los enfoques más recientes, durante el relevamiento.
- La *adaptabilidad* (subcaracterística de la portabilidad), es identificada por un solo enfoque recién en la implementación.
- La *reutilización*, la *fiabilidad*, la *disponibilidad*, la *escalabilidad* son identificadas durante el relevamiento.
- Todas las *restricciones del proceso de desarrollo* son identificadas en la fase de relevamiento.
- Las *restricciones de lenguaje* surgen en las fases de Diseño e implementación.
- Las *restricciones técnicas de Comunicación y Ambiente de desarrollo* son identificadas por NDT en la fase de relevamiento.
- Las *restricciones técnicas de Representación interna de datos, Control de Versiones y Protocolos de Interface* son identificadas por DDDP en la fase de implementación.
- Como conclusión general de la tabla 5.7 se desprende que existen Requerimientos No Funcionales que no son identificados en la fase de relevamiento; y el hecho de que surjan tarde en el ciclo de vida, puede afectar la calidad de la aplicación web resultante.

La otra perspectiva analizada en esta sección, es cómo influyen las restricciones de proceso, presentadas en el capítulo 2 de este trabajo, en el ciclo de vida de desarrollo. La tabla 5.8 refleja la forma en la que cada metodología aborda el tratamiento de los Requerimientos No Funcionales a lo largo del ciclo de vida.

El símbolo ‘S’ se utiliza para indicar que el enfoque metodológico identifica Requerimientos No Funcionales en la fase del ciclo de vida señalada y que el proceso propone actividades, técnicas y ofrece detalles de los productos a generar para el tratamiento de los mismos.

El símbolo ‘I’ indica que a pesar de que el enfoque metodológico identifica Requerimientos No Funcionales en la fase del ciclo de vida señalada, el proceso no propone una descripción suficiente de las técnicas y productos involucrados en su tratamiento.

El símbolo ‘N’ indica que el enfoque metodológico no identifica Requerimientos No Funcionales en la fase del ciclo de vida señalada.

	Relevamiento	Análisis	Diseño	Implementación	Prueba
WSDM	I	I	I	I	N
HFPM	I	I	I	I	N
UWE	I	I	I	I	N
WebML	S	I	I	I	I
NDT	S	S	N	N	N
DDDP	I	I	I	I	N

TABLA 5.8: Influencia de las Restricciones de Proceso en el Ciclo de Vida.

Del análisis de la tabla 5.8 se pueden extraer las siguientes conclusiones:

- WebML es el único proceso que identifica y trata los Requerimientos No Funcionales a lo largo del todo el ciclo de vida de desarrollo. Aunque solamente ofrece técnicas y detalles de los productos que se deben generar para la fase de relevamiento.
- WSDM, HFPM, UWE, DDDP identifican y tratan requerimientos no funcionales durante las fases de relevamiento, análisis, diseño e implementación, pero en la fase de prueba no hacen mención al tratamiento de los mismos.
- NDT, el alcance de su proceso no cubre todas las fases del ciclo de vida de desarrollo; pero ofrece un nivel de detalle de la definición del proceso completo para las fases que cubre. NDT especifica las actividades, las técnicas y ofrece detalles de los productos a generar en las fases de relevamiento y análisis.
- Solo dos de los seis enfoques estudiados, ofrecen técnicas y brindan el detalle de los productos a generar en la fase de relevamiento para el tratamiento de los Requerimientos No Funcionales.
- Cinco de los seis enfoques estudiados no ofrecen lineamientos para las actividades de la fase de prueba, lo cual no permite establecer como realizan estos la validación de los Requerimientos No Funcionales identificados en las etapas iniciales del ciclo de vida.

## 5.4 Conclusiones del análisis comparativo

El objetivo de este capítulo fue presentar un estudio comparativo de metodologías de desarrollo web, a fin de destacar que tipos de Requerimientos No Funcionales identifica y trata cada enfoque en las diferentes fases de su proceso; para lo cual se tomo en cuenta la información de los capítulos precedentes: Las bases conceptuales establecidas en el capítulo 2, a partir de lo expuesto por Cysneiros [7], que considera que los Requerimientos No Funcionales son *requerimientos de calidad y son restricciones*. Luego el estudio realizado en el capítulo 3, donde se analizaron los procesos de desarrollo de WSDM, HFPM, UWE, WebML, NDT, DDDP en forma completa para identificar los tipos de RNF, contemplados por cada uno y cuales técnicas de la IR citadas en el capítulo 4, propone cada enfoque para su tratamiento y gestión. Además se estableció en cual/es fase/s del ciclo de vida de desarrollo de software son tratados los RNF en cada metodología estudiada.

Si bien, los resultados del estudio comparativo de cada punto de vista analizado se detallan en las secciones 5.1, 5.2 y 5.3 de este capítulo, en [40] y a continuación se describen las respuestas a los interrogantes que dieron origen a este análisis:

- 1- ¿Se contemplan los Requerimientos No Funcionales en las metodologías de aplicaciones Web?  
*Si, a pesar de la carencia de consenso en su significado.*
  
- 2- ¿Utilizan técnicas de la ingeniería de requerimientos para el tratamiento de los Requerimientos No Funcionales?,  
*En los enfoques estudiados se puede ver que no hay técnicas específicas para la elicitación de Requerimientos No Funcionales, no hay consenso sobre como especificarlos y tampoco lineamientos de como validarlos.*
  
- 3- ¿Hay herramientas que soporten a las metodologías de desarrollo web?  
*Existen herramientas, pero las mismas son particulares del proceso metodológico al que soportan.*
  
- 4- ¿Cómo es el tratamiento de los Requerimientos No Funcionales en el ciclo de vida de desarrollo?

*Los procesos de desarrollo de los enfoques estudiados no brindan técnicas que soporten el tratamiento de los Requerimientos No Funcionales a lo largo del ciclo de vida.*

*Hay carencia de consenso para establecer en que fase del ciclo de vida de desarrollo se identifican los Requerimientos No Funcionales. De hecho hay Requerimientos No Funcionales que no son identificados durante la fase de relevamiento y análisis.*

Si bien actualmente la IR proporciona numerosas técnicas y herramientas para identificar, describir, validar y gestionar requerimientos, estos no son aplicados muy a menudo por los profesionales de sistemas, particularmente en el desarrollo de aplicaciones web. La madurez del proceso de ingeniería de requerimientos parece ser insuficiente y demanda que surjan nuevos enfoques o se evolucionen de los enfoques existentes para el tratamiento de los Requerimientos No Funcionales en particular.

Por su importancia, las actividades para captura, especificación y validación de los RNF, merece contar con técnicas y lineamientos específicos.



## Capítulo: 6

---

### 6. Procesos de Elicitación y Especificación de RNF para Web.

La Ingeniería de Requerimientos (IR) cubre actividades esenciales en la Ingeniería de Software, pero cuando es aplicada a la Ingeniería Web, se enfrenta a un escenario diferente en el cual debe considerar: Stakeholders no disponibles, multidisciplinariedad, alta volatilidad de requerimientos, ambientes operacionales no predecibles, poca o escasa experiencia del equipo de desarrollo con las tecnologías web y la importancia particular de aspectos de calidad como la usabilidad, seguridad y performance.

Las características propias de las aplicaciones Web, hacen que la identificación de RNF (restricciones de arquitectura, restricciones de proceso de desarrollo y los aspectos de calidad) cobren un papel relevante en la aplicación de la IR en la Ingeniería Web. En el análisis comparativo realizado en el capítulo 5 se puede ver que ninguno de los enfoques de desarrollo de aplicaciones Web estudiados, menciona con exactitud que técnicas de elicitación utilizan para identificar RNF durante la fase de captura de requerimientos; si bien algunos de ellos como UWE, WebML, mencionan el uso de Checklist durante la captura de requerimientos, no indican que estos se apliquen puntualmente para capturar RNF.

En cuanto a la especificación de RNF solo dos enfoques mencionan artefactos para documentarlos, WebML propone generar una “Lista de Requerimientos No Funcionales”, aunque no ofrece lineamientos, ni una estructura de organización para los mismos. NDT ofrece un “Patrón para describir los Requerimientos No Funcionales”, el cual representa un producto dentro de su proceso; el patrón de NDT aporta una estructura para documentar cada RNF, según lo considera Escalona en [27], “Llegar a identificar y definir todos los RNF del sistema es una tarea bastante compleja debido a la diversidad de que existe”; con lo cual propone un patrón genérico para especificar un RNF y una plantilla para recolectar todos los RNF identificados la cual que puede completar en el documento de especificación de requerimientos del software.

La Figura 6.1 extraída de [35], es la representación grafica del proceso de IR introducido en el capítulo 3.

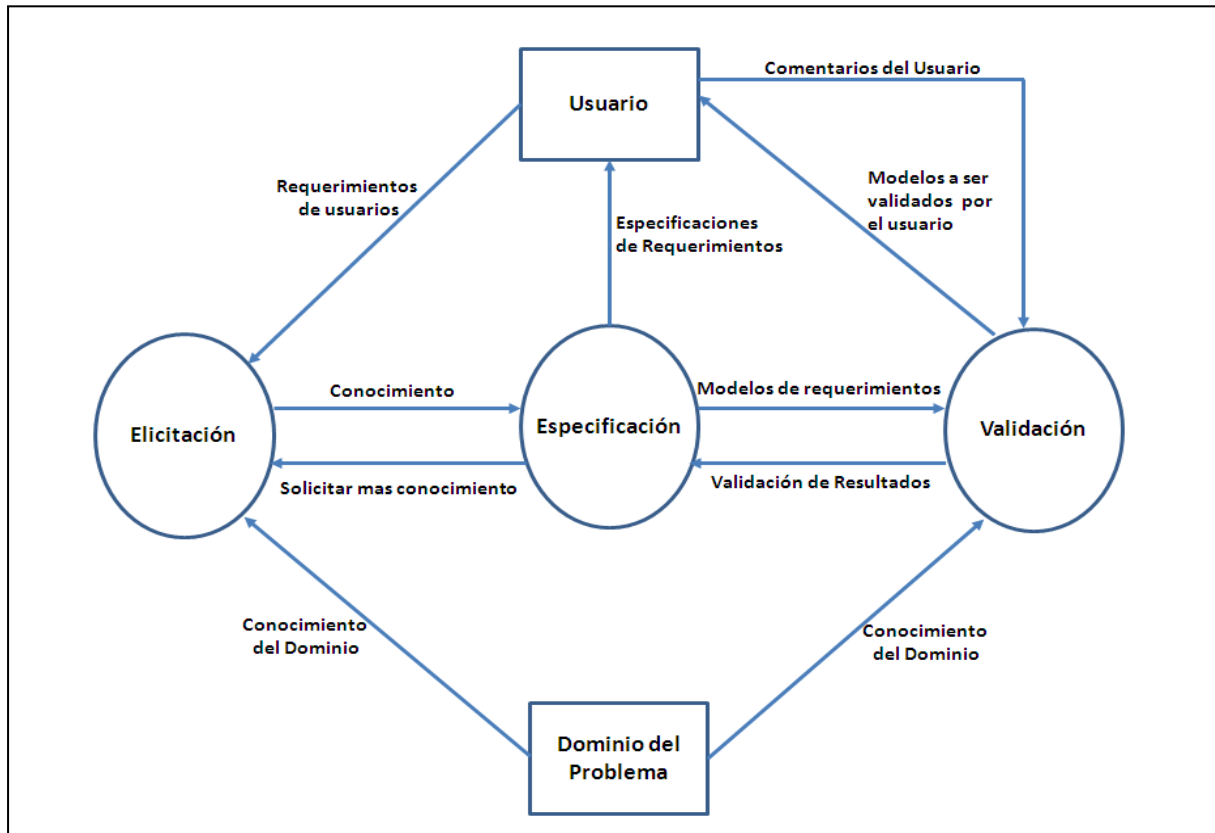


FIGURA 6.1: Framework para el Proceso de Ingeniería de Requerimientos.

En este capítulo se proponen lineamientos y técnicas enfocados a la captura y documentación de RNFs para aplicaciones Web. Se describen los procesos de Elicitación y Especificación de la IR complementándolos con actividades y herramientas particulares para adquisición del conocimiento del dominio del problema relacionado a los RNF y para la especificación de los mismos según lo ilustra la figura 6.2.

El proceso de validación de la IR está fuera del alcance de esta tesis.

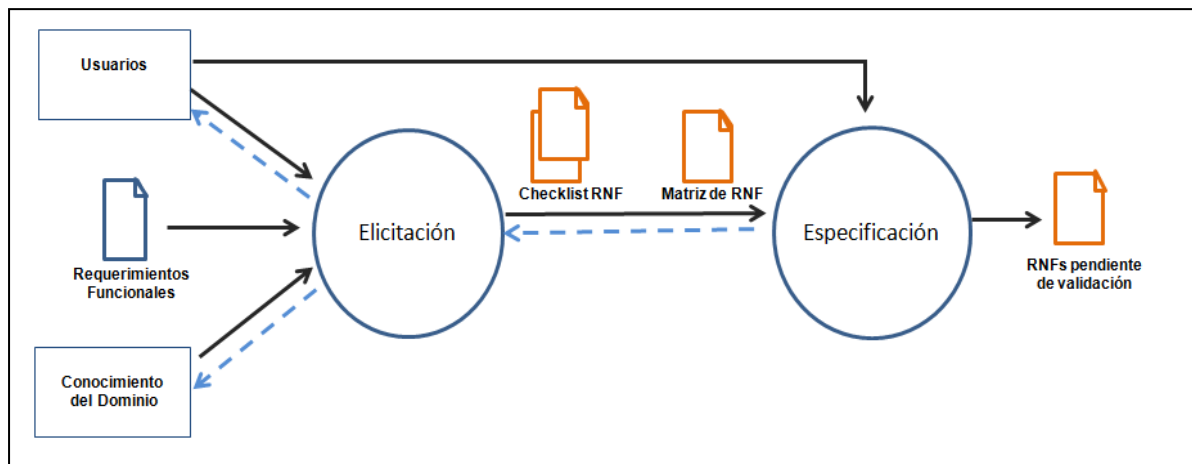


FIGURA 6.2: Proceso de Elicitación y Especificación de RNF Web.

Los procesos para la elicitación y especificación de RNF de aplicaciones Web, que propone este trabajo, se describirán siguiendo la estructura de presentación siguiente:

- **Objetivo:** Describe el propósito del proceso.
- **Entradas:** Enumera las entradas del proceso.
- **Actividades:** Describe las actividades del proceso.
- **Técnicas/Herramientas:** Menciona las técnicas y/o herramientas que se aplican en el proceso.
- **Productos:** Menciona los documentos o productos que deben ser generados en el proceso.
- **Interacción:** Menciona los procesos de la IR con los cuales se relaciona.

Antes de comenzar con la definición de cada proceso, cabe destacar que esta propuesta da mucha importancia a la IR, en el sentido que plantea realizar un gran esfuerzo para el tratamiento de los RNF Web durante la captura, análisis, definición y especificación a fin de sentar una base de RNF adecuados a las necesidades de los Stakeholders, los cuales una vez validados serán la entrada a las etapas siguientes en el ciclo de vida de desarrollo para diseñar, construir, probar una aplicación Web que además de proporcionar la funcionalidad esperada, cumpla con los requerimientos de calidad y restricciones establecidos.

## **6.1 Proceso de Elicitación de RNF Web**

Como se menciona en capítulos anteriores, en el proceso de Elicitación los RNF pueden presentarse junto a los requerimientos funcionales a través de diversas formas: políticas de negocio o externas al negocio, paradigmas de programación, tecnología de autorización, infraestructura existente, alcance del proceso de desarrollo, habilidades y conocimientos del equipo de desarrollo, interfaz con otros sistemas, atributos de calidad, entre otros. Debido a la carencia de lineamientos y técnicas específicos para poder identificarlos durante la etapa de relevamiento, los RNF en muchas ocasiones son capturados como atributos del software o reglas de negocio y se documentan en conjunto con los requerimientos funcionales, o se describen en forma de notas aisladas y el tratamiento de los mismos termina ahí.

La complejidad de los requerimientos en el desarrollo de aplicaciones Web demanda que el ingeniero de requerimientos utilice herramientas y técnicas acordes a la naturaleza de los mismos, particularmente los RNF necesitan ser identificados, clasificados, analizados para recién ser documentados apropiadamente.

A continuación se describirá el proceso de Elicitación de RNF de aplicaciones Web que ilustra gráficamente la figura 6.3. La tabla 6.1 al final de la sección resume: el objetivo, las entradas, las actividades, las técnicas/herramientas, los productos/salidas y la interacción con otros procesos de la IR, correspondientes a la Elicitación.

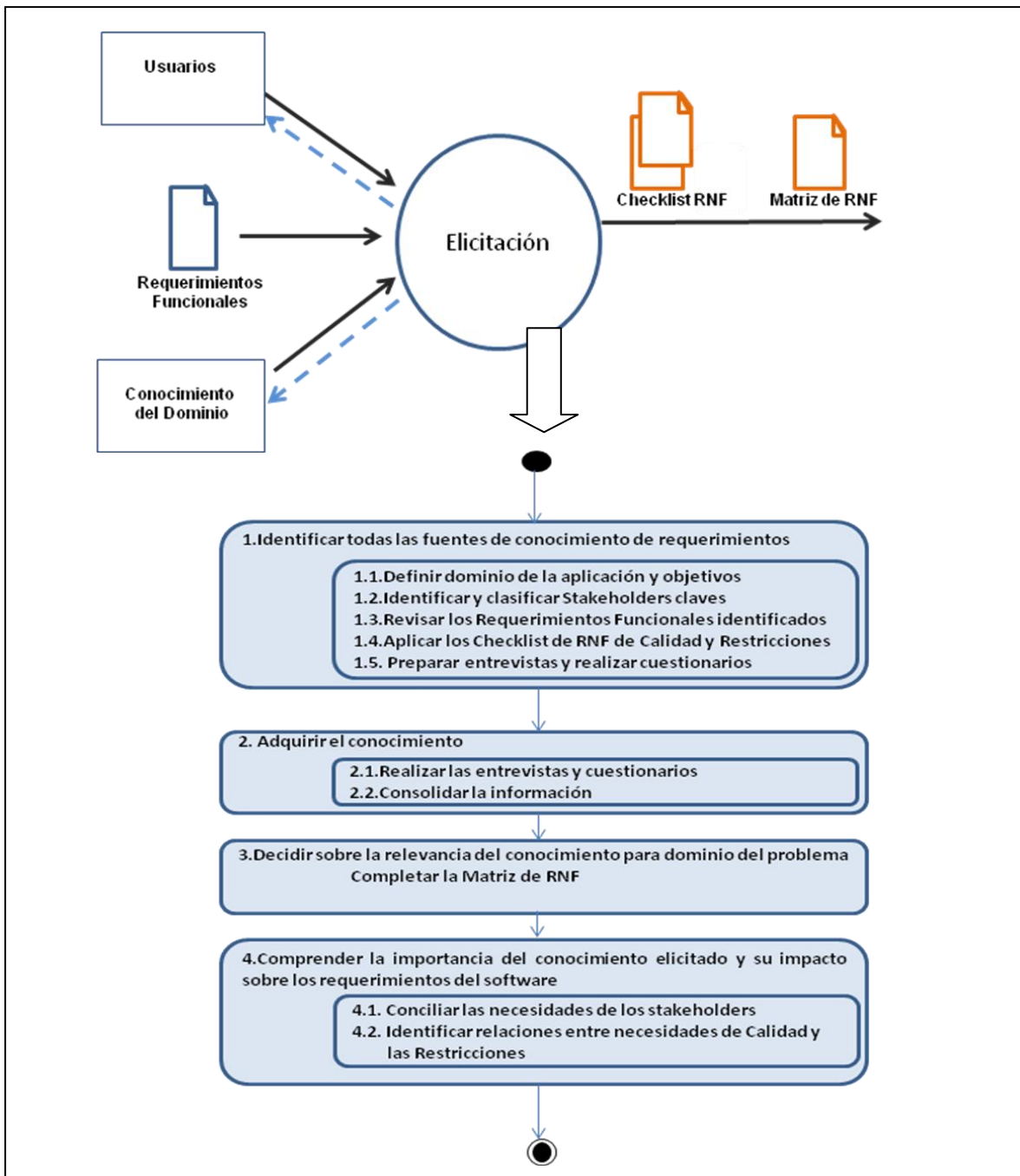


FIGURA 6.3: Proceso de Elicitación de RNF Web.

### 6.1.1 Objetivo de la Elicitación de RNF Web

El objetivo del proceso de elicitación de RNF es: adquirir el conocimiento del dominio de problema relacionado a los RNF de las aplicaciones Web.

### 6.1.2 Entradas,

Las entradas al proceso son:

- Clientes, Usuarios, Otros Stakeholders del Dominio: Los mismos se describen en el capítulo 7.
- Requerimientos Funcionales: La captura de los Requerimientos Funcionales (RF) ocurre en forma paralela durante la elicitación de requerimientos. Los RNF deberán ser mapeados a los mismos para su análisis.
- Conocimiento del Dominio: El ingeniero de requerimientos puede considerar como fuente de conocimiento a la literatura sobre el dominio, software existente en el dominio, aplicaciones de software similares en otros dominios, estándares nacionales e internacionales relevantes para la aplicación Web.

### 6.1.3 Actividades

A continuación se describen las actividades y tareas que componen el proceso en forma más detallada y se presentan las técnicas empleadas en cada una de ellas.

1. **La identificación de todas las fuentes de conocimiento de requerimientos:** Esta actividad está compuesta por las siguientes tareas
  - 1.1. *Definir dominio de la aplicación y objetivos:* El ingeniero de requerimientos debe identificar el entorno de trabajo, familiarizarse con el vocabulario a utilizar en el dominio, conocer los objetivos que se pretende con la implementación de la aplicación. Es una tarea crítica cuando el ingeniero de requerimientos no conoce el dominio. Si hay una versión anterior a la aplicación que se va a desarrollar, es aconsejable solicitar documentación relacionada a la misma.
  - 1.2. *Identificar y clasificar Stakeholders claves:* Uno de los conceptos relevantes en las aplicaciones Web, es enfrentarse a la *falta de disponibilidad de Stakeholders*, esto significa que muchos stakeholders, como por ejemplos los usuarios web potenciales, son desconocidos, por lo tanto no están disponibles durante la elicitación de requerimientos.

El ingeniero de requerimientos debe encontrar un conjunto de representantes adecuados que puedan proporcionar las necesidades reales, para lo cual se recomienda describir los usuarios potenciales y cada Stakeholder con la información que proponen en [41] Robertson and Robertson.

Para describir los usuarios potenciales se recomienda proporcionar:

- Nombre de usuario,
- Rol de usuario,
- Experiencia en el conocimiento del dominio,
- Experiencia tecnológica,
- Otras características propias de cada usuario, dentro de las características se deben describir: habilidades físicas, habilidades intelectuales, actitud hacia el trabajo, actitud hacia la tecnología, educación, conocimiento de idiomas, edad del grupo y género.

Cabe acotar que en este trabajo consideramos relevantes, las características de cada usuario, para definir RNF relacionados a la usabilidad. Su aplicación se realizará en el caso práctico.

Para describir otros Stakeholders:

- Identificación del Stakeholder (Combinación de Rol/ Título de trabajo, Nombre de la persona y nombre de la organización.
- Conocimiento necesario por el proyecto.
- Grado de influencia.
- Acuerdo sobre la manera de abordar los conflictos entre los Stakeholders que tienen interés en el mismo Conocimientos.

1.3. *Revisar los Requerimientos Funcionales identificados*: Los RF responden a la pregunta ¿Qué debe hacer la aplicación? por lo que es importante considerarlos integrados como un todo al momento de capturar el conocimiento relacionado a los RNF. Para los casos donde el proceso de elicitación es realizado por más de un ingeniero de requerimientos, se deben planificar reuniones para el debate y revisión de los RF, a fin de consolidar un modelo mental de dominio único. Por ejemplo se pueden identificar todos los

requerimientos que necesitan medición de tiempos de respuestas ya que estarán asociados a la característica *Eficiencia / Tiempo de comportamiento*.

1.4. *Aplicar los Checklist de RNF de Calidad y Restricciones*: El proceso propone dos checklist (una Plantilla de Elicitación de RNF de Calidad y una Plantilla de Elicitación de RNF restricciones); el ingeniero de requerimientos tiene que emplearlas para seleccionar la/las categoría/s de RNFs que serán consideradas de acuerdo a la aplicación Web que se está desarrollando. La subcaracterística seleccionada debe tener la marca “✓” en la plantilla caso contrario quedará en blanco. La marca “✓” indica que se realizará la captura de conocimiento relacionado a esta subcaracterística.

Para realizar esta tarea es fundamental que se hayan desarrollado las actividades precedentes; el dominio de la aplicación a desarrollar debe estar claro y se debe contar con un listado de los stakeholders involucrados. La tarea se descompone en tres instancias:

- a) *Mapear los stakeholders*: Consiste en clasificar los stakeholders involucrados con los tipos de stakeholders de las plantillas de elicitación (las cuales se describen en el capítulo 8). Para realizar esta clasificación se deben tener en cuenta los *Tipos de stakeholders* y los *Intereses a considerar* asociados a estos para cada subcaracterística/subcategoría establecidas en las plantillas de Elicitación de RNF.
- b) *Seleccionar las características / subcaracterísticas de RNF de Calidad*: El ingeniero de requerimientos debe decidir cuáles características son significativas para el dominio de la aplicación. Esta tarea es compleja, dado que está condicionada por la experiencia del ingeniero de requerimientos. En [51] se propone una clasificación de RNF relevantes para ocho dominios de aplicaciones diferentes, que puede ser tomada como referencia.
- c) *Seleccionar características / subcaracterísticas de RNF Restricciones*: Si se trata de un entorno de trabajo conocido, el ingeniero de requerimientos debe decidir cuáles características son significativas para el dominio de la aplicación. Caso contrario se recomienda considerar todas las categorías propuestas por la plantilla.



1.5. *Preparar entrevistas y realizar cuestionarios*: Según las dimensiones del proyecto, la aplicación que se está desarrollando y la cantidad de Stakeholders involucrados, la búsqueda de conocimiento relacionado a los RNF puede realizarse en conjunto con entrevistas planificadas con el objetivo de capturar conocimiento vinculado a la funcionalidad. De acuerdo a las categorías de RNF seleccionadas en el punto anterior, se pueden preparar entrevistas dirigidas a los diferentes stakeholders, para aplicar las preguntas de las plantillas de RNF. Otra alternativa es formular cuestionarios específicos dirigidos a los diferentes stakeholders, que reúnan las preguntas de las plantillas de RNF. La organización de entrevistas y cuestionarios a realizar quedan a criterio del ingeniero de requerimientos.

2. *Adquirir el conocimiento*: Consiste en las siguientes tareas.

2.1. *Realizar las entrevistas y cuestionarios*: Se desarrollan las entrevistas planificadas, se aplican los cuestionarios.

2.2. *Consolidar la información*: El ingeniero de requerimientos debe analizar y organizar la información adquirida. En caso de identificar inconsistencias, debe anotar las mismas para realizar una revisión y el seguimiento futuro de las mismas.

3. *Decidir sobre la relevancia del conocimiento para dominio del problema*: Esta actividad consiste en las siguiente tareas

*Completar la Matriz de RNF*: El ingeniero de requerimientos debe decidir si el conocimiento capturado en las actividades anteriores se adecua a las categorías de RNF seleccionadas. Si considera que el conocimiento es relevante, debe volcar la información en la Matriz de RNF que propone el proceso para dar soporte a la elicitación y especificación de RNF, la misma se describe en el capítulo 9.

En esta instancia el ingeniero de requerimientos está en condiciones de completar los siguientes campos de la matriz: *Id. RNF candidato, Tipo, Categoría/Grupo, Subcategoría/Subgrupo, Descripción, Fecha Alta, Origen, Stakeholder, Solicitante*.

En el campo descripción, deben detallarse las reglas de negocio, políticas y restricciones identificadas por ejemplo: reglas asociadas al formato de fechas, importes, reglas asociadas al tratamiento de caracteres especiales necesarios. Ej. ("/", "°", "ñ", otros), los estándares internos y externos, valores de performance, cantidades de usuarios

concurrentes, los nombres de otros sistemas identificados con los cuales debe interactuar la aplicación, nombres de navegadores, entre otros.

#### **4. Comprender la importancia del conocimiento elicitado y su impacto sobre los requerimientos del software:** Consiste en las siguientes tareas

4.1. *Conciliar las necesidades de los stakeholders:* Cuando para una misma pregunta, se obtuvieron respuestas diferentes de los Stakeholders, Por ejemplo: el formato de la fecha tiene que ser DD/MM/AAAA para un stakeholder y mientras que para otro: DD-MM-AA. Un stakeholder desea usar scroll vertical, mientras otro quiere usar paginado.

El ingeniero de requerimientos debe identificar estas diferencias y lograr un criterio uniforme con los stakeholders involucrados para volcarlo en la matriz.

4.2. *Identificar relaciones entre necesidades de Calidad y las Restricciones:* Se trata de establecer las relaciones entre las necesidades identificadas, por ejemplo *Interoperabilidad*, un stakeholder necesita que la aplicación interactúe con el sistema de CAJA. Está relacionada con las restricciones de *arquitectura*, subcaracterística *Otros sistemas*. El ingeniero de requerimientos debe completar el campo *Id. Req. Asociado* de la Matriz de RNF. Esta tarea puede resultar compleja cuando el conocimiento capturado presenta ambigüedades o existen inconsistencias no resueltas.

### **6.1.4 Técnicas/Herramientas**

Este proceso plantea el uso de checklists, cuestionarios y entrevistas.

Los Checklists: *Plantilla de Elicitación de RNF de Calidad* y *Plantilla para Elicitación de RNF Restricciones*, se proponen en este trabajo y se describen en el capítulo 8.

Los cuestionarios y entrevistas se presentaron en el capítulo 3, en la revisión de un conjunto de técnicas existentes en la IR para la elicitación de requerimientos.

### 6.1.5 Los Productos/Salidas

De acuerdo a las actividades propuestas en este proceso, el ingeniero de requerimientos debe completar: *La Plantilla de Elicitación de RNF de Calidad*, *la Plantilla para Elicitación de RNF Restricciones* y además generar la versión preliminar de la *Matriz de Requerimientos No Funcionales*, esta matriz es otro de los productos que propone este trabajo, se describe en el capítulo 9.

### 6.1.6 Interacción

Como lo ilustra la figura 6.1 el proceso de elicitación es quien brinda el material de entrada a los otros procesos de la IR. Acotado al alcance de nuestra propuesta, el proceso de elicitación de RNF Web ocurre en paralelo con el proceso de especificación de RNF Web por lo tanto tiene una estrecha interacción con este.

La tabla 6.1 muestra una síntesis del proceso de Elicitación de RNF para aplicaciones Web explicado en esta sección; enumera su objetivo, las entradas, las actividades, las técnicas/herramientas, los Productos/salidas y la interacción con otros procesos de la IR.

<b>Objetivo</b>	Adquirir el conocimiento del dominio de problema relacionado a los RNF de aplicaciones Web.
<b>Entradas</b>	<ul style="list-style-type: none"> <li>• Clientes, Usuarios, Otros Stakeholders.</li> <li>• Requerimientos Funcionales.</li> <li>• Conocimiento de Dominio.</li> </ul>
<b>Actividades</b>	<ol style="list-style-type: none"> <li>1.La identificación de todas las fuentes de conocimiento de requerimientos:             <ol style="list-style-type: none"> <li>1.1.Definir dominio de la aplicación y objetivos</li> <li>1.2.Identificar y clasificar Stakeholders claves</li> <li>1.3.Revisar los Requerimientos Funcionales identificados</li> <li>1.4.Aplicar los Checklist de RNF de Calidad y Restricciones</li> <li>1.5. Preparar entrevistas y realizar cuestionarios</li> </ol> </li> <li>2. Adquirir el conocimiento: Esta actividad propone las siguientes tareas.             <ol style="list-style-type: none"> <li>2.1.Realizar las entrevistas y cuestionarios</li> <li>2.2.Consolidar la información</li> </ol> </li> </ol>

	<p>3. Decidir sobre la relevancia del conocimiento para dominio del problema Completar la Matriz de RNF</p> <p>4. Comprender la importancia del conocimiento elicitado y su impacto sobre los requerimientos del software.</p> <p>4.1. Conciliar las necesidades de los stakeholders</p> <p>4.2. Identificar relaciones entre necesidades de Calidad y las Restricciones</p>
<b>Técnicas/ Herramientas</b>	<ul style="list-style-type: none"> <li>• Checklist: Plantilla de Elicitación de RNF de Calidad y Plantilla para Elicitación de RNF Restricciones.</li> <li>• Cuestionarios</li> <li>• Entrevistas</li> </ul>
<b>Productos/ Salidas</b>	<ul style="list-style-type: none"> <li>• Plantilla de Elicitación de RNF de Calidad</li> <li>• Plantilla para Elicitación de RNF Restricciones</li> <li>• Matriz de Requerimientos No Funcionales</li> </ul>
<b>Interacción con</b>	<ul style="list-style-type: none"> <li>• Proceso de Especificación de RNF Web.</li> </ul>

TABLA 6.1: Proceso de Elicitación de RNF Web.

## 6.2 Proceso de Especificación de RNF Web

En la práctica los RNF, muchas veces se documentan como parte de un requerimiento funcional o se deja registros de estos disperso entre los diferentes documentos que plantea el proceso de desarrollo, por lo tanto muchas veces, solo un subconjunto de estos se tienen en cuenta en la etapa de diseño y otros no son considerados directamente, en las etapas siguientes del ciclo de vida de desarrollo.

Por la importancia de los RNF en el desarrollo de aplicaciones Web, estos necesitan ser documentados adecuadamente para poder ser mapeados con otros requerimientos de la aplicación, y de esa forma poder ser diseñados, construidos y testeados apropiadamente.

[En la IR], “Una especificación puede ser vista como un contrato entre los usuarios y los desarrolladores de software, el cual definen el comportamiento funcional deseado del software (y otras propiedades de este como la performance, confiabilidad, etc.) sin mostrar como la funcionalidad se va a lograr” [35]. A esto surgen las siguientes preguntas:

¿Cómo se escribe un RNF?, ¿Dónde quedarán documentados los RNFs? sobre todo, en los casos donde el proceso de desarrollo no ofrece un documento formal para registrarlos.

A continuación se describe el proceso de especificación de RNF de aplicaciones Web que propone lineamientos para responder a las preguntas del párrafo precedente. La figura 6.4, ilustra gráficamente el proceso y la tabla 6.2 al final de la sección resume: Objetivo, Entradas, Actividades, Técnicas/Herramientas, Los Productos/salidas y la interacción con otros procesos de la IR.

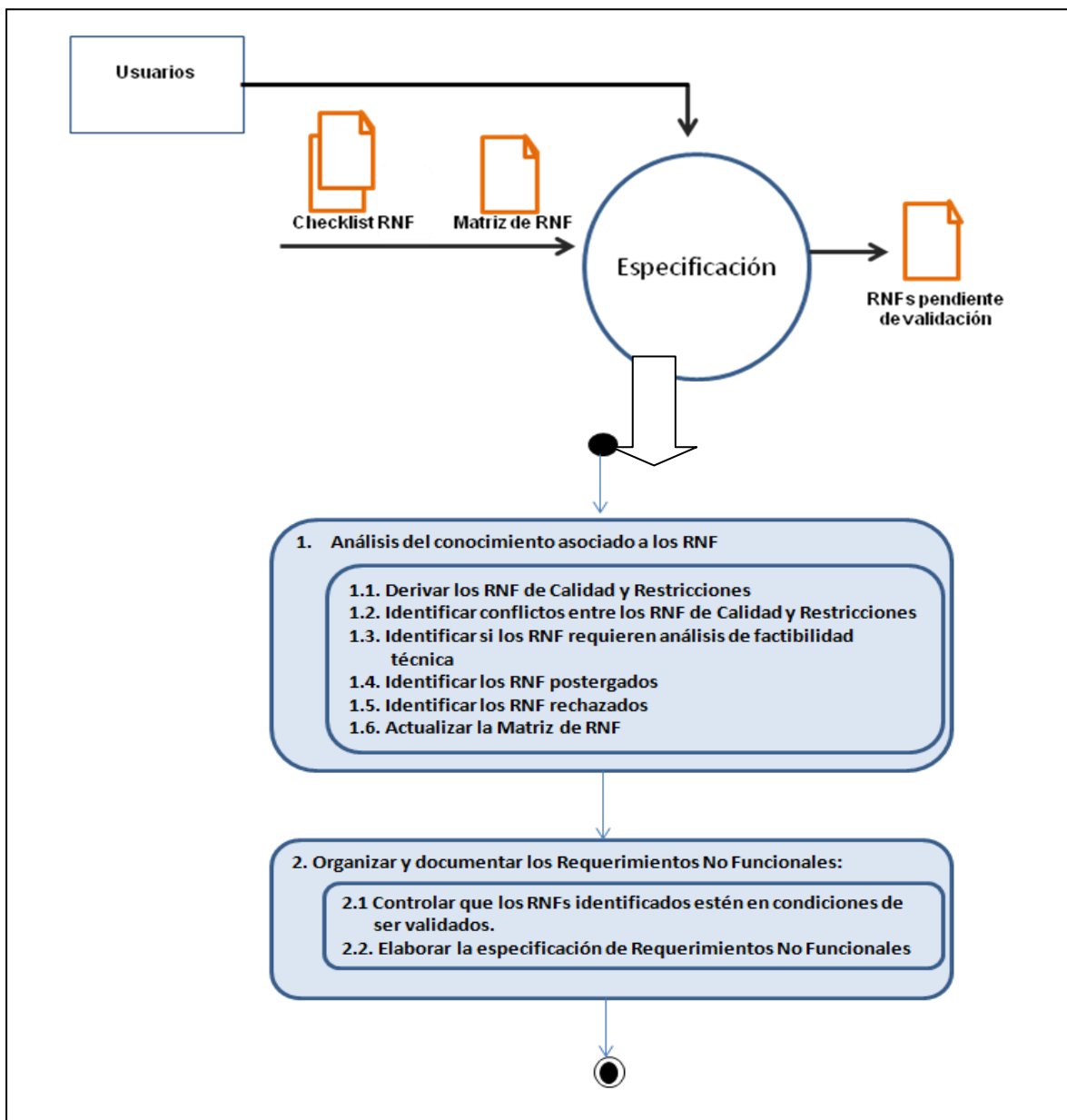


FIGURA 6.4: Proceso de Especificación de RNF Web.

## 6.2.1 Objetivo de la Especificación de RNF Web

El objetivo del proceso de especificación de RNF Web es: Generar una especificación formal de los RNF Web.

## 6.2.2 Entradas

Las entradas al proceso son:

- Clientes, Usuarios, Otros Stakeholders del Dominio, los mismos son descritos en detalle en el capítulo 7.
- Matriz de RNF, (Salida del proceso de Elicitación).

## 6.2.3 Actividades

A continuación se describen las actividades y tareas que componen el proceso en forma más detallada y se presentan las técnicas empleadas en cada una de ellas.

**1. Análisis del conocimiento asociado a los requerimientos No Funcionales:** Esta actividad propone las siguientes tareas.

1.1. *Derivar los RNF de Calidad y Restricciones:* El ingeniero de requerimientos debe redactar los RNF, en base a la información conciliada en la Matriz de RNF.

Para escribir cada RNF se debe usar el patrón de especificación que propone el proceso, el cual se describe en el capítulo 9.

Para poder especificar en forma completa cada RNF el ingeniero deberá:

- a) *Identificar la estabilidad de RNF de Calidad y Restricciones:* la estabilidad es la probabilidad de que el requerimiento sufra cambios una vez finalizada la aplicación. Es importante definirlo para prever cambios a posteriori. No es un atributo obligatorio, dado que puede no ser relevante a todas las categorías de RNF definidas. Los valores posibles:

**Alta:** El requerimiento no cambiará una vez finalizada la aplicación.

**Baja:** El requerimiento puede sufrir cambios una vez finalizada la aplicación.

Por ejemplo un requerimiento de *Arquitectura*, subcaracterística *Otros sistemas*. El Stakeholder puede decir: *La comunicación con el sistema de finanzas actualmente será por un archivo, pero cuando se implemente el nuevo sistema de finanzas tendrá que ser por Web service.*

Esta actividad demanda interacción con los Stakeholders.

b) *Identificar la prioridad de RNF de Calidad y Restricciones*: la prioridad de cada requerimiento se define en términos de:

- **Indispensable**: Significa que el RNF debe estar contemplado desde la primer versión del la aplicación.
- **Necesario**: Significa que el RNF debe estar contemplado en alguna versión del producto (menos prioridad que el anterior).
- **Preferible**: Significa que sería bueno si el RNF se pudiera incorporar en alguna versión del producto. No es indispensable.

Esta actividad demanda interacción con los Stakeholders.

c) *Identificar la versión y documentación de soporte*: la versión hace referencia a la versión de la Aplicación Web donde debe ser implementado el requerimiento. Documentación de soporte hace referencia a documentos que permiten la trazabilidad del RNF con su origen. Por ejemplo estándares corporativos o externos vinculados al RNF.

1.2. *Identificar conflictos entre los RNF de Calidad y Restricciones*: Los RNFs identificados pueden estar en conflicto unos con otros, el ingeniero de requerimientos debe identificarlos como RNF “En negociación” a fin de establecer acciones para poder resolver el conflicto.

Por ejemplo un requerimiento de calidad *Funcionalidad*, subcaracterística *Seguridad* un stakeholder necesita: *que la aplicación web respete una política de control de acceso corporativa*. Mientras que otro, *manifiesta que la aplicación tiene que tener sus propias reglas de control de acceso*.

1.3. *Identificar si los RNF requieren análisis de factibilidad técnica:* Los RNFs identificados pueden requerir análisis de factibilidad técnica. El ingeniero de requerimientos debe identificarlos como RNF “En Análisis” a fin de establecer acciones sobre los mismos.

Por ejemplo una restricción de *Arquitectura*, subcaracterística *Hardware* un stakeholder plantea la necesidad de *que una funcionalidad X de la aplicación web existente (diseñada originalmente para PC), pueda ser accedida desde un dispositivo teléfono,*

1.4 *Identificar los RNF postergados:* Si un RNF identificado es postergado, por diversos intereses o condicionado por el contexto de la aplicación. El ingeniero de requerimientos debe identificarlo como RNF “Pospuesto” y especificar las razones de esta decisión.

Por ejemplo un requerimiento de calidad de *Funcionalidad*, subcaracterística *Atractividad* un stakeholder plantea: *que la aplicación web presente contenidos multimedia con videos.* Luego en la interacción con este stakeholder y otros, se decide no considerar el RNF en la versión inicial, sino postergarlo para versiones “futuras”, quedando indefinido cuándo se realizará.

1.5 *Identificar los RNF rechazados:* Un RNF puede ser rechazado por diversos factores, por ejemplo, después de la resolución de un conflicto identificado entre requerimientos, después del análisis de factibilidad técnica, por decisiones a nivel empresa entre otros. El ingeniero de requerimientos debe identificarlo como RNF “Rechazado”, y especificar las razones de esta decisión.

1.6. *Actualizar la Matriz de RNF:* La matriz de RNF, es la herramienta que da soporte a la especificación, permite al ingeniero de requerimientos indicar la situación en la que se encuentra un requerimiento en el proceso de IR. Los estados que pueden asignarse a cada RNF son:

- **En negociación:** significa que el requerimiento fue relevado pero existe algún conflicto entre este y otros.
- **En análisis:** significa que el requerimiento fue relevado y se encuentra en estudio de factibilidad técnica.



- **Pospuesto:** significa que el requerimiento fue diferido. Informar razón en Campo Observaciones.
- **Rechazado:** significa que el requerimiento fue desestimado. Informar razón en campo Observaciones.
- **Pendiente de Validación:** Significa que el requerimiento fue especificado en la Plantilla de Especificación de RNF y está pendiente de validación.

**2. Organizar y documentar los Requerimientos No Funcionales:** Esta actividad propone las siguientes tareas.

2.1. *Controlar que los RNF identificados estén en condiciones de ser validados:* El ingeniero de requerimientos debe asegurar que no existan RNF en estado “En análisis” o “En negociación”. Los RNF en la matriz que da soporte a este proceso, deben tener los estados: Pospuesto, Rechazado, Pendiente de Validación.

2.2. *Elaborar la especificación de Requerimientos No Funcionales:* El ingeniero de requerimientos debe completar la *Plantilla de especificación de requerimientos no funcionales Web* que propone este trabajo, la cual se describe en el capítulo 9.

Para esta actividad el ingeniero de requerimientos debe considerar únicamente los RNF que están en estado “Pendiente de Validación” en la Matriz de RNF.

La Plantilla de Especificación de Requerimientos No Funcionales Web, salida del proceso de especificación, servirá de entrada a la *Validación*, último proceso de la IR.

#### **6.2.4 Técnicas/ Herramientas**

En el capítulo 3 se presentaron un conjunto de técnicas existentes en la IR para la especificación de requerimientos, en este proceso se plantea el uso de plantillas: un *Patrón para especificar RNF* y una *Planilla de Especificación de Requerimientos No Funcionales Web*, ambas se proponen en este trabajo, se describen en el capítulo 9.

La *Matriz de RNF*, la cual es otro de los productos que se propone este trabajo para dar soporte a la elicitación y especificación de RNF, se describe en el capítulo 9.

### **6.2.5 Los Productos/Salidas**

De acuerdo a las actividades propuestas en este proceso, el ingeniero de requerimientos debe generar como salida de este proceso: *La Especificación de Requerimientos No Funcionales Web*, la cual servirá de entrada a la Validación, último proceso de la IR; por lo tanto podemos referirnos a ella como *Especificación de Requerimientos No Funcionales Web Pendiente de validación*.

### **6.2.6 Interacción**

Como lo ilustra la figura 6.2 el proceso de especificación interactúa estrechamente con el proceso de Elicitación cuando se requiere mayor detalle o más información sobre el dominio de la aplicación y es afectado por este ante la presencia de un cambio en el dominio del problema. Acotado al alcance de nuestra propuesta, el proceso de especificación de RNF Web ocurre en paralelo con el proceso de elicitación de RNF Web ya que demanda la participación del usuario durante el análisis y negociación de los RNF en conflicto.

La tabla 6.2 refleja un resumen del: Objetivo, las entradas, las actividades, las técnicas/herramientas, los productos/salidas y la interacción con otros procesos de la IR, correspondiente al proceso de especificación de RNF Web.

<b>Objetivo</b>	Generar una especificación formal de los RNF Web.
<b>Entradas</b>	<ul style="list-style-type: none"> <li>• Matriz de Requerimientos No Funcionales</li> <li>• Plantilla de Elicitación de RNF de Calidad</li> <li>• Plantilla para Elicitación de RNF Restricciones</li> <li>• Clientes, Usuarios, Otros Stakeholders.</li> </ul>
<b>Actividades</b>	<ol style="list-style-type: none"> <li>1. Análisis del conocimiento asociado a los requerimientos No Funcionales             <ol style="list-style-type: none"> <li>1.1. Derivar los RNF de Calidad y Restricciones</li> <li>1.2. Identificar conflictos entre los RNF de Calidad y Restricciones</li> <li>1.3. Identificar si los RNF requieren análisis de factibilidad técnica</li> <li>1.4. Identificar los RNF postergados</li> <li>1.5. Identificar los RNF rechazados</li> <li>1.6. Actualizar la Matriz de RNF</li> </ol> </li> <li>2. Organizar y documentar los Requerimientos No Funcionales:             <ol style="list-style-type: none"> <li>2.1. Controlar que los RNFs identificados estén en condiciones de ser validados</li> <li>2.2. Elaborar la especificación de Requerimientos No Funcionales</li> </ol> </li> </ol>
<b>Técnicas/ Herramientas</b>	<ul style="list-style-type: none"> <li>• Patrón para Especificación de RNF</li> <li>• Plantilla de Especificación de Requerimientos No Funcionales Web</li> <li>• Matriz de Requerimientos No Funcionales</li> </ul>
<b>Productos/ Salidas</b>	<ul style="list-style-type: none"> <li>• Especificación de Requerimientos No Funcionales Pendiente de Validación.</li> </ul>
<b>Interacción con</b>	<ul style="list-style-type: none"> <li>• Proceso de Elicitación de RNF Web.</li> </ul>

TABLA 6.2: Proceso de Especificación de RNF Web.

## Capítulo: 7

---

### 7. Conceptos básicos para la definición de las plantillas

#### 7.1 Introducción

Este capítulo está enfocado en la descripción de los conceptos, fundamentos teóricos, consideraciones particulares empleadas en la definición de las plantillas de elicitación y especificación de RNF Web que se proponen este trabajo; las cuales se describirán detalladamente en los capítulos 8 y 9.

Al comienzo del capítulo se presentan las clases de stakeholders relevantes en la definición de las plantillas de elicitación de RNF, los mismos son considerados una fuente de elicitación de RNF para aplicaciones Web.

A continuación se describen características/subcaracterísticas del estándar de calidad internacional ISO 9126-1[34] que fue tomado como referencia para la definición de *la Plantilla de Elicitación de RNF de Calidad*, también se presentan categorías y subcategorías de restricciones definidas en *la Plantilla de Elicitación de RNF-Restricciones*. En ambos casos, para cada subcaracterística y subcategoría se mencionan consideraciones relevantes en el marco del desarrollo de aplicaciones Web, a fin de destacar la importancia de incluirlas como parte de la elicitación de RNF.

Finalmente se mencionan seis características, descritas por Wiegers en [53], que deben tenerse en cuenta para producir un buen requerimiento, las cuales son consideradas desde la perspectiva de los RNF.

#### 7.2 Clases de Stakeholders

Tal como lo definen Robertson S., Robertson en [41], “Los Stakeholders son personas que tienen un interés o un efecto en el éxito del proyecto”. “Los Stakeholders son la fuente de requerimientos” [41].

Los objetivos individuales, las necesidades y expectativas de los Stakeholders son el punto de partida del proceso de elicitación de requerimientos.

Para Robertson and Robertson en [41] el mayor problema relacionado a los stakeholders son los requerimientos que se pierden si no se identifican a todos los stakeholders, o si excluye algún stakeholder relevante durante el proceso de captura de requerimientos; lo cual nos lleva a formular esta pregunta *¿Cómo identificamos a todos los stakeholders involucrados sin pasar por alto alguno de ellos, a fin de contar con los requerimientos de todos ellos?*

En el desarrollo de sistemas tradicionales los stakeholders importantes pueden ser los clientes, los usuarios y los desarrolladores; en el caso de las aplicaciones Web se debe contemplar además dos conceptos específicos de la ingeniería de requerimientos Web: la Multidisciplinariedad y la falta de disponibilidad de stakeholders.

*Multidisciplinariedad:* el desarrollo de aplicaciones Web demanda muchas veces la participación de expertos de diferentes disciplinas, como por ejemplo: autores de contenido, expertos multimedia, expertos de usabilidad, especialistas de base de datos, arquitectos de software, expertos de dominio entre otros. Este conjunto heterogéneo de stakeholders suma complejidad al proceso de definición y consenso de requerimientos, ya que las disciplinas involucradas tienen lenguajes propios e intereses particulares diferentes que necesitan ser conciliados para alinearse a los objetivos del proyecto.

*Falta de disponibilidad de stakeholders:* Muchos stakeholders, como por ejemplos los usuarios Web potenciales, son desconocidos durante las actividades de elicitación de requerimientos. El desafío radica en encontrar un conjunto de representantes adecuados, dentro del amplio abanico de usuarios posibles en proyectos web, que puedan proporcionar las necesidades reales.

Robertson and Robertson en [41] muestran un mapa de stakeholders que despliega las clases de stakeholders que participan en las diferentes áreas (anillos organizacionales) alrededor de un producto eventual. Para cada clase de stakeholders, los autores han sugerido los roles que pueden ser representados en cada una, expresando que estos roles pueden ser adecuados a las necesidades de cada organización y /o proyecto.

En base a la categorización presentada en [41], hemos identificado un conjunto de stakeholders relevantes para las plantillas de elicitación de RNF para aplicaciones Web, los cuales serán considerados como la fuente para la elicitación de RNF:

- Clientes: “Es quien paga por el desarrollo del producto” [41].

El cliente es quien da la aceptación final de un producto, debe ser satisfecho por el producto entregado.

Dentro de los diferentes roles que pueden considerarse clientes, consideramos a:

*Manager*: El responsable de un área o departamento en una organización que será beneficiario por el producto, paga por este.

*Departamento de Marketing*: Cuando el producto será vendido a personas externas a la organización, una persona del departamento de Marketing asume el rol de cliente.

- Usuarios: “Son las personas que operarán el producto”. [41] Son quienes interactúan con el producto en alguna forma. “Son también conocidos como actores del producto.”[41]

De acuerdo al dominio del producto, podemos mencionar diferentes ejemplos de usuarios: niños, estudiantes, abogados, comerciantes, empleados de una tienda, extranjeros, usuarios remotos, público en general, usuarios casuales, gerentes, ingenieros de diferentes industrias, entre otros.

En este trabajo consideramos a:

*Usuario operador*: ...“Son las personas que tendrán contacto directo con el producto” [41]. Son quienes van a usar el producto, para quienes se hace el producto; el producto de software tendrá la finalidad de ayudarlos con el trabajo que ellos hacen.

*Miembros del público*: “Son las personas que no reciben capacitación antes de usar el producto.” [41] Cuando el producto se libera para un público más amplio.

- Otros stakeholders: Encontrar todas las personas que tienen interés y que de una y otra forma estarán afectados por el producto, para que puedan aportar requerimientos, no es tarea sencilla. La lista de roles que se listan a continuación son los *otros stakeholders* que consideramos en este trabajo.

*Expertos en la materia:* Consultores internos y externos, que pueden incluir analistas de dominio, consultores de negocio, analistas de negocio o alguien que tiene conocimiento especializado del negocio.

*Expertos técnicos:* “No necesariamente construirán el producto.

Se consideran expertos en usabilidad, consultores de seguridad, consultores de hardware, expertos en tecnologías que se pueden utilizar durante el desarrollo, especialistas en productos de software, o expertos de cualquier campo técnico que el producto podría utilizar.” [41].

*Inspectores:* “Considerar inspectores de seguridad, auditores, inspectores técnicos, y posiblemente inspectores gubernamentales, puede ser necesario para construir capacidades de inspección en el producto.”[41].

*Opinión Pública:* “¿Existe algún grupo de usuarios para su producto?” [Si existe,] “Ellos serán una de las principales fuentes de requerimientos.” [41].

Para un producto destinado al dominio público, considerar la opinión de un grupo de usuarios del dominio del producto, interesados en dar su opinión sobre el producto, puede marcar la diferencia entre aceptación y rechazo del producto.

*Operadores de Mantenimiento:* Son los encargados de mantener el producto, son quienes mantienen la infraestructura de software y hardware de la organización para la cual se construye el producto. Estas personas además tienen conocimiento de otros sistemas existentes en la organización que pueden interactuar con el producto que se está desarrollando.

### **7.3 Estándar de calidad internacional ISO/IEC 9126-1**

La plantilla para la elicitación de RNF de Calidad para aplicaciones Web que propone este trabajo, reúne las características/subcaracterísticas del estándar de calidad internacional ISO/IEC 9126-1[34] que tiene amplio reconocimiento. Cabe acotar que

actualmente, este estándar está en proceso de revisión, esperándose como resultado su aprobación y su inclusión en la nueva serie de estándares ISO/IEC 25000, también referenciados como Software Quality Requirements and Evaluation (abreviadamente, SQUARE, v. ISO, 2005).

El estándar ISO/IEC 9126-1 contiene 6 características de calidad, las cuales agrupan a 27 subcaracterísticas. La figura 7.1 presenta las 6 características definidas y su descomposición en subcaracterísticas, tal y como aparecen en el estándar ISO/IEC 9126-1[34].



Característica	Subcaracterística	
<p><b>Funcionalidad (Functionality)</b></p> <p>Capacidad del producto software para proporcionar las funcionalidades que satisfacen las necesidades explícitas e implícitas cuando el software se usa bajo unas ciertas condiciones.</p>	<p><b>Idoneidad (Suitability)</b></p>	<p>Capacidad del producto software para proporcionar un conjunto de funciones apropiado para tareas específicas y objetivos de usuario.</p>
	<p><b>Precisión (Accuracy)</b></p>	<p>Capacidad del producto software para proporcionar los resultados o efectos correctos o acordados, con el grado necesario de precisión.</p>
	<p><b>Interoperabilidad (Interoperability)</b></p>	<p>Capacidad del producto software para interactuar con uno o más sistemas especificados.</p>
	<p><b>Seguridad (Security)</b></p>	<p>Capacidad del producto software para proteger información y datos de manera que las personas o sistemas no autorizados no puedan leerlos o modificarlos, y que las personas o sistemas autorizados tengan acceso a ellos.</p>
	<p><b>Cumplimiento de la funcionalidad (Functionality compliance)</b></p>	<p>Capacidad del producto software para adherirse a normas, convenciones o regulaciones en leyes y prescripciones similares relacionadas con la funcionalidad.</p>
<p><b>Confiabilidad (Reliability)</b></p> <p>Capacidad del producto software para mantener un nivel especificado de performance cuando se usa bajo ciertas condiciones.</p>	<p><b>Madurez (Maturity)</b></p>	<p>Capacidad del producto software para evitar fallar como resultado de defectos en el software.</p>
	<p><b>Tolerancia a fallas (Fault tolerance)</b></p>	<p>Capacidad del producto de software para mantener un nivel especificado de performance en caso de defectos del software o de la vulneración de su interfaz.</p>
	<p><b>Recuperabilidad (Recoverability)</b></p>	<p>Capacidad del producto software para reestablecer un cierto nivel de performance y de recuperar los datos directamente afectados en caso de una falla.</p>
	<p><b>Cumplimiento de confiabilidad (Reliability compliance)</b></p>	<p>Capacidad del producto software para adherirse a normas, convenciones o regulaciones relacionadas con al confiabilidad.</p>
<p><b>Usabilidad (Usability)</b></p> <p>Capacidad del producto software para ser entendido, aprendido, usado y ser atractivo para el usuario, cuando se usa bajo condiciones especificadas.</p>	<p><b>Comprensibilidad (Understandability)</b></p>	<p>Capacidad del producto software que permite al usuario entender si el software es adecuado y cómo puede ser usado para las tareas particulares y condiciones de uso.</p>
	<p><b>Facilidad de aprendizaje (Learnability)</b></p>	<p>Capacidad del producto software para permitir al usuario aprender su aplicación.</p>
	<p><b>Operatividad (Operability)</b></p>	<p>Capacidad del producto software para permitir al usuario administrarlo y controlarlo.</p>
	<p><b>Atractividad (Attractiveness)</b></p>	<p>Capacidad del producto software para ser atractivo al usuario.</p>
	<p><b>Cumplimiento de usabilidad (Usability compliance)</b></p>	<p>Capacidad del producto software para adherirse a normas, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad.</p>

FIGURA 7.1: Características y Subcaracterísticas del Estándar ISO/IEC 9126-1.

Característica	Subcaracterística	
<p><b>Eficiencia (Efficiency)</b></p> <p>Capacidad del producto software para proporcionar performance apropiada, relativa a la cantidad de recursos usados, bajo condiciones determinadas.</p>	<p><b>Tiempo de comportamiento (Time behaviour)</b></p>	<p>Capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados y velocidad de rendimiento cuando esta ejecutando sus funciones bajo condiciones establecidas.</p>
	<p><b>Utilización de recursos (Resource utilisation)</b></p>	<p>Capacidad del producto software para usar las cantidades y tipos de recursos adecuados cuando el software ejecuta sus funciones bajo condiciones establecidas.</p>
	<p><b>Cumplimiento de eficiencia (Efficiency compliance)</b></p>	<p>Capacidad del producto software para adherirse a normas o convenciones relacionadas con la eficiencia.</p>
<p><b>Mantenibilidad (Maintainability)</b></p> <p>Capacidad del producto software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software, hasta cambios del ambiente y en los requerimientos y las especificaciones funcionales.</p>	<p><b>Analizabilidad (Analysability)</b></p>	<p>Capacidad del producto de software para ser diagnosticado por deficiencias o causas de fallas en el software, o de las partes a ser modificadas para ser identificados.</p>
	<p><b>Variabilidad (Changeability)</b></p>	<p>Capacidad del producto software para permitir que una modificación especificada sea implementada.</p>
	<p><b>Estabilidad (Stability)</b></p>	<p>Capacidad del producto software para evitar efectos inesperados debidos a modificaciones del software.</p>
	<p><b>Comprobabilidad (Testability)</b></p>	<p>Capacidad del producto software para permitir que el software modificado sea validado.</p>
	<p><b>Cumplimiento de mantenibilidad (Maintainability compliance)</b></p>	<p>Capacidad del producto software para adherirse a normas o convenciones relacionadas con la mantenibilidad.</p>
<p><b>Portabilidad (Portability)</b></p> <p>Capacidad del producto software para ser transferido de un ambiente a otro.</p>	<p><b>Adaptabilidad (Adaptability)</b></p>	<p>Capacidad del producto software para ser adaptado a diferentes entornos especificados, sin aplicar acciones o medios distintos de aquellos proporcionados por el software considerado, para este propósito.</p>
	<p><b>Capacidad de instalación (Installability)</b></p>	<p>Capacidad del producto software para ser instalado en un ambiente especificado.</p>
	<p><b>Co-Existencia (Co-existence)</b></p>	<p>Capacidad del producto software para coexistir con otro software independiente, en un ambiente común, compartiendo recursos comunes.</p>
	<p><b>Intercambiabilidad (Replaceability)</b></p>	<p>Capacidad del producto software para ser usado en lugar de otro producto software, para el mismo propósito, en el mismo ambiente.</p>
	<p><b>Cumplimiento de portabilidad (Portability compliance)</b></p>	<p>Capacidad del producto software para adherirse a normas o convenciones relacionadas con la portabilidad.</p>

FIGURA 7.1 Continuación:Características y Subcaracterísticas del Estándar ISO/IEC 9126-1

### 7.4 Características consideradas en la plantilla de elicitación de RNF de Calidad

En la Plantilla de Elicitación de RNF de Calidad que propone este trabajo, se consideran 5 de las 6 características de calidad definidas estándar ISO/IEC 9126-1 y 11 de las 27 subcaracterísticas.

La figura 7.2 podemos ver la estructura completa del estándar ISO/IEC 9126-1, en la columna “¿Se incluye en el Patrón?” se utilizo “Si” para indicar que la subcaracterística está contemplada en la plantilla de Elicitación de RNF de Calidad, las mismas serán descriptas a continuación. El valor “No” significa que la subcaracterística no fue incluida en la plantilla de Elicitación de RNF de Calidad, para cada caso, se agregó un comentario que respalda esta decisión.

Característica	Subcaracterística	¿Se incluye en el Patrón ?	Comentario
Funcionalidad (Functionality)	Idoneidad Suitability	[No]	Esta subcaracterística está relacionada a los requerimientos funcionales.
	Precisión Accuracy	[Si]	
	Interoperabilidad Interoperability	[Si]	
	Seguridad Security	[Si]	
	Cumplimiento de la funcionalidad Functionality compliance	[No]	Esta caracteristica está relacionada con las subcaracterísticas: Precisión, Interconectividad y Seguridad.
Confiabilidad Reliability	Madurez Maturity	[Si]	
	Tolerancia a fallas Fault tolerance	[Si]	
	Recuperabilidad Recoverability	[Si]	
	Cumplimiento de confiabilidad Reliability compliance	[No]	Esta caracteristica está relacionada con las subcaracterísticas: Madurez, Tolerancia a Fallas y recuperabilidad.

FIGURA 7.2: Características de ISO/IEC 9126-1.

Característica	Subcaracterística	¿Se incluye en el Patrón ?	Comentario
<b>Usabilidad</b> <i>Usability</i>	<a href="#">Comprensibilidad</a> <i>Understandability</i>	[Si]	
	<a href="#">Facilidad de aprendizaje</a> <i>Learnability</i>	[No]	<i>Esta característica está relacionada con Comprensibilidad y Atractividad.</i>
	<a href="#">Operatividad</a> <i>Operability</i>	[No]	<i>Esta subcaracterística está relacionada con: aspectos confidencialidad, adaptabilidad, mantenibilidad.</i>
	<a href="#">Atractividad</a> <i>Attractiveness</i>	[Si]	
	<a href="#">Cumplimiento de usabilidad</a> <i>Usability compliance</i>	[No]	<i>Esta característica está relacionada con las subcaracterísticas: Comprensibilidad y Atractividad.</i>
<b>Eficiencia</b> <i>Efficiency</i>	<a href="#">Tiempo de comportamiento</a> <i>Time behaviour</i>	[Si]	
	<a href="#">Utilización de recursos</a> <i>Resource utilisation</i>	[No]	<i>Esta subcaracterística está relacionada con aspectos de diseño y restricciones de infraestructura.</i>
	<a href="#">Cumplimiento de eficiencia</a> <i>Efficiency compliance</i>	[No]	<i>Esta subcaracterística está relacionada con la subsubcaracterística: Tiempo de Comportamiento.</i>
<b>Mantenibilidad</b> <i>Maintainability</i>	<a href="#">Analizabilidad</a> <i>Analysability</i>	[No]	<i>Esta subcaracterística está relacionada con aspectos de diseño considerando los estándares de codificación corporativo identificados en Reliability (Maturity).</i>
	<a href="#">Variabilidad</a> <i>Changeability</i>	[No]	<i>Esta subcaracterística está relacionada con aspectos de diseño.</i>
	<a href="#">Estabilidad</a> <i>Stability</i>	[No]	<i>Esta subcaracterística está relacionada con las subcaracterísticas de la Confiabilidad, y también esta relacionada con restricciones de proceso.</i>
	<a href="#">Comprobabilidad</a> <i>Testability</i>	[No]	<i>Esta subcaracterística está relacionada con restricciones de proceso.</i>
	<a href="#">Cumplimiento de mantenibilidad</a> <i>Maintainability compliance</i>	[No]	<i>Esta subcaracterística está relacionada con restricciones de proceso.</i>
<b>Portabilidad</b> <i>Portability</i>	<a href="#">Adaptabilidad</a> <i>Adaptability</i>	[Si]	
	<a href="#">Capacidad de instalación</a> <i>Installability</i>	[Si]	
	<a href="#">Co-Existencia</a> <i>Co-existence</i>	[No]	<i>Esta subcaracterística está relacionada con aspectos de diseño.</i>
	<a href="#">Intercambiabilidad</a> <i>Replaceability</i>	[No]	<i>Esta subcaracterística está relacionada con aspectos de diseño.</i>
	<a href="#">Cumplimiento de Portabilidad</a> <i>Portability compliance</i>	[No]	<i>Esta subcaracterística está relacionada con las subsubcaracterísticas: Adaptabilidad y Facilidad de instalación..</i>

FIGURA 7.2: Características de ISO/IEC 9126-1

Como lo refleja la figura 7.2, la Mantenibilidad y sus subcaracterísticas no están incluidas en la Plantilla de Elicitación de RNF de Calidad, aunque la Mantenibilidad es un factor asociado a la calidad de la aplicación final, al igual que se expresa en [22], consideramos que esta característica está relacionada con el proceso de desarrollo. Si se cuenta con un proceso de desarrollo completo, organizado, documentado y con lineamientos que den soporte a las actividades de todo el ciclo de vida de desarrollo, el mantenimiento de la aplicación Web y la gestión de cambio serán facilitadas por contar, con modelos de diseño, un software modular y la documentación asociada que permite la trazabilidad de los requerimientos para la corrección de errores o para su evolución.

A continuación se describen las once Subcaracterísticas de calidad del estándar ISO 92126-1[34] que se incluyen en la Plantilla de Elicitación de RNF de Calidad, siguiendo la siguiente estructura de organización:

**Nombre de la característica:** *Es el nombre de la característica según el estándar ISO 9126-1.*

**Definición:** *Es la definición de la característica extraída del estándar ISO 9126-1.*

**Consideraciones:** *Se describe la importancia de la característica de calidad, en el contexto de las aplicaciones web, para algunos casos se citan ejemplos de los intereses de los Stakeholders de cara a la misma.*

### **Nombre de la característica 1: Precisión (Accuracy).**

Es subcaracterística de *Funcionalidad (Functionality)*.

#### **Definición:**

“Capacidad del producto software para proporcionar los resultados o efectos correctos o acordados, con el grado necesario de precisión” [34].

#### **Consideraciones:**

Cuando los requerimientos funcionales de la aplicación Web involucran manipular números, cantidades monetarias, ejecución de cálculos, temperaturas, el ingeniero de requerimientos debe identificar las expectativas de los stakeholders: clientes y usuarios para la precisión del producto respecto de los resultados esperados por ejemplo:

- Los stakeholders esperan que todos los importes monetarios tengan una precisión de dos decimales o que la precisión de la temperatura se exprese con + 20 °C.

- El formato de la fecha para un grupo de Stakeholders debe verse como DD-MM-YY y para otros DD/MM/YYYY.

Es común que este tipo de definiciones se omitan en las especificaciones funcionales y sean supuestos en el diseño de la aplicación, por lo tanto aparecerán en la etapa de pruebas en forma de errores. Los ingenieros de software deben preocuparse por su captura desde la etapa de elicitación de requerimientos.

### **Nombre de la Subcaracterística 2: Interoperabilidad (Interoperability).**

Es subcaracterística de *Funcionalidad (Functionality)*

#### **Definición:**

“Capacidad del producto software para interactuar con uno o más sistemas especificados” [34].

#### **Consideraciones:**

Cuando la aplicación Web tiene que interactuar con otros sistemas (aplicaciones Legacy, aplicaciones y/o componentes que han sido desarrollados externamente o comprados a terceros) el ingeniero de requerimientos se debe identificarlos a fin de establecer como restricción la integración con los mismos.

Por ejemplo:

- Los Stakeholders necesitan que la aplicación Web muestre listados en base a la información que se actualiza en el módulo SAP FI / CO - Financials & Controlling.
- Los Stakeholders necesitan que las quejas de prioridad “Alta” registrados a través del sistema de “Atención al cliente” se generen automáticamente en la Aplicación Web “Resolución de problemas”.

Esta característica debe ser complementada con la definición de las restricciones de *Arquitectura /Sistemas externos* definida en la plantilla de Elicitación de RNF restricciones. Posteriormente en la etapa de diseño los desarrolladores deben definir el “como” hacer posible la integración de la aplicación Web con otros sistemas.

### **Nombre de la Subcaracterística 3: Seguridad (Security).**

Es subcaracterística de *Funcionalidad (Functionality)*.

#### **Definición:**

“Capacidad del producto software para proteger información y datos de manera que las personas o sistemas no autorizados no puedan leerlos o modificarlos, y que las personas o sistemas autorizados tengan acceso a ellos” [34].

#### **Consideraciones:**

“El desarrollo de aplicaciones seguras se ha transformado, en el comienzo de este siglo, en un tema de interés para el mercado mundial de desarrollo de software” [49], para abordar esta problemática en [49] se plantea la construcción de software seguro utilizando patrones de diseño específicos del dominio de la seguridad desde la elicitación de los requerimientos.

La seguridad es una preocupación tanto para el desarrollo de aplicaciones tradicionales como para las aplicaciones Web, sin embargo en las aplicaciones Web se presentan características particulares y estas deben ser consideradas al momento de analizar y diseñar la seguridad de la aplicación Web, como lo establecen en [5], en contraste con los programas que están instalados en hosts individuales, las aplicaciones Web están disponibles al público a través de Internet, con la intención de alcanzar un gran número de usuarios potenciales.

La seguridad en sí es un concepto bastante abstracto, los proveedores de servicios y clientes de una aplicación Web pueden tener diferentes nociones de la misma y probablemente la seguridad es el criterio más importante para una aplicación Web: La necesidad de regular el acceso a la información, verificar la identidad de los usuarios, la protección y encriptación de la información confidencial son de importancia primordial.

Si bien los responsables de Seguridad de las organizaciones, tienen la responsabilidad de asegurar que las aplicaciones no se publiquen con vulnerabilidades, son los ingenieros de software quienes deben preocuparse por la seguridad de la aplicación web desde la etapa de captura de requerimientos atendiendo a los aspectos de interés para los Stakeholders.

A los Stakeholders les preocupan aspectos como:

- ¿Quién puede acceder a los datos? ¿Quién puede modificar y borrar datos?

- ¿Dónde están los permisos de acceso? ¿Cómo hacen los usuarios para ingresar a la aplicación?, en cuanto a la integridad de la información: ¿Cómo está protegida la información? ¿Los datos están encriptados?

#### **Nombre de la Subcaracterística 4: Madurez (Maturity).**

Es subcaracterística de *Confiabilidad (Reliability)*.

#### **Definición:**

“Capacidad del producto software para evitar fallar como resultado de defectos en el software” [34].

#### **Consideraciones:**

A continuación se detalla la terminología extraída de [42], usada normalmente en la ingeniería de software:

“**Error**: una equivocación de una persona al desarrollar alguna actividad de desarrollo de software.

**Defecto**: se produce cuando una persona comete un error. Ej: un diseñador comprende mal un requerimiento y crea un diseño inadecuado.

**Falla**: es un desvío respecto del comportamiento esperado del sistema. Puede producirse en cualquier etapa.

Defecto es una vista interna, lo ven los desarrolladores. Falla es una vista externa, la ven los usuarios.”

Siguiendo las convenciones de la comunidad Web, nos referimos a las fallas como *errores*. A continuación se mencionan diferentes tipos de errores extraídos de las categorizaciones de errores Web presentadas por Alaeddine y Tian en [43] y por Guo y Sampath en [44]; las cuales pueden tomarse como guías para la tipificación de los errores que los desarrolladores y responsables de pruebas deberán considerar en el ciclo de desarrollo de la aplicación Web:

**Errores de caché**: Incluyen errores de navegador, servidor y caché de base de datos. Por ejemplo, los errores pueden ocurrir cuando los datos nuevos son cargados y la memoria caché no se actualiza.

**Errores de interface**: Incluyen llamadas a proxy, aplicaciones back-end, Legacy y bases de datos. Por ejemplo, los errores pueden producirse porque existen desajustes en la definición de la interfaz (campos que faltan o datos perdidos).



***Errores de ambiente/ configuración/ instalación:*** Son los errores por configuraciones incorrectas o faltantes, problemas de compatibilidad debido a una amplia variedad de plataformas y versiones.

***Link mal formados o rotos:*** Los errores se producen por enlaces mal formados o rotos.

***Código de error - interface de usuario:*** Incluye los errores en el código que se ejecuta del lado del cliente, scripts, applets y plug-ins.

***Errores de lógica y programación:*** Incluyen errores en el código que se ejecuta en el servidor, debido a implementación incorrecta de la lógica de negocio o errores de programación.

***Errores de apariencia:*** Incluye a los errores en el código de la aplicación que controla la forma en la cual una página web es mostrada.

- Los Stakeholders necesitan conocer “Qué pasó” cuando se produce un error.

Un mensaje de error incomprensible, no es aceptable.

- Los Stakeholders esperan que la aplicación Web identifique y trate los errores.

Tan importante como contar los buenos mensajes de avisos o errores, es contar un diseño que prevenga que ocurran los mismos.

- Los Stakeholders esperan que la aplicación Web se encuentre disponible 7x24.

Si bien esta expectativa significa una alta exigencia de calidad relacionada a la disponibilidad de la aplicación Web; la Madurez, la tolerancia a fallas y la disponibilidad son características relacionadas a:

- a) La arquitectura de la aplicación Web,
- b) Al hardware disponible e inclusive la conexión de red;

El ingeniero de requerimientos debe atender a las expectativas de los Stakeholders de perfiles técnicos y los usuarios finales a fin de identificar las restricciones que deben ser consideradas para diseñar una aplicación web confiable.

**Nombre de la Subcaracterística 5: Tolerancia a fallas (Fault tolerance).**

Es subcaracterística de *Confiabilidad (Reliability)*.

**Definición:**

“Capacidad del producto de software para mantener un nivel especificado de performance en caso de defectos del software o de la vulneración de su interfaz” [34].

**Consideraciones:**

Ver consideraciones de Madurez (Maturity).

**Nombre de la Subcaracterística 6: Tolerancia a fallas (Fault tolerance).**

Es subcaracterística de *Confiabilidad (Reliability)*.

**Definición:**

“Capacidad del producto de software para mantener un nivel especificado de performance en caso de defectos del software o de la vulneración de su interfaz” [34].

**Consideraciones:**

Ver consideraciones de Madurez (Maturity).

**Nombre de la Subcaracterística 7: Comprensibilidad (Understandability).**

Es subcaracterística de *Usabilidad (Usability)*

**Definición:**

“Capacidad del producto software que permite al usuario entender si el software es adecuado y cómo puede ser usado para las tareas particulares y condiciones de su uso” [34].

**Consideraciones:**

En el estándar ISO 9126-1 se complementa la definición de esta subcaracterística con la Nota: “Esto dependerá de la documentación y la impresión inicial dada por el software” [34].

El escenario ideal es que una aplicación Web, pueda ser navegada sin necesidad de ayuda, pero, dado que en la realidad existen dominios que demandan aplicaciones complejas (aplicaciones Web de uso comercial, instituciones bancarias o gubernamentales, educativas, entre otras), se debe incorporar un soporte de asistencia en línea integrada al contenido de la misma mediante enlaces a secciones de ayuda con ejemplos y/o secciones

de preguntas frecuentes relacionadas al contenido que se visualiza y la funcionalidad que permite realizar la aplicación web. Por ejemplo:

- Los Stakeholders necesitan que la aplicación Web proporcione soporte cuando en un proceso de registro donde se debe completar un formulario; en la apertura de una cuenta, o cuando está realizando una compra o un pedido.

Es importante destacar que además de la ayuda particular en funciones específicas, la aplicación Web debe contar con una sección de ayuda completa, y proporcionar referencias para que el usuario operador, pueda ponerse en contacto para hacer comentarios y sugerencias.

### **Nombre de la Subcaracterística 8: Atractividad (Attractiveness).**

Es subcaracterística de *Usabilidad (Usability)*.

#### **Definición:**

“Capacidad del producto software para ser atractivo al usuario” [34].

#### **Consideraciones:**

En el estándar ISO 9126-1 se complementa la definición de esta subcaracterística con la siguiente “Nota: Esto se refiere a los atributos del software que intentan hacer el software más atractivo para el usuario, tales como el uso del color y de la naturaleza del diseño gráfico.” [34].

Si bien de la definición, puede inferirse que está relacionada con aspectos vinculados a la de estética, esta característica está incluida como parte de la *usabilidad*, que es uno de los factores de calidad más relevantes para las aplicaciones Web. Una aplicación con un diseño poco atractivo, pobre, puede hacer que los usuarios rechacen la aplicación Web.

A pesar de que existe un gran número de guías de usabilidad que dan soporte al diseño de aplicaciones web [45], [46], [47], [48], la usabilidad es una característica que no puede concretarse de una sola vez; lograr que una aplicación sea atractiva y que cubra las necesidades y objetivos del usuario, requiere que de la aplicación un proceso de desarrollo iterativo, que involucre métodos que permitan validar si la aplicación Web cumple con los objetivos de usabilidad definidos.

Las diferentes categorías de aplicaciones Web, plantean requerimientos de usabilidad complejos, si lo comparamos con las aplicaciones tradicionales (No Web); sumado a ello el incremento en el uso de dispositivos móviles demanda un cambio en la forma de

afrontar los diseño para que el usuario pueda utilizar las aplicaciones Web, dado que se deben tener en cuenta las características de los contextos móviles de uso (como son las propiedades de hardware de los dispositivos móviles, las condiciones de uso y la conducta específica del usuario en este tipo de situaciones), los teléfonos inteligentes por ejemplo, se caracterizan por pantallas pequeñas, que imponen restricciones en los textos (longitud de títulos, tamaño de letra), el uso de imágenes, un concepto de navegación muy simple, entre otras.

En [5] se menciona que hay que considerar, la presión social para hacer que la Web sea accesible a personas con discapacidad motora o visual [45].

- Los Stakeholders necesitan que la aplicación web sea intuitiva y fácil de usar.

Por lo tanto las aplicaciones Web, deben ser diseñadas para los usuarios.

Es de suma importancia que el ingeniero de requerimientos durante la captura de requerimientos se preocupe en identificar las necesidades y objetivos de los usuarios o de un conjunto de usuarios potenciales, para quienes estará dirigida la aplicación, el contexto de uso en cada caso, considerar las características de los dispositivos móviles (en el caso de que aplique esta restricción). Durante las etapas siguientes del proceso de desarrollo se deberán aplicar mecanismos de retroalimentación adecuados, para validar si la aplicación Web cumple con las restricciones de usabilidad definidas.

### **Nombre de la Subcaracterística 9: Tiempo de Comportamiento (Time Behaviour).**

Es subcaracterística de *Eficiencia (Efficiency)*.

#### **Definición:**

“Capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados y velocidad de rendimiento cuando está ejecutando sus funciones bajo condiciones establecidas” [34].

#### **Consideraciones:**

Una de las características de calidad relevante para el éxito de las aplicaciones Web es la performance, en [22] cuando habla de performance, se refiere a la eficiencia con la que la aplicación aprovecha o explota los recursos disponibles. En el contexto de las aplicaciones Web, el tiempo es un recurso crítico, y la performance se mide en términos de: a) *Rendimiento*: el número de peticiones que pueden ser servidas por cada unidad de tiempo y b) *Tiempo de respuesta*: generalmente definido como el tiempo entre el envío de

una solicitud y la recepción de una respuesta. Los ítems a y b, podemos considerarlos como dentro de esta característica: *Tiempo de comportamiento*.

- Los stakeholders necesitan que la aplicación Web trabaje eficientemente.

Es importante destacar que la performance de una aplicación puede ser impactada por restricciones tecnológicas que no pueden ser manejadas desde la aplicación como lo son las características de hardware del servidor donde se encuentra disponible y el ancho de banda de la red, esta última que tiene impacto directo del lado del cliente.

Durante la captura de restricciones vinculadas a la performance, el ingeniero de requerimientos debe considerar aquellas vinculadas a las necesidades de los stakeholders relacionadas con:

*Número de peticiones de página:* El número promedio y número pico de peticiones emitidas por clientes, expresadas por ejemplo, en páginas por segundo. Este número expresa el rendimiento de la aplicación percibido por el cliente.

*Número máximo de usuarios simultáneos/ concurrentes:* Expresa la cantidad media y máxima de usuarios que acceden a la aplicación simultáneamente. El número de usuarios simultáneos afecta al back-end de la aplicación, donde las colas se pueden producir a nivel de capa de negocios y a nivel de la capa de datos.

*Tiempo de respuesta:* El número máximo de segundos que el cliente debe esperar la respuesta. Idealmente, el tiempo de respuesta debe ser relativamente estable en condiciones pico.

Ya que las mismas no solo impactarán en el diseño de la arquitectura de la aplicación sino que además pueden traducirse como parámetros a ser evaluados en la fase de prueba en el ciclo de desarrollo.

### **Nombre de la Subcaracterística 10: Adaptabilidad (Adaptability).**

Es subcaracterística de *Portabilidad (Portability)*.

#### **Definición:**

“Capacidad del producto software para ser adaptado a diferentes entornos especificados, sin aplicar acciones o medios distintos de aquellos proporcionados por el software considerado, para este propósito” [34].

#### **Consideraciones:**

En el estándar ISO 9126-1 se complementa la definición de esta subcaracterística con la siguiente “Nota: Adaptabilidad incluye la escalabilidad de la capacidad interna (por ejemplo, campos de las pantallas, las tablas, los volúmenes de transacciones, formatos de informes, etc.)” [34].

De acuerdo con la norma ISO 9126-1, la adaptabilidad implica independencia de plataforma (sistema operativo y tecnología particular), en este trabajo estas características están contempladas en la plantilla de elicitación de restricciones, la cual considera las restricciones de tecnológicas que condicionan el producto final.

Como lo indica la nota complementaria de la norma ISO 9126-1, la adaptabilidad incluye la *escalabilidad* de la capacidad interna, vamos a considerar relevante elicitar aspectos vinculados a la escalabilidad de las aplicaciones Web.

Hoy por hoy los dispositivos móviles hacen que aplicaciones Web puedan estar disponibles en cualquier lugar, por consiguiente cada vez más grupos de usuarios potenciales pueden acceder; lo cual significa que tanto el número como las características de los grupos de usuarios son realmente amplios y desconocidos. Este tipo de aplicaciones Web que están dirigidas a un público masivo, tienen que ser escalables quizás en una medida que debe proyectarse antes del proceso de desarrollo.

Una aplicación Web que no es escalable o que tiene una escalabilidad pobre, puede tener pérdidas notables de rendimiento, que serán percibidas por los usuarios una vez que se alcanza un máximo  $x$  de usuarios operando. Esta situación puede agravar aún más, dado que no solo impactará en la performance de la aplicación, sino, que en el peor de los casos puede inclusive causar pérdidas de datos; por ejemplo una aplicación de banca electrónica poco escalable, podría sufrir pérdida de datos que se traducen en transacciones financieras incompletos o incorrectas, lo cual afectará la reputación de la empresa.

En [22] se considera a la escalabilidad como un RNF relevante para las aplicaciones web, y la define como “la capacidad de aumentar la performance de la aplicación en respuesta al incremento del volumen de peticiones”. La escalabilidad se logra por clonación de elementos de la arquitectura, de este modo más recursos (servidores, conexiones de red y dispositivos de red) pueden adecuarse a un tráfico mayor. El factor clave para lograr la escalabilidad es una topología de arquitectura adecuada.

- Los Stakeholders necesitan que la aplicación Web pueda ser extensible, de modo que, cuando la carga de trabajo aumenta, sea posible agregar más potencia y mantener estable la performance de la aplicación.

Si bien el concepto de escalabilidad está ligado con el diseño de la arquitectura de la aplicación, el ingeniero de requerimientos durante la captura de necesidades, debe identificar todos aquellos aspectos vinculados a incrementar el volumen de trabajo, y definir las tasas de crecimiento previstas a fin de que en las fases siguientes del ciclo de desarrollo de la aplicación web, el diseñador de arquitectura pueda plantear una arquitectura que contemple las necesidades de escalabilidad establecidas.

### **Nombre de la Subcaracterística 11: Capacidad de Instalación (Installability).**

Es subcaracterística de *Portabilidad (Portability)*.

#### **Definición:**

“Capacidad del producto software para ser instalado en un ambiente especificado” [34].

#### **Consideraciones:**

En el estándar ISO 9126-1 se complementa la definición de esta subcaracterística con la siguiente “Nota: Si el software debe ser instalado por un usuario final, la capacidad de instalación puede afectar la idoneidad y operatividad.” [34].

Si bien esta subcaracterística está ligada con el diseño de la arquitectura de la aplicación, el ingeniero de requerimientos durante la captura de necesidades, debe identificar si es relevante para los stakeholders esta subcaracterística, a fin de que en la fase de diseño se consideren las restricciones pertinentes.

- Los Stakeholders necesitan que la aplicación Web pueda ser instalada por un perfil no necesariamente técnico.

## **7.5 Restricciones relevantes para aplicaciones Web**

La plantilla para la elicitación de RNF de restricciones de aplicaciones Web que propone este trabajo contiene cinco categorías de restricciones, algunas de cuáles agrupan un conjunto de subcategorías:

**Proceso de desarrollo:** Es el marco de trabajo que define conceptos, técnicas, métodos y herramientas para el desarrollo sistemático de las aplicaciones Web.

Las subcategorías que considera son:

- **Alcance**
- **Nivel de detalle**

**Lenguaje:** Hace referencia al lenguaje de programación que se utilizará para la codificación de la aplicación Web.

**Ambiente:** Hace referencia a las características relevantes de la infraestructura técnica (hardware y software) para las cuales se implementará la aplicación web.

Las subcategorías que considera son:

- **Implementación**
- **Sistema Operativo**
- **Base de datos**
- **Servidor de Aplicaciones**

**Arquitectura:** Es el conjunto de restricciones técnicas que deben ser consideradas en la definición de la arquitectura de la aplicación Web, sean estas, la distribución de la aplicación, sistemas externos, diseños en capas, mensajería, entre otros.

Las subcategorías que considera son:

- **Framework**
- **Diseño de Capas**
- **Sistemas Externos**
- **Solicitudes de Mails / Mensajería**
- **Accesibilidad**
- **Librerías / Componentes**

**Navegadores:** Hace referencia a los diferentes navegadores (Browser) con los cuales la aplicación web debe ser compatible.



A continuación se describe cada subcategoría de restricciones siguiendo la siguiente estructura de organización:

**Nombre de la subcategoría:** *Es el nombre de la subcategoría de restricciones.*

**Definición:** *Es el significado de la subcategoría de restricción.*

**Consideraciones:** *Se describe la importancia de la restricción, en el contexto de las aplicaciones web, se citan en algunos casos, ejemplos de los intereses de los Stakeholders de cara a la misma.*

**Nombre de la subcategoría 1: Alcance.** Es subcategoría de *Proceso de Desarrollo*.

**Definición:**

Relacionada a las fases del ciclo de vida que presenta el proceso, el cual puede ser: *Parcial* significa, que el proceso de desarrollo no contempla y describe fases para todo el ciclo de vida de desarrollo. *Completo* significa, que el proceso de desarrollo contempla y describe las fases para todo el ciclo de vida de desarrollo.

**Consideraciones:**

El alcance del proceso de desarrollo es una restricción relevante para la planificación del proyecto: las personas y roles involucrados, las actividades y los productos a generar a lo largo del desarrollo de la aplicación Web. Contar con un proceso de desarrollo completo, que cubra todas las fases del ciclo de vida, permitirá generar una aplicación de mayor calidad que si se cuenta con un proceso de desarrollo de alcance parcial. Por ejemplo un proceso de desarrollo que abarca las primeras fases del ciclo de vida de desarrollo, solamente permitirá dar garantía de la calidad de los productos intermedios, pero no dará visibilidad de la calidad la aplicación web resultante, ya que las fases del ciclo de desarrollo ausentes estarán ligadas a la experiencia del equipo de trabajo.

- Los stakeholders quieren que la aplicación Web siga un proceso de desarrollo definido por la empresa, el cual cubre todas las fases del ciclo de vida de desarrollo y tiene certificación CMMI nivel 3.
- Un stakeholder quiere aplicar un proceso de desarrollo que alcanza solamente las fases de relevamiento y diseño; la fase de construcción se llevara a cabo por diferentes grupos, no existen lineamientos específicos de diseño, codificación y las pruebas.

El ingeniero de requerimientos debe identificar esta característica del proceso de desarrollo durante la elicitación, ya que el alcance del proceso de desarrollo determina el

marco de trabajo para el desarrollo de la aplicación Web y tiene impacto directo en la calidad de la aplicación Web final.

**Nombre de la subcategoría 2: Nivel de detalle.** Es subcategoría de *Proceso de Desarrollo*.

**Definición:**

Relacionada a la profundidad del detalle de la especificación del proceso, las actividades, las técnicas utilizadas en cada fase y los productos generados en las mismas. El cual puede ser: *Parcial* significa, que el proceso de desarrollo describe solamente las actividades a ser aplicadas en cada fase, pero no describe técnicas a utilizar ni ofrece detalles o templates de los productos que deben ser producidos en cada fase.

*Completo* significa, que el proceso de desarrollo describe las actividades, las técnicas a ser aplicadas y los productos que deben ser producidos en cada fase.

**Consideraciones:**

Por lo general, en algunas organizaciones, existe un proceso de desarrollo y el problema concreto es adaptar las actividades, técnicas, notaciones y herramientas para el desarrollo de aplicaciones Web, lo cual queda librado a la experiencia de los roles involucrados en el proyecto y fomenta el desarrollo no sistemático de aplicaciones.

El nivel de detalle del proceso de desarrollo es una restricción relevante para la planificación del proyecto: sobre todo las actividades a ejecutar y productos a generar a lo largo del desarrollo de la aplicación Web, contar con un nivel de detalle incompleto en las fases del proceso de desarrollo, dará lugar a una forma de trabajo no sistemática donde la calidad de los productos intermedios y la aplicación web resultante estarán ligados a la idoneidad y experiencia del ingeniero de requerimientos.

- Los stakeholders quieren que la aplicación Web siga un proceso de desarrollo que tiene certificación CMMI, a fin de que respete los lineamientos y genere toda la documentación definida para todas las fases del ciclo de vida de desarrollo.
- El stakeholder quiere aplicar un proceso de desarrollo, que carece de lineamientos específicos para las actividades y productos a desarrollar en cada fase. No existen estándares para diseño de GUI, tampoco para el diseño de la arquitectura, existen lineamientos de codificación pero no existen lineamientos para el control de la codificación. Existen herramientas para automatizar las pruebas, pero la aplicación de las mismas está ligada a la experiencia de la persona encargada de las pruebas.

El ingeniero de requerimientos debe identificar esta característica del proceso de desarrollo durante la elicitación, debido a que el alcance y el nivel de detalle del proceso de desarrollo establecen el marco de trabajo a seguir para el desarrollo de la aplicación por lo tanto tienen impacto directo en la calidad de los productos intermedios y la aplicación Web resultante.

**Nombre de la subcategoría 3: Lenguaje de Programación.** Es subcategoría de *Lenguaje*.

**Definición:**

Define el/ los lenguaje/s de programación con el/los cual/es debe implementarse la aplicación Web.

**Consideraciones:**

Muchos factores pueden influir en la selección del lenguaje de programación de la aplicación Web, entre ellos encontramos:

*Imposición del cliente:* Es el lenguaje corporativo, existen estándares de codificación definidos para el/ los lenguajes y existen mecanismos para control de calidad del código generado.

*Costos:* Desarrollo rápido de la aplicación, reusando componentes existentes y productivos.

*Objetivo del proyecto:* Explorar nuevas tecnologías.

El ingeniero de requerimientos debe identificarlos durante la captura de necesidades debido a que el/los lenguajes influirán en las fases siguientes del ciclo de vida, principalmente en el diseño y codificación de la aplicación Web.

**Nombre de la subcategoría 4: Implementación.** Es subcategoría de *Ambiente*.

**Definición:**

Define la plataforma de implementación de la aplicación Web, como por ejemplo, notebook, dispositivos móviles, entre otros y características del servidor en el cual va a correr la aplicación Web.

**Consideraciones:**

Con el surgimiento de dispositivos móviles poderosos en los últimos años, el rol de la Web como una plataforma de contenidos, aplicaciones y servicios para estos tipos de dispositivos es cada vez más importante.

- Los stakeholders necesitan que la aplicación web esté disponible en diferentes tipos de dispositivos.

El importante que durante la captura de necesidades el ingeniero de requerimientos identifique cuales son los dispositivos tipo y modelo a fin de determinar las características de cada uno (tamaño de pantalla, memoria, software instalado, entre otros), ya que las mismas limitarán no solo el diseño de pantallas: contenidos de texto, datos, imágenes, para las funciones que serán específicas para el/los dispositivo/s, sino que además deben ser consideradas en el diseño de arquitectura de la aplicación.

**Nombre de la subcategoría 5: Sistema Operativo.** Es subcategoría de *Ambiente*.

**Definición:**

Define el/ los Sistema/s Operativo/s para el/los cual/es debe implementarse la aplicación Web.

**Consideraciones:**

El importante que el ingeniero de requerimientos identifique si la aplicación web deberá ser implementada para más de un sistema operativo, ya que ésta restricción debe ser considerada para el diseño, la codificación y en las pruebas de la aplicación, principalmente en esta última, ya que significa que la aplicación debe ser probada varias veces.

**Nombre de la subcategoría 6: Base de Datos.** Es subcategoría de *Ambiente*.

**Definición:**

Define el/ las Base/s de Dato/s para la/las cual/es debe implementarse la aplicación Web.

**Consideraciones:**

El importante que el ingeniero de requerimientos, durante la elicitación de restricciones, identifique si la aplicación web deberá integrarse o no a una base de datos existente, en tal caso debe identificar las características de la base de datos, ya que deberán ser consideradas en el diseño de arquitectura de la aplicación.

**Nombre de la subcategoría 7: Servidor de Aplicaciones.** Es subcategoría de *Ambiente*.

**Definición:**

Define el servidor de aplicaciones y versión, disponible en el ambiente productivo o host de la aplicación Web.

**Consideraciones:**

El importante que el ingeniero de requerimientos identifique durante la elicitación, cuál será el servidor de aplicaciones por ejemplo: APP Server Jboss, Webserver Tomcat, Internet information Server y la versión que estará disponible en el ambiente donde residirá la aplicación web. Esta restricción debe ser considerada para el diseño de la aplicación web.

**Nombre de la subcategoría 8: Framework.** Es subcategoría de *Arquitectura*.

**Definición:**

Determina si existen estructuras conceptuales o tecnológicas de soporte definidas a nivel de la empresa, para el diseño de la aplicación Web.

**Consideraciones:**

En algunas organizaciones existe un marco de trabajo tecnológico para estandarizar el diseño y desarrollo de aplicaciones web de la empresa; por lo general este marco de trabajo o framework corporativo, contiene un conjunto de frameworks de aplicaciones web disponibles en la industria que han sido homologados para su uso en la empresa.

Durante la captura de necesidades el ingeniero de requerimientos debe identificar si existen frameworks para las capas de presentación, de negocio y datos, ya que los mismos serán restricciones que condicionarán el diseño de la aplicación. Por otro lado, si bien un framework ofrece conceptos y componentes con la intención de hacer más fácil el desarrollo, cuando se identifica un framework desconocido o evolucionado y adaptado para la empresa, se deberá contemplar tiempo de aprendizaje del/ los framework/s por parte de las personas que estarán involucradas en el diseño y la construcción de la aplicación web.

**Nombre de la subcategoría 9: Diseño de Capas.** Es subcategoría de *Arquitectura*.

**Definición:**

Determina si la arquitectura de la aplicación Web debe considerar separación física de las capas de la arquitectura.

**Consideraciones:**

Las restricciones de infraestructura de hardware de la organización para la cual se desarrolla la aplicación, el tipo de aplicación Web, la criticidad de la funcionalidad, de los datos, el nivel de escalabilidad requerido hasta el presupuesto disponible influirán en la decisión de la separación física de las capas de arquitectura. El Stakeholder técnico puede plantear la necesidad de separación física de las capas de la arquitectura, por ejemplo, cuando se trata de aplicaciones Web que requieren mucho acceso a datos; donde quizás es conveniente pensar en un servidor dedicado solamente a la base de datos, si en el futuro surge la necesidad del negocio de ampliar a una base de datos distribuida, el impacto de implementar esta decisión afectará solo en una capa de la aplicación.

Durante la captura de necesidades el ingeniero de requerimientos debe identificar esta restricción ya que la misma impacta en las soluciones de arquitectura de la aplicación Web.

**Nombre de la subcategoría 10: Sistemas Externos.** Es subcategoría de *Arquitectura*.

**Definición:**

Determina cual será la forma de conectar la aplicación Web con sistemas externos.

**Consideraciones:**

Esta categoría es complementaria de la característica de calidad: *Funcionalidad/Interoperabilidad*, la cual permite generar una lista de los otros sistemas (aplicaciones legacy, aplicaciones y/o componentes que han sido desarrollados externamente o comprados a terceros) con los cuales deberá interactuar la aplicación web. En esta categoría el ingeniero de requerimientos debe identificar cual es la forma en que la integración con los mismos será posible, por ejemplo a través de: archivos, cola de mensajería, web services; estas restricciones impactarán posteriormente en la fase de diseño donde se establece el *cómo* será la integración de la aplicación Web con los sistemas externos. Por ejemplo en [22] los autores mencionan que para la interoperabilidad con aplicaciones Legacy: “El servidor de aplicaciones puede ser

equipado con puertas de enlace predeterminadas o kits de desarrollos de software para intercambiar mensajes y datos con las aplicaciones desarrolladas sobre plataformas con tecnologías obsoletas”.

**Nombre de la subcategoría 11: Solicitudes de Mails/Mensajería.** Es subcategoría de *Arquitectura*.

**Definición:**

Determina cual es el servidor de correo para administrar el envío y recepción de mensajes y archivos, y además determina si la aplicación Web debe contar con mensajería en línea.

**Consideraciones:**

En algunas organizaciones existe un servidor de correo a nivel corporativo, el cual debe ser considerado si las aplicaciones Web de la empresa requieren envío y recepción de mensajes y archivos. Durante la captura de necesidades el ingeniero de requerimientos debe identificar si existe un servidor de correo corporativo o no, a los efectos de establecer esta restricción en el diseño de la aplicación Web. Por otro lado si se trata de un servidor de correo desconocido, se deberá contemplar en la planificación tiempo de aprendizaje por parte de las personas que estarán involucradas en el diseño y la construcción de la aplicación web.

**Nombre de la subcategoría 12: Accesibilidad.** Es subcategoría de *Arquitectura*.

**Definición:**

Determina como será el acceso a la aplicación Web. Por ejemplo: desde internet, en una intranet o extranet.

Además determina como se implementará la autenticación y autorización de usuarios.

**Consideraciones:**

Esta restricción es relevante para el diseño de arquitectura de la aplicación Web. El ingeniero de requerimientos debe determinar si la aplicación Web:

- a) Será implementada en una intranet, dirigida a usuarios internos de la organización, pero solo podrán ingresar quienes tengan acceso autorizado.
- b) Será implementada en una extranet, dirigida a usuarios internos de la organización y usuarios externos y sólo podrán ingresar quienes tengan acceso autorizado.
- c) Estará disponible en internet, dirigida a usuarios desconocidos.

Las personas involucradas en el diseño de la aplicación deben evaluar principalmente los aspectos vinculados a la seguridad de acceso en cada caso.

**Nombre de la subcategoría 13: Librerías / Componentes.** Es subcategoría de *Arquitectura*.

**Definición:**

Determina que librerías y componentes realizados por la empresa o por terceros, deberán ser incluidos en el diseño de la arquitectura de la aplicación Web.

**Consideraciones:**

Esta categoría puede ser complementaria a la restricción de *Framework*, en el caso de que existan librerías/ componentes homologados por la empresa que tienen que ser usados para el diseño de la solución. El ingeniero de requerimientos debe identificar cuáles son las librerías, componentes y las versiones de cada uno de estos, ya que las mismas deben ser consideradas en las fases de diseño y codificación.

**Nombre de la subcategoría 14: Navegadores.** Es subcategoría de *Arquitectura*.

**Definición:**

Define el/ los navegadores para el/los cual/es debe implementarse la aplicación Web.

**Consideraciones:**

El importante que el ingeniero de requerimientos identifique si la aplicación Web debe implementarse para una versión específica de un navegador corporativo o si la misma deberá ser compatible con diferentes versiones de múltiples navegadores; ésta restricción debe ser considerada en el diseño, la codificación y en las pruebas de la aplicación, principalmente en esta última, ya que significa que la aplicación deberá ser probada con:

- Un solo navegador,
- Varias veces con diferentes versiones de un navegador
- Varias veces con diferentes versiones de diferentes navegadores.

## 7.6 Características de un buen requerimiento

A continuación se detallan las seis características, descritas por Wiegers en [53], que deben tenerse en cuenta para producir un buen requerimiento, las cuales serán



consideradas desde la perspectiva de los RNF y serán aplicadas en la definición del Patrón de especificación de RNF que propone este trabajo en el capítulo 9:

- Correcto: “Cada requerimiento debe describir precisamente la funcionalidad a ser entregada. Sólo los Stakeholders involucrados pueden determinar la exactitud de los requerimientos”... [53]

Cada RNF debe describir la regla de negocio, el estándar, las leyes internas o externas a la organización, los atributos de calidad que deben ser implementados en la aplicación Web.

- Factible: “... debe ser posible implementar cada requerimiento dentro de las capacidades conocidas y las limitaciones del sistema y su ambiente.” [53].

Cada RNF debe ser viable. Para evitar contar con RNF inviables, en algunos casos será conveniente realizar un análisis de factibilidad y tener instancias de negociación con los stakeholders involucrados.

- Necesario: “Cada requerimiento debe documentar algo que los clientes realmente necesitan o algo que es requerido para la conformidad con un requerimiento externo, una interfaz externa o un estándar”... [35].

En el caso de los RNF, también se deben considerar las restricciones de ambiente, arquitectura, características y atributos de calidad que los stakeholders requieran.

Se debe trazar cada requerimiento con su origen, si no se puede identificar el origen, posiblemente el requerimiento no es necesario realmente.

- Priorizado: “Asignar una prioridad de implementación a cada requerimiento, para indicar si es esencial, para incluirlo en una versión particular del producto.” [35]

En los RNF, esta característica quedará reflejada en la sección *Datos Específicos*; ver campo *Prioridad*.

- No ambiguo: “Un lector de un requerimiento debe ser capaz de sacar una interpretación de este. También múltiple lectores de un requerimiento deben arribar a la misma interpretación. El lenguaje natural es muy propenso a la ambigüedad. Se recomienda evitar las palabras subjetivas tales como: fácil de usar, simple, rápido, eficiente, varios, maximizar, minimizar entre otras. Las palabras que son claras para el autor, pueden no ser claras para los lectores. Cada

requerimiento debe ser escrito en forma breve, usando el lenguaje del dominio.” [53].

En el caso de los RNF, se recomienda realizar oraciones cortas y completas. Evitar párrafos con múltiples requerimientos y evitar el uso de conjunciones y/o y evitar palabras subjetivas.

- Verificable: .. “Vea si puede diseñar casos de pruebas o utilizar otros métodos de verificación, tales como inspección o demostración, para determinar si cada requerimiento es implementado correctamente en el producto”... “Requerimientos que no son coherentes, factibles o son ambiguos, no son verificables”. [53].

Aplicado a la IR, esta característica es propia del proceso de Validación. En los procesos de Elicitación y Especificación de RNF planteados en este trabajo, esta característica puede aplicarse solamente para validar los RNF de calidad de “Precisión y Atractividad” para los cuales se puede usar prototipos como herramienta de soporte en la definición de los mismos.

## Capítulo: 8

---

### 8. Plantillas para la Elicitación de RNF Web.

#### 8.1 Introducción

Este capítulo está enfocado en la descripción de los checklist, (plantillas) que fueron diseñadas para dar soporte al proceso de Elicitación de Requerimientos No Funcionales Web presentado en el capítulo 6. Cabe destacar que en el capítulo 7, se agruparon los conceptos y fundamentos teóricos que respaldan la definición de estas plantillas.

La descripción de la *Plantilla de Elicitación de RNF de Calidad* y la *Plantilla de Elicitación de RNF-Restricciones* comienza presentado la estructura de las partes que la componen y luego se describen cada una de las secciones que las componen.

Las plantillas que se proponen en este trabajo pueden considerarse no solo como una herramienta para capturar el conocimiento, sino que además sirven como una guía que da soporte al ingeniero de requerimientos durante la búsqueda y adquisición del conocimiento relevante del dominio del problema relacionado a los RNF.

#### 8.2 Estructura de la Plantilla de elicitación de RNF de Calidad

La figura 8.1, presenta la estructura de organización de la plantilla de elicitación de RNF de calidad, con la finalidad de introducir las secciones que la componen.

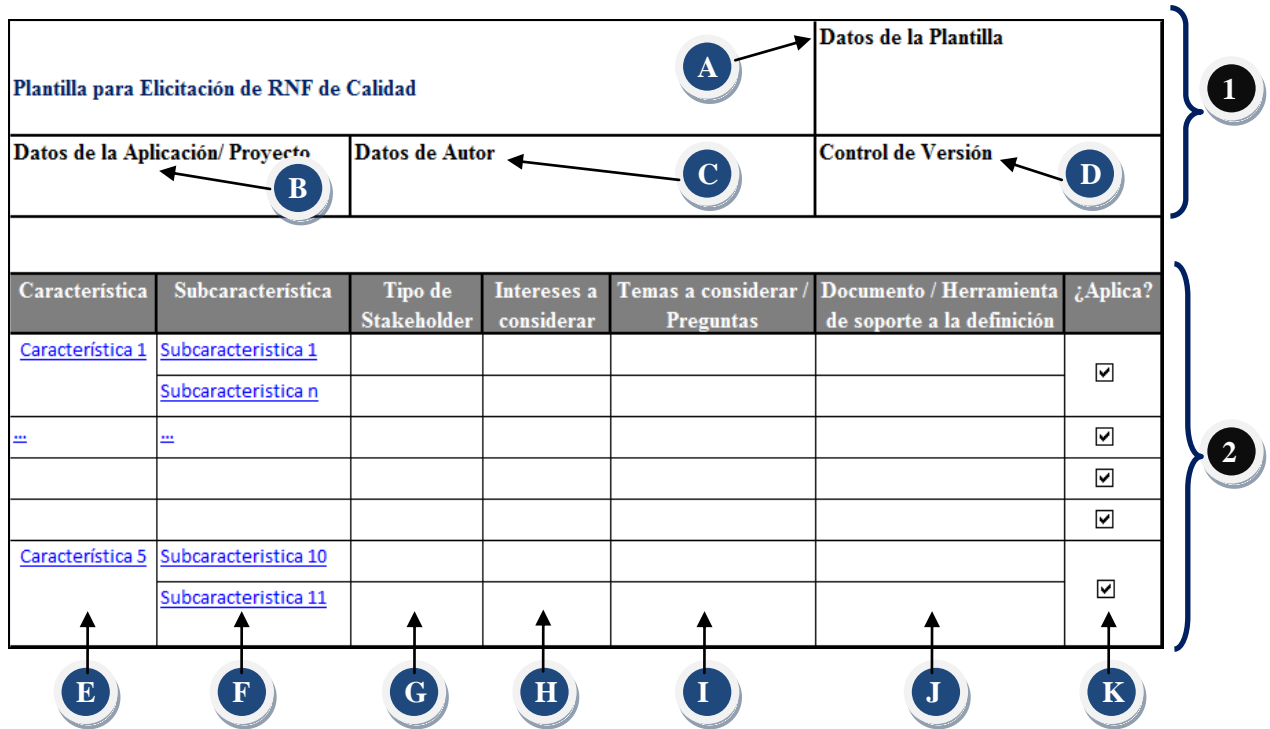


FIGURA 8.1: Estructura de la Plantilla de Elicitación de RNF de Calidad

La sección 1 **encabezado**, reúne datos e información relacionada a:

- (A) Datos de la plantilla,
- (B) A la aplicación Web objetivo del proyecto,
- (C) Al Autor,
- (D) Al documento.

La sección 2 **cuerpo**, reúne datos e información relacionada a la captura de conocimiento para la definición de RNF de calidad. A continuación se presentan sus partes:

- (E) Características: Contiene cinco características del estándar de calidad internacional ISO/IEC 9126-1[34] consideradas en la plantilla.
- (F) Subcaracterísticas: Contiene once subcaracterísticas del estándar internacional ISO/IEC 9126-1 consideradas en la plantilla.
- (G) Tipo de Stakeholders: contiene los stakeholders que se proponen como roles relevantes para la captura de los RNF, asociados a cada subcaracterística de calidad.

- (H) Intereses a considerar, describe los objetivos, las necesidades y expectativas de los stakeholders asociados a cada subcaracterística de calidad.
- (I) Temas a considerar/ Preguntas: contiene las preguntas elaboradas como parte de la plantilla para la captura del conocimiento asociado a cada subcaracterística de calidad.
- (J) Documento de soporte a la definición, hace referencia al documento y/o herramienta adicional que se plantea utilizar para complementar la captura del conocimiento relacionado a la subcaracterística correspondiente.
- (K) ¿Aplica? Indica si es necesario la captura del conocimiento asociado a cada Subcaracterística de calidad de la plantilla.

### **8.3 Descripción de la Plantilla de Elicitación de RNF de Calidad**

La plantilla fue diseñada en una hoja de cálculo Excel, dada su dimensión, optamos por presentar una vista parcial de la misma. La Figura 8.2 muestra la Plantilla para la Elicitación de RNF de Calidad para aplicaciones Web (Característica: Funcionalidad, Subcaracterística: Precisión).

La Plantilla de Elicitación de RNF de Calidad completa está disponible en el archivo Excel *Plantilla Elicitacion RNF de Calidad- v1.0.xls*. que complementa este trabajo. En la sección Anexo se describe como está organizado el archivo.

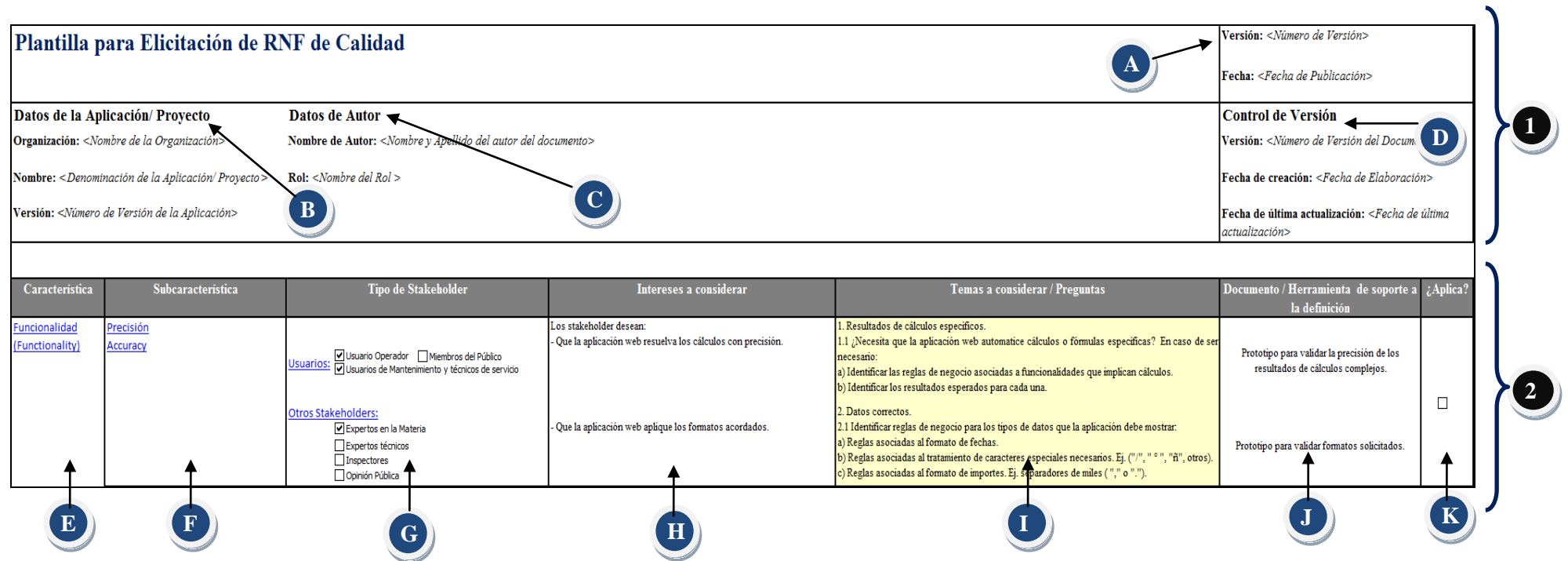


FIGURA 8.2: Plantilla para Elicitación de RNF de Calidad – Aplicaciones Web.- Vista parcial

A continuación se explican cada una de las secciones que forman parte del **encabezado** y del **cuerpo** de la Plantilla para Elicitación de RNF de Calidad.

### 8.3.1 Encabezado de la Plantilla de Elicitación de RNF de Calidad

El **encabezado** se compone de las secciones: **A, B, C, D**. La sección **A**, tiene información relacionada a la plantilla la cual no puede ser modificada por el ingeniero de requerimientos. Las secciones **B, C** y **D** contienen campos relacionados a la aplicación Web (objetivo del proyecto), al autor y al versionado del documento, los cuales deben ser completados por el ingeniero de requerimientos cuando realiza la captura de RNF.

Cada sección del **encabezado** se describe en base a la siguiente estructura de presentación:

**Contenido:** *En el Contenido se detallan los campos que forman parte de la sección, la presencia de un asterisco (\*) al lado de cada campo significa que es obligatorio ingresar los datos correspondientes.*

**Consideraciones:** *Se describen consideraciones particulares sobre el contenido de la sección, que deben tenerse en cuenta cuando se completa la plantilla.*

#### (A) Datos de la plantilla:

##### **Contenido:**

**Título:** <El nombre de la plantilla>.

**Datos Plantilla:** *Esta sección está relacionada a versión y fecha de publicación de la plantilla.*

- **Versión:** <Número de Versión>
- **Fecha:** <Fecha de Publicación>

##### **Consideraciones:**

Los datos de la plantilla no pueden ser modificados por el ingeniero de requerimientos. Esta información solo puede ser modificada por el autor de la plantilla en caso de un cambio en el contenido de la misma.

#### (B) Datos de la Aplicación:

##### **Contenido:**

- **Organización:** <Nombre de la Organización.> (\*)
- **Nombre:** <Denominación de la Aplicación/ Proyecto.> (\*)
- **Versión:** <Número de Versión de la aplicación.> (\*)

### Consideraciones:

En esta sección se debe completar el nombre de la *organización* para la cual se desarrolla el producto, el *nombre* que identificará a la aplicación o al proyecto; en el campo *versión* se debe indicar si se trata de la versión inicial del producto o una evolución del mismo.

### (C) Datos de Autor:

#### Contenido:

- **Nombre de Autor:** <Nombre y Apellido del autor del documento> (\*)
- **Rol:** <Nombre del Rol que desempeña el autor del documento> (\*)

#### Consideraciones:

En esta sección se debe completar con el *nombre y apellido* y el *rol* que desempeña el ingeniero de requerimientos que tiene a cargo la captura de RNF.

### (D) Control de Versión:

#### Contenido:

- **Versión:** <Número de Versión del Documento> (\*)
- **Fecha de creación:** <Fecha de la primera versión del documento> (\*)
- **Fecha de última actualización:** <Fecha de la última modificación del documento.> (\*)

#### Consideraciones:

Esta sección deberá ser actualizada si se realizan modificaciones relevantes durante el desarrollo del proyecto.

La *fecha de creación*, se debe completar únicamente cuando se realiza la primera versión del documento, no podrá ser modificada después.

El número de *versión* y la *fecha de última actualización* se deben modificar cada vez que se genera una versión actualizada del documento.

## 8.3.2 Cuerpo de la Plantilla de Elicitación de RNF de Calidad

El **cuerpo** de la plantilla, contiene los ítems relacionados a la captura de RNF de calidad. Está compuesta por las secciones **E, F, G, H, I, J, K**.



Las secciones **E, F, G, H, I, J**, sirven como guía, no pueden ser modificados por el ingeniero de requerimientos. La información solo puede ser modificada por el autor de la planilla en caso de un cambio en el contenido de la misma.

La sección **K**, debe ser completada por el ingeniero de requerimientos, quien, en base a la información de las secciones **F, G, H, I, J**, podrá decidir si la subcaracterística de calidad es relevante para la aplicación Web introduciendo la marca “✓” o dejarla en blanco si la misma no será considerada en la elicitación.

Para una mejor organización, primero presentaremos la sección E, que contiene cinco características del estándar ISO 9126-1 [34], siguiendo la siguiente estructura de organización:

**Nombre de la característica:** *Nombre de cada característica del estándar ISO92126-1.*

**Subcaracterísticas:** *Se detallan las subcaracterísticas, incluidas en la plantilla de elicitación de RNF de calidad.*

#### **(E) Características:**

##### **Nombre de la característica 1: Funcionalidad (Functionality)**

###### **Subcaracterísticas:**

- **Precisión (Accuracy)**
- **Interoperabilidad (Interoperability)**
- **Seguridad (Security)**

##### **Nombre de la característica 2: Confiabilidad (Reliability)**

###### **Subcaracterísticas:**

- **Madurez (Maturity)**
- **Tolerancia a fallas (Fault tolerance)**
- **Recuperabilidad (Recoverability)**

##### **Nombre de la característica 3: Usabilidad (Usability)**

###### **Subcaracterísticas:**

- **Comprensibilidad (Understandability)**
- **Atractividad (Attractiveness)**

#### **Nombre de la característica 4: Eficiencia (Efficiency)**

##### **Subcaracterísticas:**

- **Tiempo de comportamiento (Time behaviour)**

#### **Nombre de la característica 5: Portabilidad (Portability)**

##### **Subcaracterísticas:**

- **Adaptabilidad (Adaptability)**
- **Capacidad de instalación (Installability)**

A continuación se describe la sección **F**, que contiene once Subcaracterísticas de calidad del estándar ISO 92126-1, siguiendo la siguiente estructura de organización:

Nombre de la Subcaracterística: *Describe la Subcaracterística según el estándar ISO 9126-1[34] e indica cual es la característica que la agrupa.*

Secciones **G, H, I, J, K**: son particulares a cada subcaracterística por lo cual se explican con cada una de ellas.

#### **(F) Subcaracterísticas:**

**Nombre de la Subcaracterística 1: Precisión (Accuracy).** Es subcaracterística de Funcionalidad (Functionality).

##### **G. Tipo de Stakeholder**

- Usuarios: *Usuario operador.*
- Otros Stakeholders: *Expertos en la materia, Operador de Mantenimiento.*

##### **H. Intereses a considerar**

- Que la aplicación web resuelva los cálculos con precisión.
- Que la aplicación web aplique los formatos acordados.

##### **I. Temas a considerar / Preguntas:**

###### **1. Resultados de cálculos específicos.**

¿Necesita que la aplicación web automatice cálculos o fórmulas específicas? En caso de ser necesario:

- a) Identificar las reglas de negocio asociadas a funcionalidades que implican cálculos.
- b) Identificar los resultados esperados para cada una."

## **2. Datos correctos.**

Identificar reglas de negocio para los tipos de datos que la aplicación debe mostrar:

- a) Reglas asociadas al formato de fechas.
- b) Reglas asociadas al tratamiento de caracteres especiales necesarios. Ej. ("/", "°", "ñ", otros).
- c) Reglas asociadas al formato de importes. Ej. separadores de miles ("," o ".").

## **J. Documento / Herramienta de soporte a la definición**

Se recomienda utilizar la técnica de prototipo como complemento de las preguntas diseñadas para la elicitación de necesidades asociadas a la subcaracterística *Precisión*. Se plantean:

- Un prototipo para validar la precisión de los resultados de cálculos complejos.
- Un prototipo para validar formatos solicitados.

## **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la Subcaracterística 2: Interoperabilidad (Interoperability).** Es subcaracterística de Funcionalidad (Functionality).

## **G. Tipo de Stakeholder**

- Otros Stakeholders: *Expertos Técnicos, Operador de Mantenimiento.*

## **H. Intereses a considerar**

- Que la aplicación Web interactúe con otros Sistemas.

## **I. Temas a considerar / Preguntas:**

### **1. Interacción con otros sistemas, para enviar o recibir datos.**

1.1 Identificar cuáles son los sistemas.

1.2 Para cada sistema definir:

- a) Nombre y versión con la cual debe interactuar.
- b) Que datos se necesitan enviar o recibir.
- c) ¿Cuál es la frecuencia de interacción?

## **J. Documento / Herramienta de soporte a la definición**

Si bien en la plantilla no se recomienda un documento o herramienta de soporte a la definición. En caso de que cuente con un inventario actualizado de los sistemas vigentes en la organización, se puede utilizar la información detallada en el mismo para formular RNF asociados a esta característica.

## K. ¿Aplica?

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la Subcaracterística 3: Seguridad (Security).** Es subcaracterística de Funcionalidad (Functionality).

## G. Tipo de Stakeholder

- Clientes: *Manager*
- Usuarios: *Usuario operador..*
- Otros Stakeholders: *Expertos Técnicos, Operador de Mantenimiento.*

## H. Intereses a considerar

- Que la aplicación Web asegure que solo accedan los usuarios autorizados.
- Que el acceso a la aplicación Web sea seguro.
- Que se protejan los datos privados.

## I. Temas a considerar / Preguntas:

### 1. Usuarios autorizados:

1.1 ¿Necesita que la aplicación web maneje diferentes tipos de usuarios? ¿Cuáles?

1.2 ¿Cada tipo de usuario estará autorizado a realizar funciones diferentes? Definir una matriz de control de autorización de acuerdo a la plantilla de soporte: **Matriz de Control de Autorización de Accesos**, que se describe en la sección J.3.

### 2. Acceso seguro:

2.1 ¿Necesita que la aplicación web respete una política de control de acceso corporativa? En caso de que no exista definir:

- a) Reglas para las claves de usuario (contraseñas débiles)
- b) Reglas para la caducidad y cambio de la clave de usuario.
- c) Uso de algoritmos de encriptación de claves.

2.2 ¿Necesita que la aplicación web registre el historial de los accesos de usuario a fin de detectar posibles desviaciones de la política de control de acceso y uso?

2.3 ¿Se requiere una regla para la eliminación de cookies/ credenciales de autenticación? Definir periodicidad.

2.4 ¿Necesita que una cookie / credencial de autenticación se asocie a una dirección IP?

2.5 ¿ Se requiere identificación física del usuario que accede a la aplicación web?  
 ¿Necesita que se registre esta información?

**3. Protección de datos**

3.1 ¿Existen actualmente en la organización algoritmos de encriptación de datos privados? Por ejemplo: Nro. de tarjetas de crédito, Nro. de Cuentas bancarias.

**J. Documento / Herramienta de soporte a la definición**

Se propone el uso de una Matriz de Control de Autorización de Acceso, la cual se describe a continuación:

Funcionalidades	Tipos de usuarios			
	Tipo Usuario 1	Tipo Usuario 2	Tipo Usuario 3	Tipo Usuario N
Funcionalidad 1	Si			
Funcionalidad 2	Si	Si	Si	
Funcionalidad 3	Si	Si		
Funcionalidad 4	Si			
Funcionalidad 5	Si	Si		
Funcionalidad 6	Si		Si	

FIGURA 7.5: Matriz de Control de Autorización de Accesos

- En las filas se enumeran "las Funciones" de la aplicación Web, que fueron relevadas en la etapa de captura de Requerimientos Funcionales.
- En las columnas se detallan "los tipos de usuarios" de la aplicación Web, identificados en la etapa de captura de Requerimientos Funcionales.
- En las intersecciones se utiliza "Si" para indicar que el tipo de usuario está autorizado. Dejar en blanco " " cuando el usuario no está autorizado.

**K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la Subcaracterística 4: Madurez (Maturity).** Es subcaracterística de Confiabilidad (Reliability).

**G. Tipo de Stakeholder**

- Usuarios: *Usuario operador, Miembros del público.*
- Otros Stakeholders: *Expertos técnicos, Operador de Mantenimiento.*

## H. Intereses a considerar

- Que la aplicación web no tenga fallas frecuentes.
- Que la aplicación web identifique fallas por codificación defectuosa.
- Que la aplicación web identifique fallas por instalación defectuosa.

## I. Temas a considerar / Preguntas:

### 1. Defectos de programación:

Cuando la aplicación falla por un defecto en la programación, usted desea que:

- 1.1 ¿La aplicación informe el tipo de error para poder agilizar su solución?
- 1.2 En caso de que no exista un listado con tipos de errores definir una tipificación de errores.

### 2. Defectos de Instalación:

Cuando la aplicación falla por un defecto en la instalación usted desea que:

- 2.1 ¿La aplicación informe errores de instalación para poder agilizar su solución?
- 2.2 En caso de que no exista un listado con tipos de errores frecuentes en la instalación de aplicación de características similares, definir una tipificación de errores.

## J. Documento / Herramienta de soporte a la definición

Si bien en la plantilla no se recomienda un documento o herramienta de soporte a la definición. En caso de que cuente con una tipificación de los errores de los sistemas vigentes en la organización, se puede utilizar la información detallada en el mismo para formular RNF asociados a esta característica.

Se pueden tener en cuenta la tipificación de errores propuesta en [43] y mencionadas en la sección “Consideraciones”.

## K. ¿Aplica?

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la Subcaracterística 5: Tolerancia a fallas (Fault tolerance).** Es subcaracterística de Confiabilidad (Reliability).

## G. Tipo de Stakeholder

- Usuarios: *Usuario operador, Miembros del público.*
- Otros Stakeholders: *Operador de Mantenimiento.*

## H. Intereses a considerar

- Que la aplicación web siga disponible después de una falla.

## **I. Temas a considerar / Preguntas:**

### ***1. Fallas de la aplicación:***

Cuando la aplicación falla y no puede finalizar la funcionalidad que se estaba ejecutando, usted desea que:

- a) La aplicación informe al usuario la situación (que sucedió) y que le permita continuar en otra opción que no tenga conflictos.
- b) La aplicación informe al usuario la situación (que sucedió) y luego cierre la aplicación.

## **J. Documento / Herramienta de soporte a la definición**

Si bien en la plantilla no se recomienda un documento o herramienta de soporte a la definición. En caso de que cuente con una tipificación de los errores de los sistemas vigentes en la organización, se puede utilizar la información detallada en el mismo para formular RNF asociados a esta característica.

Se pueden tener en cuenta la tipificación de errores propuesta en [43] y mencionadas en la sección “Consideraciones”.

## **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la Subcaracterística 6: Recuperabilidad (Recoverability).** Es subcaracterística de Confiabilidad (Reliability).

## **G. Tipo de Stakeholder**

- Usuarios: *Usuario operador, Miembros del público.*
- Otros Stakeholders: *Operador de Mantenimiento.*

## **H. Intereses a considerar**

- Que ante una falla (por causas internas, de un dispositivo o externas a la empresa) la aplicación web se restablezca; y que recupere los datos.

## **I. Temas a considerar / Preguntas:**

### ***1. Disponibilidad y No disponibilidad***

Cuando la aplicación web queda no disponible por una falla, ¿Necesita establecer un tiempo máximo para que vuelva a estar disponible? Si la respuesta es “Si”, se debe especificar:

- a) El tiempo máximo de no disponibilidad de la aplicación web, el cual deberá estar comprendido entre x hasta y (días, horas, minutos).

## **2. Datos disponibles:**

Cuando la aplicación falla y no pudo finalizar la funcionalidad que se estaba ejecutando, usted desea que:

- a) ¿Los datos se restauren al momento previo a ejecutar la función?
- b) ¿Los datos se restauren al momento en el cual se produjo la falla?

## **J. Documento / Herramienta de soporte a la definición**

En la plantilla no se recomienda un documento o herramienta de soporte a la definición.

### **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la Subcaracterística 7: Comprensibilidad (Understandability).** Es subcaracterística de Usabilidad (Usability.)

## **G. Tipo de Stakeholder**

- Cliente: *Manager.*
- Usuarios: *Usuario operador, Miembros del público.*

## **H. Intereses a considerar**

- Que la aplicación web sea comprensible.

## **I. Temas a considerar / Preguntas:**

### **1. Manual / instructivo de uso**

¿Desea contar con un manual/instructivo de uso en forma online o papel?

### **2. Multilenguajes**

¿Necesita que la aplicación Web se publique en varios idiomas? Si la respuesta es SI, mencione cuales.

## **J. Documento / Herramienta de soporte a la definición**

En la plantilla no se recomienda un documento o herramienta de soporte a la definición.

### **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.



**Nombre de la Subcaracterística 8: Atractividad (Attractiveness).** Es subcaracterística de Usabilidad (Usability).

### **G. Tipo de Stakeholder**

- Cliente: *Manager, Departamento de Marketing*
- Usuarios: *Usuario operador, Miembros del público.*  
Otros Stakeholders: *Inspectores*

### **H. Intereses a considerar**

- Que la aplicación web sea intuitiva
- Que la aplicación web fácil de usar

### **I. Temas a considerar / Preguntas:**

#### ***Estándares de estilos de páginas***

¿Necesita que la aplicación web respete un estándar de estilos corporativo? En caso de que no exista definir reglas para:

#### ***1. Navegación***

1.1 ¿Se debe incluir mapa de uso del sitio?

#### ***2. Scroll y Paginación***

2.1 ¿Necesita usar scroll horizontal/ Vertical?

2.2 ¿Necesita usar paginación en lugar de Scroll?

#### ***3. Encabezados, títulos y etiquetas***

3.1 ¿Necesita definir encabezados de páginas fijos?

3.1 ¿Necesita definir categorías de etiquetas?

3.2 ¿Necesita Contar con títulos de páginas descriptivos?

#### ***4. Enlaces***

4.1 ¿Necesita usar enlaces mediante textos?

4.2 ¿Necesita incluir enlace a través de una imagen (imágenes clickables)?

4.3 Si las imágenes tienen enlaces parciales, por ejemplo un mapa. ¿Necesita que la aplicación proponga y/o destaque los enlaces parciales?

4.4 ¿Necesita acceder a la información de ayuda por un enlace de imagen o texto?

#### ***5. Apariencia de texto y mensajes de aviso y error***

5.1 ¿Cuál es el formato para títulos? Tamaño, tipo de letra, uso de negrita.

5.2 ¿Cuál es el formato para texto en prosa?

5.3 ¿Cuál es el código de colores e instrucciones?

5.4 ¿Cómo se destacarán datos y/o información crítica?

5.5 ¿Cómo se informarán mensajes de avisos y los mensajes de error?

#### ***6. Pantallas con Controles***

6.1 ¿Necesita utilizar radio buttons para las selecciones mutuamente excluyentes?

6.2 ¿Necesita mostrar valores predeterminados?

6.3 ¿Necesita utilizar listas para elegir uno de muchos valores?

### **7. Gráficos, Imágenes y Multimedia**

7.1 ¿Necesita mostrar imágenes de fondo?

7.2 ¿Necesita usar imágenes de etiquetas Clickables?

7.3 ¿Necesita uso de video, animación y audio?

### **8. Búsquedas:**

8.1 ¿Necesita contar con opciones de búsqueda simples/ complejas?

8.2 ¿Necesita contar con Plantillas de búsquedas?

### **9. Accesibilidad**

9.1 ¿Se deben considerar restricciones de accesibilidad? En caso de si ver [45].

## **J. Documento / Herramienta de soporte a la definición**

Se recomienda utilizar la técnica de prototipo como complemento de las preguntas diseñadas para la elicitación de necesidades asociadas a la subcaracterística *Atractividad*. Se plantea:

- Prototipo de pantallas para identificar y validar necesidades de usabilidad.

De acuerdo al dominio de aplicación Web, se recomienda consultar además:

- [45] Políticas de accesibilidad Web.
- [46] Writing for theWeb. (Nielsen, J.)
- [48] Research-Based Web Design & Usability Guidelines. (Leavitt, M. O., Shneiderman, B.)

## **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

## **Nombre de la Subcaracterística 9: Tiempo de Comportamiento (Time Behaviour).**

Es subcaracterística de Eficiencia (Efficiency).

### **G. Tipo de Stakeholder**

- Usuarios: *Usuario operador*.
- Otros Stakeholders: *Inspectores, Operador de Mantenimiento*.

### **H. Intereses a considerar**

- Que la aplicación cumpla los tiempos de respuestas establecidos.

## **I. Temas a considerar / Preguntas:**

### **1. Tiempo de respuesta.**

1.1 ¿Cuál será el número máximo de usuarios concurrentes?

1.2 ¿Cuál es tiempo de respuesta (en segundos) que el cliente deberá esperar para la respuesta de la aplicación?

### **2. Volúmenes a considerar:**

2.1 ¿Puede indicar un promedio del volumen de transacciones esperadas, por día, por mes?

2.2 ¿Puede indicar la cantidad de usuarios potenciales?

2.3 En casos que se debe manipular tablas con grandes cantidades de registros indicar ¿Cual es el volumen de registros de cada tabla relevante para la solución?

## **J. Documento / Herramienta de soporte a la definición**

En la plantilla no se recomienda un documento o herramienta de soporte a la definición.

## **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la Subcaracterística 10: Adaptabilidad (Adaptability).** Es subcaracterística de Portabilidad (Portability).

## **G. Tipo de Stakeholder**

- Clientes: *Manager*.
- Otros Stakeholders: *Inspectores, Operador de Mantenimiento*.

## **H. Intereses a considerar**

- Que la aplicación sea escalable.

## **I. Temas a considerar / Preguntas:**

### **1. Tasas de crecimiento a considerar:**

1.1 ¿Considera que la cantidad de usuarios puede incrementar en 1, 2 año? ¿En qué proporción?

1.2 ¿Considera que la cantidad de usuarios concurrentes puede incrementar en 1, 2 año? ¿En qué proporción?

1.3 ¿Considera que el número de transacciones puede incrementar en 1, 2 año? ¿En qué proporción?

### **J. Documento / Herramienta de soporte a la definición**

En la plantilla no se recomienda un documento o herramienta de soporte a la definición.

### **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la Subcaracterística 11: Capacidad de Instalación (Installability).** Es subcaracterística de Portabilidad (Portability).

### **G. Tipo de Stakeholder**

- Clientes: *Manager*.
- Otros Stakeholders: *Inspectores*

### **H. Intereses a considerar**

- Que la aplicación web pueda ser instalada por una persona que no tiene un perfil técnico.

### **I. Temas a considerar / Preguntas:**

#### **1. Instalación:**

1.1 ¿Necesita que la aplicación web respete un estándar de instalación corporativo? En caso de que no exista definir. Reglas para validar la instalación.

1.2 ¿Necesita contar con un instalador automático sin intervención de una persona?

1.3 ¿Necesita contar con un instalador automático que involucre la participación mínima de una persona?

### **J. Documento / Herramienta de soporte a la definición**

En la plantilla no se recomienda un documento o herramienta de soporte a la definición.

### **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

### 8.4 Estructura de la Plantilla para elicitación de RNF Restricciones.

La Planilla de Elicitación de RNF Restricciones, contempla la captura de las restricciones de tecnológicas y restricciones del proceso que condicionan el producto final.

La figura 8.4, presenta la estructura de organización de la plantilla de elicitación de RNF restricciones, con la finalidad de introducir las secciones que la componen.

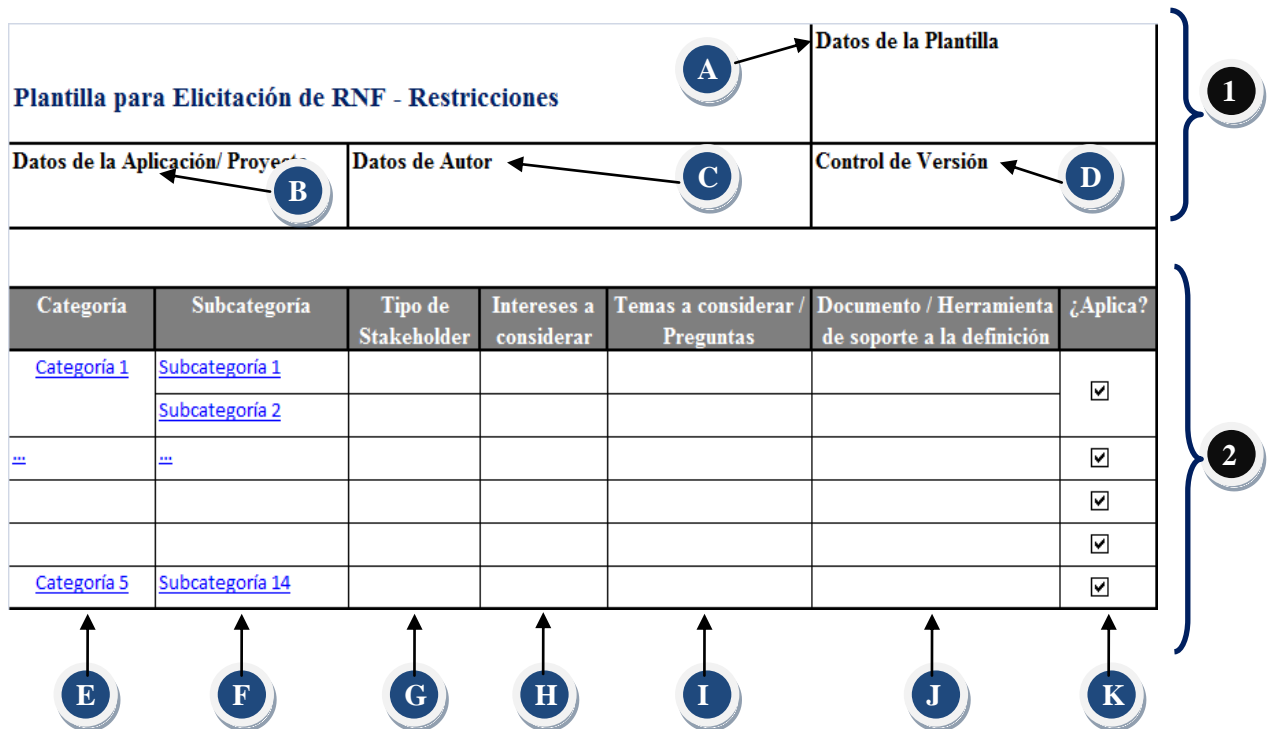


FIGURA 8.4: Estructura de la Plantilla de Elicitación de RNF Restricciones

La sección 1 **encabezado**, reúne datos e información relacionada a:

- (A) Datos de la plantilla,
- (B) A la aplicación Web objetivo del proyecto,
- (C) Al Autor,
- (D) Al documento.

La sección 2 **cuerpo**, reúne datos e información relacionada a la captura de conocimiento para la definición de RNF Restricciones. A continuación se presentan sus partes:

- (E) Categoría: Contiene las 5 categorías de restricciones consideradas en la plantilla.

- (F) Subcategoría: Contiene 14 subcategorías de restricciones consideradas en la plantilla.
- (G) Tipo de Stakeholders: contiene los stakeholders que se proponen como roles relevantes para la captura de los RNF asociados a cada subcategoría de restricciones.
- (H) Intereses a considerar, describe los objetivos, las necesidades y expectativas de los stakeholders asociados a cada cada subcategoría de restricciones.
- (I) Temas a considerar/ Preguntas: contiene las preguntas elaboradas como parte de la plantilla para la captura del conocimiento asociado a cada subcategoría de restricciones.
- (J) Documento de soporte a la definición, hace referencia al documento y/o herramienta adicional que se plantea utilizar para complementar la captura del conocimiento relacionado a la subcategoría de restricciones correspondiente.
- (K) ¿Aplica? Indica si es necesario la captura del conocimiento asociado a la Subcategoría de restricción correspondiente.

## **8.5 Descripción de la Plantilla de Elicitación de RNF Restricciones**

La plantilla fue diseñada en una hoja de cálculo Excel, dada su dimensión, optamos por presentar una vista parcial de la misma. La Figura 8.5 muestra la Plantilla para la Elicitación de RNF Restricciones para aplicaciones Web (Categoría: Ambiente).

La Plantilla de Elicitación de RNF Restricciones completa está disponible en el archivo Excel *Plantilla Elicitacion RNF Restricciones v1.0.xls* que complementa este trabajo. En la sección Anexo se describe como está organizado el archivo.

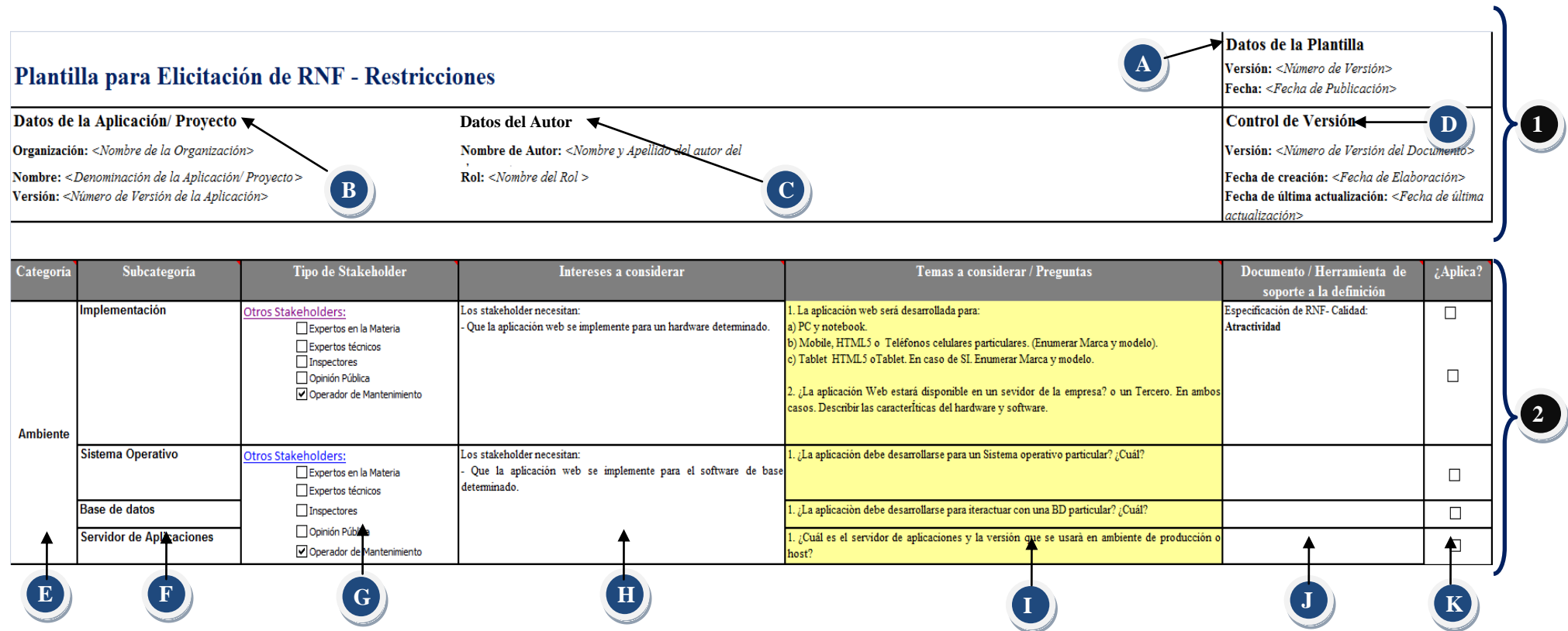


FIGURA 8.5: Plantilla para Elicitación de RNF de Restricciones - Vista parcial

A continuación se explican cada una de las secciones que forman parte del **encabezado** y del **cuerpo** de la Plantilla para Elicitación de RNF Restricciones.

### 8.5.1 Encabezado de la Plantilla de Elicitación de RNF-Restricciones

El **encabezado** se compone de las secciones: **A, B, C, D**. La sección **A**, tiene información relacionada a la plantilla, las secciones **B, C** y **D** contienen campos relacionados a la aplicación Web (objetivo del proyecto), al autor y al versionado del documento. Los campos de estas secciones se corresponden con los descritos en la Plantilla de Elicitación de RNF de Calidad, ver la explicación de la sección 8.3.1

### 8.5.2 Cuerpo de la Plantilla de Elicitación de RNF -Restricciones

El **cuerpo** de la plantilla, contiene los ítems relacionados a la captura de RNF que son restricciones. Está compuesta por las secciones **E, F, G, H, I, J, K**.

Las secciones E, F, G, H, I, J, sirven como guía, no pueden ser modificados por el ingeniero de requerimientos. La información solo puede ser modificada por el autor de la planilla en caso de un cambio en el contenido de la misma. La sección **K**, debe ser completada por el ingeniero de requerimientos, quien, en base a la información de las secciones **F, G, H, I, J**, podrá decidir si la subcategoría de restricciones es relevante para la aplicación Web introduciendo la marca “ ✓ ” o dejarla en blanco si la misma no será considerada.

Para una mejor organización, primero presentaremos la sección **E**, que contiene las categorías de restricciones incluidas en la planilla, siguiendo la siguiente estructura de organización:

**Nombre de la categoría:** *Nombre de la categoría.*

**Subcategorías:** *Se enumeran las subcategorías de restricciones incluidas en la Plantilla de Elicitación de RNF que son restricciones.*

#### **(E) Categorías de restricciones:**

**Nombre de la categoría 1: Proceso de desarrollo**

**Subcategorías:**

- **Alcance**
- **Nivel de detalle**



## **Nombre de la categoría 2: Lenguajes**

### **Subcategoría:**

- **Lenguajes de programación**

## **Nombre de la categoría 3: Ambiente**

### **Subcategorías:**

- **Implementación**
- **Sistema Operativo**
- **Base de datos**
- **Servidor de Aplicaciones**

## **Nombre de la categoría 4: Arquitectura**

### **Subcategorías:**

- **Framework**
- **Diseño de Capas**
- **Sistemas Externos**
- **Solicitudes de Mails / Mensajería**
- **Accesibilidad**
- **Librerías / Componentes**

## **Nombre de la categoría 5: Navegadores**

A continuación se describe la sección **F**, que contiene catorce Subcategorías de restricciones consideradas, siguiendo la siguiente estructura de organización:

**Nombre de la subcategoría:** *Es el nombre de la categoría de restricciones. Indica cual es la categoría que la agrupa.*

Secciones **G, H, I, J, K**, son particulares a cada subcategoría por lo tanto se explican con cada una de ellas.

## **F) Subcategorías:**

**Nombre de la subcategoría 1: Alcance.** Es subcategoría de Proceso de Desarrollo.

### **G. Tipo de Stakeholder**

- Clientes: *Manager*
- Otros Stakeholders: *Expertos en la materia.*

### **H. Intereses a considerar**

- Que la aplicación web siga un proceso de desarrollo.

### **I. Temas a considerar / Preguntas:**

#### ***1. Metodología de desarrollo***

1.1 Existe una metodología de desarrollo (Tradicional o Ágil) definida para aplicaciones

Web. ¿Cuál?

1.2 La metodología de desarrollo cubre las fases de: Relevamiento, Análisis, Diseño, Implementación y Prueba.

### **J. Documento / Herramienta de soporte a la definición**

En la plantilla no se recomienda un documento o herramienta de soporte a la definición.

### **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la subcategoría 2: Nivel de detalle.** Es subcategoría de Proceso de Desarrollo.

### **G. Tipo de Stakeholder**

- Otros Stakeholders: *Expertos en la materia, Inspectores, Operador de mantenimiento.*

### **H. Intereses a considerar**

- Que la aplicación web cumpla con los estándares corporativos de desarrollo de software.

### **I. Temas a considerar / Preguntas:**

#### ***1. Entregables***

1.1 ¿Cuales son los artefactos a generar en cada fase de la metodología de desarrollo?

1.2 ¿Cuales se consideran entregables obligatorios de cada fase, según la metodología de desarrollo?

## **2. Documentación de Requerimientos:**

¿Cómo se documentan los RF y RNF?

## **3. Diseño**

3.1 ¿Qué tipos de modelos/ diagramas se consideran obligatorios?

3.2 ¿Utilizan un lenguajes de modelado conocido, por ej. UM, u otros?

## **4. Codificación**

4.1 ¿Tienen Estándares de Codificación?

4.2 ¿Tienen Checklists de control de calidad y aceptación del código fuente?

## **5. Modelo de datos**

5.1 ¿Existe un estándar corporativos para realizar el modelado de datos? (Nomenclatura de tablas, campos entre otros).

## **6. Pruebas de la Aplicación.**

6.1 ¿Se utilizan herramientas automatizadas para realizar test de Seguridad?

6.2 ¿Se planifican pruebas en base a la guía de pruebas de OWASP (Open Web Application Security Project) Ver [54] Guía\_de\_pruebas\_de\_OWASP\_ver\_3.0.

6.3 ¿Se utilizan herramientas automatizadas para realizar pruebas de Performance?

6.4 ¿Se utilizan herramientas automatizadas para realizar pruebas de estrés?

6.5 ¿Se utilizan herramientas automatizadas para realizar pruebas de usabilidad?

6.6 ¿Qué tipos de pruebas se realizan relacionadas a la usabilidad? Ver [48] Research-Based Web Design & Usability Guidelines. (Leavitt, M. O., Shneiderman, B.) Capítulo 18.

## **7. Control de versiones**

7.1 ¿Existe algún repositorio para administrar los artefactos (documentación y código) del proyecto?

7.2 ¿Existe algún repositorio de versión de código corporativo? ¿Cuál? (Por ejemplo SVN).

## **J. Documento / Herramienta de soporte a la definición**

En la plantilla no se recomienda un documento o herramienta de soporte a la definición.

De acuerdo al dominio de aplicación Web, se recomienda consultar:

- [48] Research-Based Web Design & Usability Guidelines. (Leavitt, M. O., Shneiderman, B.) . Capítulo 18.
- [54] Guía\_de\_pruebas\_de\_OWASP\_ver\_3.0.

## **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la subcategoría 3: Lenguaje de Programación.** Es subcategoría de Lenguaje.

**G. Tipo de Stakeholder**

- Clientes: *Manager*
- Otros Stakeholders: *Operador de mantenimiento.*

**H.3 Intereses a considerar**

- Que la aplicación web se implemente en el lenguaje corporativo.

**I.3 Temas a considerar / Preguntas:**

**1. Lenguaje de programación**

Existe un Lenguaje de Programación a nivel corporativo, en el cual tenga que implementarse la aplicación web: Por ejemplo, Java, .net, PHP

**J.3 Documento / Herramienta de soporte a la definición**

En la plantilla no se recomienda un documento o herramienta de soporte a la definición.

**K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la subcategoría 4: Implementación.** Es subcategoría de *Ambiente.*

**G. Tipo de Stakeholder**

- Otros Stakeholders: *Operador de mantenimiento.*

**H. Intereses a considerar**

- Que la aplicación web se implemente para plataformas determinadas.

**I. Temas a considerar / Preguntas:**

**1. La aplicación web será desarrollada para:**

- a) PC y notebook.
- b) Mobile, HTML5 o Teléfonos celulares particulares. (Enumerar Marca y modelo).
- c) Tablet HTML5 o Tablet. En caso de Si. Enumerar Marca y modelo.

**2. La aplicación Web estará disponible en:**

- a) Un servidor propio

- b) Un servidor de una empresa externa.  
En ambos casos. Describir las características del hardware y software.

#### **J. Documento / Herramienta de soporte a la definición**

En la plantilla se recomienda consultar la Especificación de RNF de Calidad relacionada con la característica *Atractividad* y generar nuevas preguntas de ser necesario.

#### **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la subcategoría 5: Sistema Operativo.** Es subcategoría de Ambiente.

#### **G. Tipo de Stakeholder**

- Otros Stakeholders: *Operador de mantenimiento.*

#### **H. Intereses a considerar**

- Que la aplicación web se implemente para el software de base determinado..

#### **I. Temas a considerar / Preguntas:**

1. *¿La aplicación debe desarrollarse para un Sistema operativo particular? ¿Cuáles y qué versión?*

#### **J. Documento / Herramienta de soporte a la definición**

En la plantilla no se recomienda un documento o herramienta de soporte a la definición.

#### **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la subcategoría 6: Base de Datos.** Es subcategoría de Ambiente.

#### **G. Tipo de Stakeholder**

- Otros Stakeholders: *Operador de mantenimiento.*

#### **H. Intereses a considerar**

- Que la aplicación web se implemente para el software de base determinado..

### **I. Temas a considerar / Preguntas:**

1. *¿La aplicación debe desarrollarse para interactuar con una Base de Datos particular? ¿Cuál es y que versión?*

### **J. Documento / Herramienta de soporte a la definición**

En la plantilla no se recomienda un documento o herramienta de soporte a la definición.

### **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la subcategoría 7: Servidor de Aplicaciones.** Es subcategoría de Ambiente.

### **G. Tipo de Stakeholder**

- Otros Stakeholders: *Operador de mantenimiento.*

### **H. Intereses a considerar**

- Que la aplicación web se implemente para el software de base determinado..

### **I. Temas a considerar / Preguntas:**

1. *¿Cuál es el servidor de aplicaciones y la versión que se usará en ambiente de producción o host?*

### **J. Documento / Herramienta de soporte a la definición**

En la plantilla no se recomienda un documento o herramienta de soporte a la definición.

### **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la subcategoría 8: Framework.** Es subcategoría de Arquitectura.

### **G. Tipo de Stakeholder**

- Otros Stakeholders: *Operador de mantenimiento.*

#### **H. Intereses a considerar**

- Que la aplicación web se implemente en una arquitectura flexible a sus necesidades.

#### **I. Temas a considerar / Preguntas:**

- *¿Existe un framework corporativo que defina el marco de la arquitectura tecnológica a aplicar? Por ejemplo*
  - a) Frameworks para la capa de presentación
  - b) Frameworks para la capa de negocio.
  - b) Frameworks para la capa de acceso a datos.

#### **J. Documento / Herramienta de soporte a la definición**

En la plantilla se recomienda consultar la Especificación de RNF de Calidad relacionada con las características *Precisión, Madurez, Tolerancia a Fallas, Recuperabilidad, Tiempo de comportamiento, Atractividad y Comprensibilidad* y generar nuevas preguntas de ser necesario.

#### **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la subcategoría 9: Diseño de Capas.** Es subcategoría de Arquitectura.

#### **G. Tipo de Stakeholder**

- Otros Stakeholders: *Operador de mantenimiento.*

#### **H. Intereses a considerar**

- Que la aplicación web se implemente en una arquitectura flexible a sus necesidades.

#### **I. Temas a considerar / Preguntas:**

1. *¿El diseño de arquitectura debe separarse por capas físicas?*

#### **J. Documento / Herramienta de soporte a la definición**

En la plantilla se recomienda consultar la Especificación de RNF de Calidad relacionada con las características *Tiempo de comportamiento, Capacidad de Instalación, Adaptabilidad* y generar nuevas preguntas de ser necesario.

#### **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la subcategoría 10: Sistemas Externos.** Es subcategoría de Arquitectura.

**G. Tipo de Stakeholder**

- Otros Stakeholders: *Operador de mantenimiento.*

**H. Intereses a considerar**

- Que la aplicación web se implemente en una arquitectura flexible a sus necesidades.

**I. Temas a considerar / Preguntas:**

**1. Sistemas Legacy:**

En caso de que la aplicación tenga que interactuar con sistemas Legacy. ¿Cómo se conectarán? Ej: archivos, cola de mensajería, web services.

**2. Otras aplicaciones de terceros.**

En caso de que la aplicación tenga que interactuar con otras aplicaciones. ¿Cómo se conectarán? Ej: archivos, web services.

**J. Documento / Herramienta de soporte a la definición**

En la plantilla se recomienda consultar la Especificación de RNF de Calidad relacionada con la característica *Interoperabilidad* y generar nuevas preguntas de ser necesario.

**K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la subcategoría 11: Solicitudes de Mails/Mensajería.** Es subcategoría de Arquitectura.

**G. Tipo de Stakeholder**

- Otros Stakeholders: *Expertos Técnicos, Operador de mantenimiento.*

**H. Intereses a considerar**

- Que la aplicación web se implemente en una arquitectura flexible a sus necesidades.

**I. Temas a considerar / Preguntas:**

**1. Envío de mails:**

1.1 ¿La aplicación deberá integrarse con un servidor de mails corporativo? ¿Cuál? Por ejemplo Outlook, Lotus Notes entre otros.



1.2 Existe un servicio o componente corporativo para administrar el envío y recepción de mail.

## **2. Mensajería en Línea**

¿La aplicación deberá contar con mensajería en línea?

### **J. Documento / Herramienta de soporte a la definición**

En la plantilla no se recomienda un documento o herramienta de soporte a la definición.

### **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la subcategoría 12: Accesibilidad.** Es subcategoría de Arquitectura.

### **G. Tipo de Stakeholder**

- Otros Stakeholders: *Expertos Técnicos, Operador de mantenimiento.*

### **H. Intereses a considerar**

- Que la aplicación web se implemente en una arquitectura flexible a sus necesidades.

### **I. Temas a considerar / Preguntas:**

#### **1. Acceso:**

La aplicación se podrá acceder a través de ¿Internet, una Intranet o Extranet?

#### **2. Seguridad:**

2.1 ¿Existe un estándar corporativo para implementar la autenticación y autorización de usuarios? Ej. Active Directory.

2.2 ¿Cómo son gestionados y dónde están los permisos de acceso? ¿Los datos están encriptados?

2.3 ¿Cómo son encriptados los datos privados, por ejemplo las claves y el Número de la tarjeta de crédito?

### **J. Documento / Herramienta de soporte a la definición**

En la plantilla se recomienda consultar la Especificación de RNF de Calidad relacionada con las características *Seguridad, Adaptabilidad* y generar nuevas preguntas de ser necesario.

### **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la subcategoría 13: Librerías / Componentes.** Es subcategoría de Arquitectura.

### **G. Tipo de Stakeholder**

- Otros Stakeholders: *Expertos Técnicos, Operador de mantenimiento.*

### **H. Intereses a considerar**

- Que la aplicación web se implemente en una arquitectura flexible a sus necesidades.

### **I. Temas a considerar / Preguntas:**

#### **1. Librerías / Componentes:**

1.1 En caso de que la aplicación Web tenga que usar librerías de terceros y/o componentes desarrollados en la empresa, indicar como será la interacción con cada uno de ellos.

1.2 Si es importante utilizar recursos técnicos que requieran la descarga de plug-ins ¿La aplicación deberá informar al usuario esta situación y explicar la importancia de hacerlo?

#### **2. Motores de búsquedas:**

¿Es importante contar con un motor de búsquedas para todo el sitio?

### **J. Documento / Herramienta de soporte a la definición**

En la plantilla se recomienda consultar la Especificación de RNF de Calidad relacionada con la característica *Interoperabilidad* y generar nuevas preguntas de ser necesario.

### **K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

**Nombre de la subcategoría 14: Navegadores.** Es subcategoría de Arquitectura.

**G. Tipo de Stakeholder**

- Clientes: *Manager, Departamento de Marketing*
- Otros Stakeholders: *Inspectores, Operador de mantenimiento.*

**H. Intereses a considerar**

- Que la aplicación web se implemente para el navegador corporativo..

**I. Temas a considerar / Preguntas:**

**I. Navegadores:**

1.1 ¿La aplicación web debe ser realizada para un navegador corporativo? Defina cual y que versión.

1.2 ¿Necesita que la aplicación web sea compatible con otros navegadores? Defina cuáles y qué versiones.

**J. Documento / Herramienta de soporte a la definición**

En la plantilla no se recomienda un documento o herramienta de soporte a la definición.

**K. ¿Aplica?**

La marca “✓” indica que se debe realizar la captura de conocimiento relacionado a esta subcaracterística. Cuando queda en blanco no será considerada.

## Capítulo: 9

---

### 9. Plantillas para Especificación de RNF Web.

#### 9.1 Introducción

Este capítulo está enfocado en la descripción de las plantillas que fueron diseñadas para dar soporte al proceso de Especificación de Requerimientos No Funcionales Web presentado en el capítulo 6.

Se describen detalladamente un *Patrón para especificar RNF* y una *Plantilla de Especificación de RNF Web*, diseñados en base a las prácticas recomendadas en el estándar IEEE 830-1998 [36] para la especificación de requerimientos de software.

Finalmente se presenta una *Matriz de RNF*, la cual fue diseñada para dar soporte a los procesos de Elicitación y Especificación de RNF Web respectivamente.

#### 9.2 Patrón para especificación de RNF.

La figura 9.1, presenta la estructura y secciones que componen al patrón para especificación de RNF, el mismo fue diseñado en base a las prácticas recomendadas en el estándar IEEE 830-1998 [36] para la especificación de requerimientos de software.

Identificación	Tipo	Categoría	Subcategoría	Id. Requerimiento
		1-RNF-Calidad	Funcionalidad	Tiempo de comportamiento
Descripción del requerimiento				
Datos específicos	Id. Req. Asociado		Estabilidad	Prioridad
	Versión #		Documentación de Soporte/ Versión	
	Stakeholder		Solicitante	

FIGURA 9.1: Patrón para Especificación de RNF

El patrón está compuesto por las secciones **A**, **B**. La sección **A** reúne información relacionada a identificación y la descripción del RNF, la sección **B** contiene datos específicos del RNF. Cada sección se describe en base a la siguiente estructura de presentación:

*Contenido:* En el Contenido se detallan los campos que forman parte de la sección, la presencia de un asterisco (\*) al lado de cada campo significa que es obligatorio ingresar los datos correspondientes.

*Consideraciones:* Se describen consideraciones particulares sobre el contenido de la sección, que deben tenerse en cuenta cuando se utiliza el patrón para escribir los RNF.

**A) Identificación:**

**Contenido:**

- **Tipo:** < Tipo de Requerimiento No Funcional > (\*)
- **Categoría:** < Indica la característica de Calidad o la Categoría de Restricción a la que corresponde el RNF >. (\*)

- **Subcategoría:** <Indica la Subcaracterística de Calidad o la Subcategoría de Restricción a la que corresponde el RNF>. (\*)
- **Id. De Requerimiento:** <Identifica unívocamente al RNF la nomenclatura sugerida es <CodProy>-<NroSeq> Ej: PXX-001-0001> (\*)
- **Descripción del requerimiento:** <Descripción del RNF> (\*)

### Consideraciones:

Los campos de esta sección deben ser completados por el ingeniero de requerimientos, la información debe extraerse de la Matriz de RNF.

*El Tipo* hace referencia al tipo del RNF. Los valores posibles son: 1 – RNF de Calidad, 2 - RNF Restricciones.

*La Categoría* hace referencia la característica de calidad o categoría de restricción del RNF. Los valores posibles son las cinco características de Calidad, o las cinco Categorías de Restricciones que proponen las Planillas de Elicitación de RNF de calidad y restricciones descritos en el capítulo 8.

*La Subcategoría* hace referencia la subcaracterística de calidad o subcategoría de restricción del RNF. Los valores posibles son las 11 subcaracterísticas de Calidad, o las 14 subcategorías de restricciones que proponen las Planillas de Elicitación de RNF de calidad y restricciones descritos en el capítulo 8.

*El Id. De Requerimiento:* es la identificación del requerimiento, en [36] se establece que todos los requerimientos deben ser identificables unívocamente. En este patrón se propone la siguiente nomenclatura:<PXXX-RNF-000> donde PXXX sirve para identificar el proyecto, RNF es el prefijo que indica que es un Requerimiento No Funcional y 000 es un número de secuencia de 0 a 999 para identificar cada RNF Ej: *PM01-RNF-001*.

*La Descripción del Requerimiento:* Contiene la declaración del Requerimiento. Llegar a contar con una descripción adecuada del RNF demandará de interacción con el proceso de Elicitación y de la participación de los stakeholders involucrados. Cada requerimiento debe considerar las características descritas por Wiegers en [53] para producir un buen requerimiento; las cuales fueron presentadas en el capítulo 7 y son: Correcto, Factible, Necesario, Priorizado, No ambiguo y Verificable.

## B) Datos Específicos:

### Contenido:

- **Id. Req. Asociado:** <Hace referencia al RNF al cual está asociado>.
- **Estabilidad:** <Hace referencia permanencia del RNF>.
- **Prioridad:** <Indica la prioridad del RNF, asignada por el/los Stakeholder/s>. (\*)
- **Versión #:** <Indica la versión de la Aplicación Web para la cual el RNF debe ser implementado>. (\*)
- **Documentación de Soporte/ Versión:** <Nombre y versión de la documentación de referencia>(\*)
- **Stakeholder:** <Rol del Stakeholder que originó el requerimiento> (\*)
- **Solicitante:** <Nombre de la Persona y/o Área > (\*)

### Consideraciones:

Para completar los campos de esta sección de la plantilla el ingeniero de requerimientos debe considerar lo siguiente:

La *Estabilidad*: es la probabilidad de que el requerimiento sufra cambios una vez finalizada la aplicación. Es importante definirlo para prever cambios a posteriori. No es un campo obligatorio, dado que puede no ser relevante a todas las categorías de RNF definidas, los valores posibles son Baja indica que el requerimiento puede sufrir cambios una vez finalizada la aplicación, Alta indica que el requerimiento no cambiará una vez finalizada la aplicación.

La *Prioridad*: Indica la prioridad del RNF, los valores posibles son 1-Indispensable: indica que el RNF debe estar contemplado desde la primera versión del producto. 2-Necesario: indica que el RNF debe estar contemplado en alguna versión del producto (menos prioridad que el anterior). 3-Preferible: indica que sería bueno si se pudiera incorporar en alguna versión del producto. No es indispensable.

La *Versión #*: Indica la versión de la aplicación Web para la cual debe ser implementado. Este campo tiene sentido cuando se trata de aplicaciones grandes, donde la gestión de versiones sea una tarea crítica.

*Documentación de Soporte/ Versión:* Hace referencia al nombre y versión de los documentos de Especificaciones de RF, Estándares corporativos.

*Stakeholder:* Describe el rol del Stakeholder que originó el requerimiento.

*Solicitante:* Nombre de la Persona/Área que solicita el requerimiento, es importante en el momento de la negociación de los RNF.

### **9.3 Estructura de la Plantilla de Especificación de RNF Web.**

Esta planilla integra todos los RNF relevados; reúne las secciones del patrón de RNF descrito en la sección 9.2 y responde a las prácticas recomendadas en el estándar IEEE 830-1998 [36] para la especificación de requerimientos de software.

La figura 9.2 presenta la estructura de organización de la plantilla de especificación de RNF Web, con la finalidad de introducir las secciones que la componen.



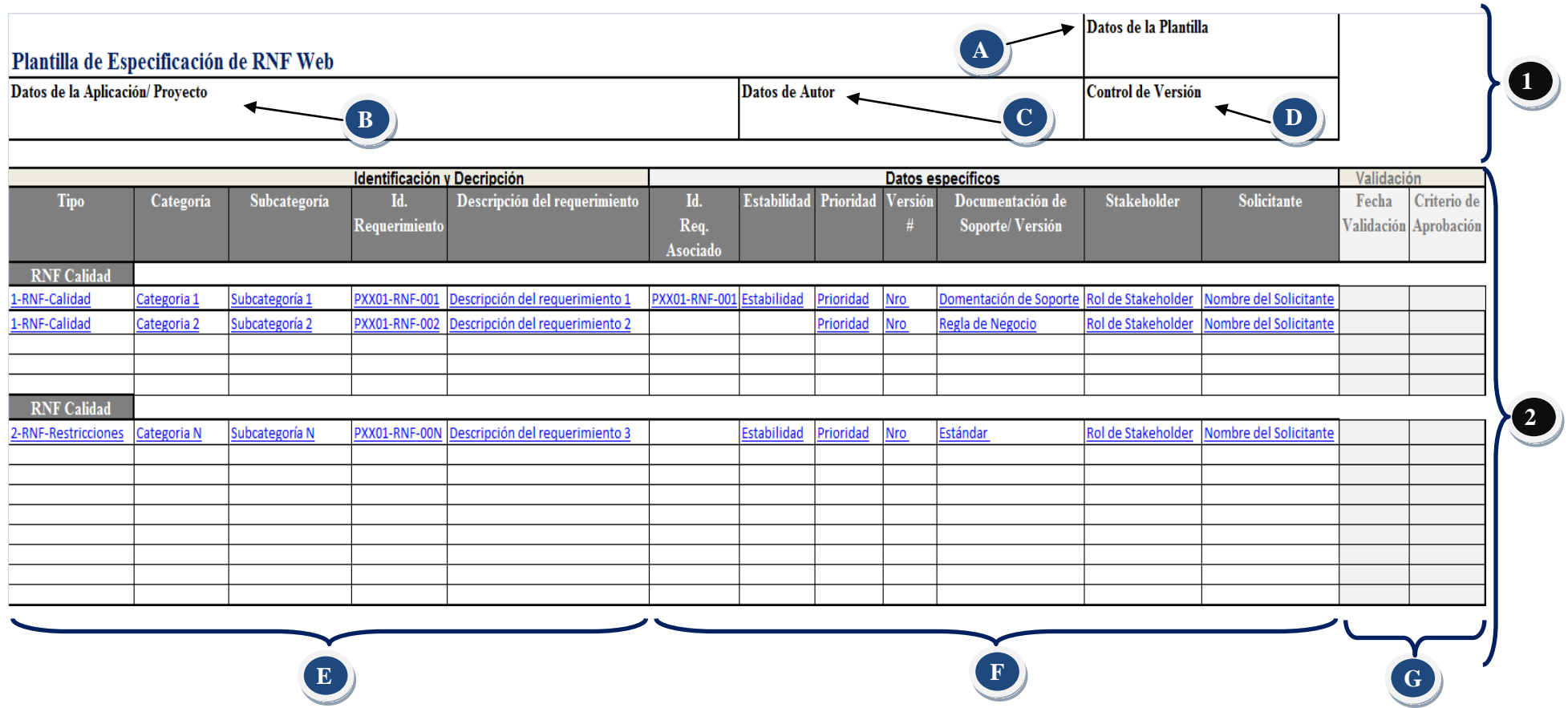


FIGURA 9.2: Estructura de la Plantilla de Especificación de RNF Web

La sección 1 **encabezado**, reúne datos e información relacionada a:

- (A) Datos de la plantilla,
- (B) A la aplicación Web objetivo del proyecto,
- (C) Al Autor,
- (D) Al documento.

La sección 2 **cuerpo**, reúne datos e información relacionada a la especificación los RNF de calidad y restricciones, está compuesto por:

- (E) Identificación y descripción de los RNF.
- (F) Datos específicos de los RNF.
- (G) Datos de Validación. Si bien el proceso de Validación de la IR está fuera del alcance de esta tesis, la plantilla contiene los campos más relevantes a completarse cuando se lleve a cabo la validación de los RNF.

#### **9.4 Descripción de la Plantilla de Especificación de RNF Web**

La Figura 9.3 muestra la Plantilla de Especificación de RNF web, esta plantilla fue diseñada en una hoja de cálculo de Excel, la misma está disponible en el archivo Excel *Plantilla de Especificacion RNF Web-v1.0.xls* que complementa este trabajo. En la sección Anexo se describe como está organizado el archivo.

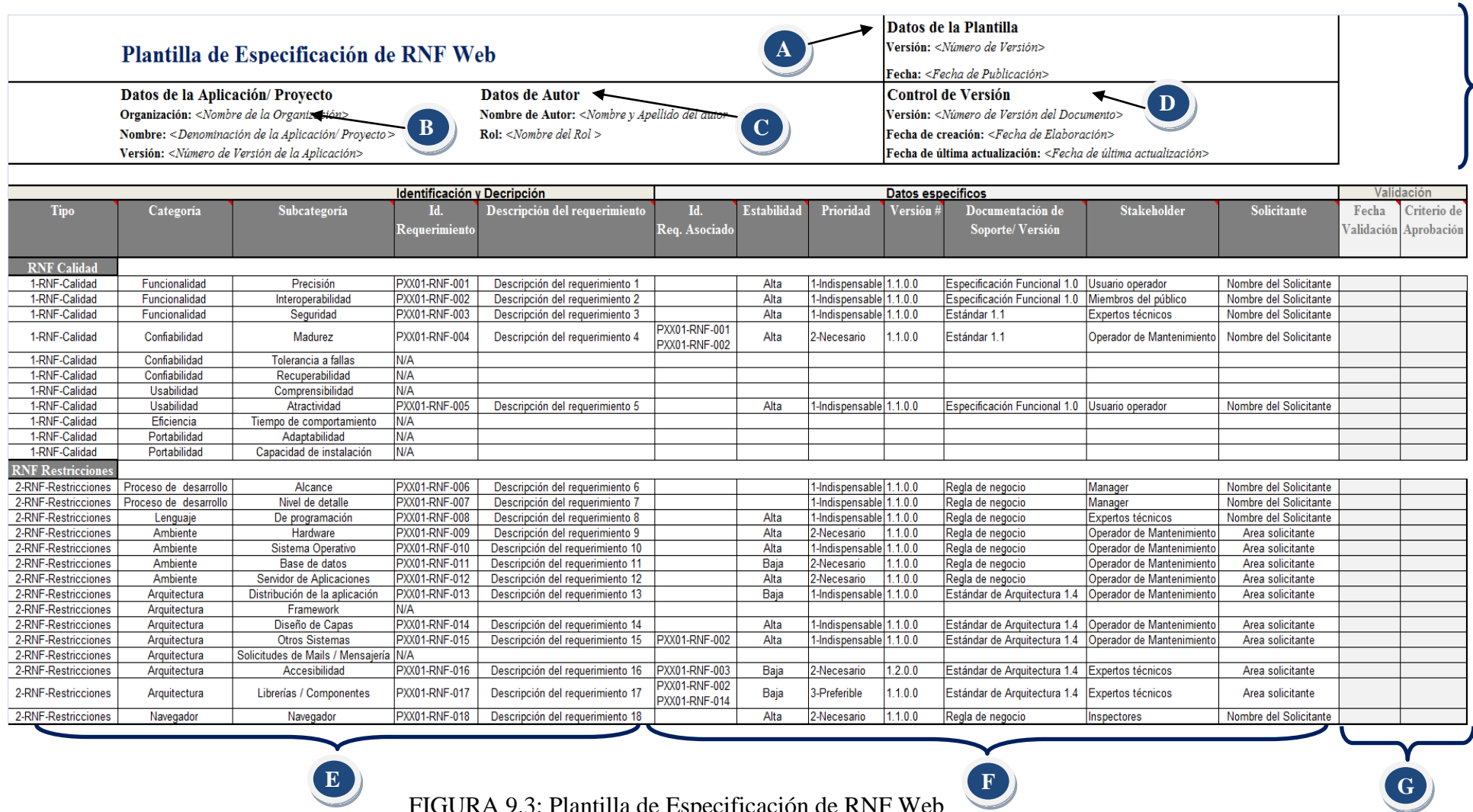


FIGURA 9.3: Plantilla de Especificación de RNF Web

### 9.4.1 Encabezado de la Plantilla de Especificación de RNF Web

El **encabezado** se compone de las secciones: **A, B, C, D**. La sección **A**, tiene información relacionada a la plantilla, las secciones **B, C** y **D** contienen campos relacionados a la aplicación Web (objetivo del proyecto), al autor y al versionado del documento. Los campos de estas secciones se corresponden con los descritos en la Plantilla de Elicitación de RNF de Calidad, ver la explicación de la sección 8.3.1

### 9.4.2 Cuerpo de la Plantilla de Especificación de RNF Web

La sección 2 **cuerpo**, reúne datos e información relacionada a la especificación los RNF, a los cuales agrupa como RNF de Calidad y RNF Restricciones. Está compuesto por las secciones **E, F** y **G**. La sección **E** reúne información relacionada a identificación y la descripción del RNF, la sección **F** contiene datos específicos del RNF y la sección **G** contiene datos relacionados a la validación de los RNF. Cada sección se describe en base a la siguiente estructura de presentación:

*Contenido:* En el Contenido se detallan los campos que forman parte de la sección, la presencia de un asterisco (\*) al lado de cada campo significa que es obligatorio ingresar los datos correspondientes.

*Consideraciones:* Se describen consideraciones particulares sobre el contenido de la sección, que deben tenerse en cuenta cuando se completa la plantilla.

#### **E) Identificación:**

##### **Contenido:**

- **Tipo:** < Tipo de Requerimiento No Funcional > (\*)
- **Categoría:** < Indica la característica de Calidad o la Categoría de Restricción a la que corresponde el RNF >. (\*)
- **Subcategoría:** < Indica la Subcaracterística de Calidad o la Subcategoría de Restricción a la que corresponde el RNF >. (\*)
- **Id. de Requerimiento:** < Identifica unívocamente al RNF la nomenclatura sugerida es <CodProy>-<NroSeq> Ej: PXX-001-0001 > (\*)
- **Descripción del requerimiento:** < Descripción del RNF > (\*)

### Consideraciones:

Ver las consideraciones de cada campo detalladas en el Patrón para Especificación de RNF, sección 9.2 (A) Identificación.

### F) Datos Específicos:

#### Contenido:

- **Id. Req. Asociado:** <Hace referencia al RNF al cual está asociado>.
- **Estabilidad:** <Hace referencia permanencia del RNF>.
- **Prioridad:** <Indica la prioridad del RNF, asignada por el/los Stakeholder/s>. (\*)
- **Versión #:** <Indica la versión de la Aplicación Web para la cual el RNF debe ser implementado>. (\*)
- **Documentación de Soporte/ Versión:** <Nombre y versión de la documentación de referencia>(\*)
- **Stakeholder:** <Rol del Stakeholder que originó el requerimiento> (\*)
- **Solicitante:** <Nombre de la Persona y/o Área > (\*)

### Consideraciones:

Ver las consideraciones de cada campo detalladas en el Patrón para Especificación de RNF, sección 7.5 (B) Datos Específicos.

### G) Validación:

#### Contenido:

- **Fecha Validación:** <Es una fecha>.
- **Criterio Aprobación:** <Descripción del/los criterios de aprobación del RNF>.

### Consideraciones:

Los campos de esta sección deben completarse cuando se lleve a cabo la validación de los RNF. El ingeniero de requerimientos deberá tener presente que:

*Fecha Validación:* Es la fecha en la que se valida formalmente el requerimiento con los Stakeholders involucrados.

*Criterio de aprobación:* Se describen la/ las condición/ es bajo las cuales el usuario dará la aceptación del requerimiento.

### 9.5 Estructura de la Matriz de RNF.

Las figuras 9.4 y 9.5 presentan la estructura de organización de la Matriz de RNF Web, con la finalidad de introducir las secciones que la componen.

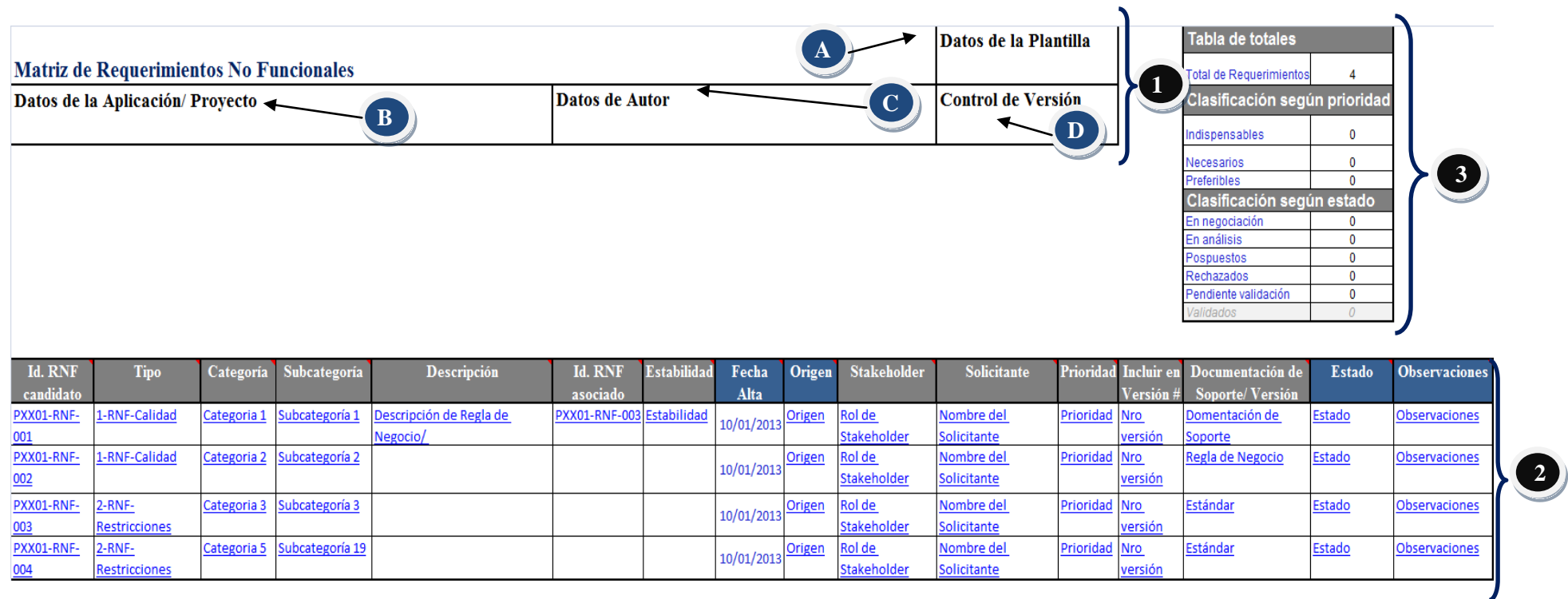


FIGURA 9.4: Matriz de RNF



4

FIGURA 9.5: Matriz de RNF – Gráfico de estado de Requerimientos

La sección 1 **encabezado**, reúne datos e información relacionada a:

- (A) Datos de la plantilla,
- (B) A la aplicación Web objetivo del proyecto,
- (C) Al Autor,
- (D) Al documento.

La sección 2 **cuerpo**, reúne datos e información relacionada a la elicitación y especificación los RNF de calidad y restricciones.

La sección 3 **tabla de totales**, muestra el total de requerimientos clasificados según la “Prioridad” y el “Estado”.

La sección 4 **gráfico**, presenta un gráfico de los RNF según su estado.

## 9.6 Descripción de la Matriz de RNF

La Matriz de RNF fue diseñada para dar soporte a los procesos de elicitación y especificación de RNF Web propuestos en este trabajo. La información que se registra en esta matriz durante el proceso de elicitación, permitirá dar origen a un conjunto de requerimientos



no funcionales candidatos, los cuales en el proceso de especificación serán analizados y descriptos correctamente, algunos RNF serán considerados como requerimientos definitivos y podrán conformar la *planilla de RNF pendiente de validación*, mientras que otros no serán considerados por diferentes razones. La matriz de RNF no solo es un nexo entre los dos procesos, sino que además reúne la información vinculada a RNF candidatos que fueron postergados o rechazados durante el proceso de IR con las observaciones que sustentan estas decisiones.

Dado que la Matriz de RNF brinda soporte a los procesos de Elicitación y Especificación, el cuerpo de la matriz contiene información propia de la aplicación de la IR, que no se traslada a la Planilla de especificación de RNF. Por ejemplo la *Fecha de Alta* y el *Origen* son relevantes durante la Elicitación. El *Estado* y *Observaciones*, reflejan la situación de cada RNF durante el proceso de Especificación y la interacción con el proceso de Elicitación.

La Figura 9.4 muestra la Matriz de RNF. La plantilla está disponible en el archivo Excel *Matriz de RNF -v1.0.xls* que complementa este trabajo. En la sección Anexo se describe como está organizado el archivo.

### 9.6.1 Encabezado de la Matriz de RNF

El **encabezado** se compone de las secciones: **A, B, C, D**. La sección **A**, tiene información relacionada a la plantilla, las secciones **B, C** y **D** contienen campos relacionados a la aplicación Web (objetivo del proyecto), al autor y al versionado del documento. Los campos de estas secciones se corresponden con los descritos en la Plantilla de Elicitación de RNF de Calidad, ver la explicación de la sección 8.3.1.

### 9.6.2 Cuerpo de la Matriz de RNF

La sección 2 **cuerpo**, reúne información relacionada a la elicitación y especificación los RNF de calidad y restricciones. La descripción de sus campos sigue la siguiente estructura de presentación:

**Contenido:** *En el Contenido se detallan los campos que forman parte de la sección, la presencia de un asterisco (\*) al lado de cada campo significa que es obligatorio ingresar los datos correspondientes.*

**Consideraciones:** *Se describen consideraciones particulares sobre el contenido de la sección, que deben tenerse en cuenta cuando se completa la plantilla.*

**Contenido:**

- **Id. de RNF Candidato:** *<Identifica unívocamente al RNF la nomenclatura sugerida es <CodProy>-<NroSeq> Ej: PXX-001-0001> (\*)*
- **Tipo:** *<Tipo de Requerimiento No Funcional> (\*)*
- **Categoría:** *<Indica la característica de Calidad o la Categoría de Restricción a la que corresponde el RNF>. (\*)*
- **Subcategoría:** *<Indica la Subcaracterística de Calidad o la Subcategoría de Restricción a la que corresponde el RNF>. (\*)*
- **Descripción del requerimiento:** *<Descripción del RNF> (\*)*
- **Id. Req. Asociado:** *< Hace referencia al RNF al cual está asociado>*
- **Estabilidad:** *< Hace referencia permanencia del RNF>.*
- **Fecha Alta:** *<Indica la fecha de solicitud del RNF>. (\*)*
- **Origen:** *<Indica de donde surge el requerimiento> (\*)*
- **Stakeholder:** *<Rol del Stakeholder que originó el requerimiento> (\*)*
- **Solicitante:** *<Nombre de la Persona y/o Área > (\*)*
- **Prioridad:** *<Indica la prioridad del RNF, asignada por el Stakeholder>. (\*)*
- **Incluir en versión #:** *<Indica la versión de la Aplicación Web para la cual el RNF debe ser implementado>. (\*)*
- **Documentación de Soporte/ Versión:** *<Nombre y versión de la documentación de referencia>(\*)*
- **Estado:** *<Indica la situación del requerimiento en el proceso de IR. >. (\*\*)*
- **Observaciones:** *<Indica cualquier información que se considere oportuna.>. (\*\*)*

### Consideraciones:

Para completar los campos de esta sección de la Matriz el ingeniero de requerimientos debe considerar lo siguiente:

- Durante el proceso de Elicitación debe ingresar: *Id. de RNF, candidato* (es equivalente el Id de RNF), *Tipo, Categoría, Subcategoría y Descripción del requerimiento*, siguiendo las consideraciones de cada campo detalladas en el Patrón para Especificación de RNF, sección 9.2 (A) Identificación.

Cabe acotar que durante la elicitación el campo Descripción del RNF, puede contener la regla de negocio, o una sentencia preliminar, que luego durante el proceso de especificación, se adecuará como RNF.

*Fecha Alta:* Hace referencia a fecha de solicitud del RNF, está relacionada a la fecha en la cual se realizó la recolección de información.

*Origen:* Indica de donde surge el requerimiento, los valores posibles son: Cliente, Interno (equipo de desarrollo), Otros.

- Durante el proceso de Especificación debe reformular la *Descripción del requerimiento*, debe ingresar el *Id. Req. Asociado y Estabilidad* (de los requerimientos que contengan estos atributos); *Prioridad, Versión, Documentación de Soporte/ Versión*, siguiendo las consideraciones de cada campo detalladas en el Patrón para Especificación de RNF, sección 9.2 (B) Datos Específicos.

*Estado:* Hace referencia a la situación del requerimiento en el proceso de IR. Los valores posibles son:

- **En negociación:** significa que el requerimiento fue relevado pero existe algún conflicto entre este y otros.
- **En análisis:** significa que el requerimiento fue relevado y se encuentra en estudio de factibilidad técnica.
- **Pospuesto:** significa que el requerimiento fue diferido. Informar razón en Campo Observaciones.
- **Rechazado:** significa que el requerimiento fue desestimado. Informar razón en campo Observaciones.
- **Pendiente de Validación:** Significa que el requerimiento fue especificado en la Plantilla de Especificación de RNFs y está pendiente de validación.

*Observaciones:* Indica cualquier información que se considere oportuna, asociada al estado del requerimiento “Rechazado”, “Pospuesto” o relevante para el RNF “Pendiente de validación”.

### 9.6.3 Tabla de totales de RNF

La sección 3 **tabla de totales**, resume el total de RNF identificados y muestra los mismos agrupados según “Prioridad” y “Estado”. Esta tabla se genera a partir de los datos disponibles en la Matriz de RNF, columnas Prioridad y Estado.

De acuerdo a la prioridad que definan los stakeholders durante el proceso de IR, los RNF pueden ser clasificados como: Indispensables, Necesarios, Preferibles. La tabla contiene dos columnas, una detalla las tres prioridades establecidas y la otra columna muestra la cantidad de RNF correspondientes a cada prioridad.

El estado de los RNF en la matriz, representa la situación del requerimiento en el proceso de IR. Los estados que propone la matriz son: En negociación, En análisis, Pospuestos, Rechazados, Pendiente de Validación, Validados. La tabla contiene dos columnas, una detalla los seis estados mencionados y la otra columna muestra la cantidad de RNF que se encuentran en ese estado.

### 9.6.4 Gráfico de estado de RNF

La sección 4 **gráfico**, es una representación gráfica del estado de los RNF, es generado a partir de los datos disponibles en la tabla de totales.

Este gráfico ofrece al ingeniero de requerimientos un resumen visual de la situación de los RNF durante el proceso de IR.

## Capítulo: 10

---

### 10.Caso de Estudio

La aplicación de los procesos de elicitación y especificación de RNF Web presentados en el capítulo 6 y las plantillas de soporte a los mismos, descritas los capítulos 8 y 9, son útiles cuando demuestra su validez en un caso concreto.

En este capítulo, como soporte al contenido teórico de la tesis, se describe la aplicación de los lineamientos y las plantillas definidas para la elicitación y la especificación de los Requerimientos No Funcionales (de calidad y restricciones) en un proyecto real, en el cual actualmente se está desarrollando una aplicación Web. Es importante mencionar que en este capítulo no se describen en detalle las actividades relacionadas a la captura y especificación de los requerimientos funcionales, tampoco se presentan el resto de fases del ciclo de vida involucradas en el desarrollo, ni la aplicación Web resultante, dado que los mismos no son parte de los objetivo de este trabajo.

El objetivo del caso de estudio es demostrar con un caso real, la aplicación de los lineamientos y herramientas definidos para el tratamiento de los Requerimientos No Funcionales (de calidad y restricciones) en los procesos de Elicitación y Especificación de la Ingeniería de Requerimientos Web.

#### 10.1 Presentación de Dominio.

La organización para la cual se desarrolla la aplicación web es un operador portuario Argentino, gestor de un tercio del volumen total del comercio exterior de contenedores en el país. Sus principales servicios son:

- *Terminal de Contenedores*: Cuenta con zonas de atraque para buques portacontenedores. Cuenta con una terminal ferropuertaria para combinar el ferrocarril con el transporte carretero y el transporte marítimo.
- *Carga General*: Atención de buques break bulk y carga de proyecto asignando un área dedicada al acopio, acondicionamiento y estiba de carga suelta.

- *Servicios a contenedores*: Para la Recepción, administración y entrega contenedores vacíos de las líneas marítimas que operan en el puerto. Además se reacondiciona y repara los contenedores secos y reefer, complementando así los requerimientos de IICL (Institute of International Container Lessors).
- *Depósito Fiscal*: Realiza servicios de consolidado, desconsolidado y almacenaje de mercadería de importación y exportación.

La aplicación web a desarrollar está relacionada al servicio de *Carga General*. Actualmente existen sistemas específicos y herramientas automatizadas que dan soporte a los procedimientos de trabajo llevados a cabo para producir este servicio; la gerencia de operaciones necesita disponer de una aplicación que utilice los datos generados durante la carga y descarga de barcos, para mostrar indicadores que permitan la toma de decisiones.

En la implementación de la aplicación Web participan: personas que pertenecen a la organización portuaria que tienen conocimiento y experiencia en el dominio de la aplicación, sector de sistemas y otras personas que tienen conocimiento de la infraestructura de hardware y software de la organización, más ingenieros de software que pertenecen a una empresa que brinda servicios de desarrollo de aplicaciones.

## **10.2 Aplicación de la IR Web para el tratamiento de los RNF**

Obtener RNF no es un proceso lineal, como se explicó en el capítulo 6, la Elicitación y la Especificación no son procesos aislados, los mismos interactúan entre sí, para la captura de conocimiento, análisis, descripción de RNF y negociaciones con los Stakeholders. Si bien en este capítulo describimos la aplicación de los procesos siguiendo la secuencia de actividades y tareas propuestas por cada proceso (Elicitación y Especificación), en la práctica real, la lista de RNF de la aplicación Web se obtuvo como parte de una serie de iteraciones entre los dos procesos.

Considerando las dimensiones del proyecto, el tipo de aplicación y la cantidad de Stakeholders involucrados, la captura del conocimiento relacionado a los RNF se realizó en paralelo con la adquisición del conocimiento vinculado a la funcionalidad requerida para la aplicación Web. El relevamiento se llevo a cabo mediante siete reuniones presenciales,

comunicación fluida por correo electrónico y conferencias telefónicas. En la reunión inicial con el gerente de sistemas de la organización (cliente) se ejecutaron las actividades 1.1 y 1.2 del proceso de elicitación, comunes para la captura de requerimientos funcionales y no funcionales; los datos e información obtenidos permitieron planificar las seis reuniones siguientes.

La figura 10.1 muestra el esquema de las reuniones realizadas como parte del proceso de obtención de los requerimientos de la aplicación Web y sus participantes.

Reuniones	Objetivos	Participantes
Reunión 1	Definir dominio de la aplicación, objetivos y stakeholders involucrados	Gerente de Sistemas Ingeniero de Software
Reunión 2	Capturar necesidades, deseos y expectativas de usuario / empresa ( Funcionalidad y restricciones de Proceso)	Gerente de Sistemas Gerente de Área de Carga Ingeniero de Software
Reunión 3	Capturar conocimiento asociado a los requerimientos funcionales.	Gerente de Área de Carga Subgerente de Área de Carga Empleado de Área de Carga Ingeniero de Software
Reunión 4	Adquirir conocimiento de RNF de Calidad	Gerente de Área de Carga Subgerente de Área de Carga Empleado de Área de Carga Referente técnico de Aplicaciones Ingeniero de Software
Reunión 5	Adquirir conocimiento de RNF restricciones	Gerente de Sistemas Referente técnico de Aplicaciones Referente técnico de Infraestructura Ingeniero de Software
Reunión 6	Negociación de RNF	Gerente de Sistemas Gerente de Área de Carga Referente técnico de Aplicaciones Ingeniero de Software
Reunión 7	Revisión de los RNF	Gerente de Sistemas Gerente de Área de Carga Referente técnico de Aplicaciones Ingeniero de Software

FIGURA 10.1: Esquema de reuniones de relevamiento de la aplicación Web

A continuación describiremos el detalle de las actividades llevadas a cabo por el ingeniero de requerimientos para la Elicitación de conocimiento vinculado a los RNF, las entradas consideradas y las salidas obtenidas; en la sección siguiente presentaremos el trabajo

realizado para escribir los RNF en base al patrón y los lineamientos definidos para la especificación de RNF y finalmente se exhibirá el estado general de la matriz de RNF generada durante la aplicación de los procesos (Elicitación y Especificación) y se presentará la Plantilla de RNF Web, salida del proceso de especificación, que reúne los requerimientos de calidad y restricciones identificados para la aplicación Web “Indicadores de Grúas y Barcos”.

Cabe acotar que si bien se listan los Requerimientos Funcionales obtenidos, no se describirán los detalles del proceso aplicado para la recolección de los mismos.

### **10.2.1 Aplicación del proceso de Elicitación de RNF**

#### **Entradas:**

- Clientes, Usuarios, Otros Stakeholders: Conformado por el personal del *sector de operaciones de carga y descarga* y el personal del *sector de sistemas* y personal de *infraestructura* de la organización portuaria.
- Requerimientos Funcionales: La figura 10.2 muestra el listado de los requerimientos funcionales de la aplicación Web “Indicadores de Grúas y Barcos”.
- Conocimiento del Dominio: Detallado en el sección presentación del dominio 10.1



Id. Req.	Lista de Requerimientos Funcionales
IGB01-RF-001	El sistema debe visualizar los indicadores Gross Vessel Rate (GVR) y Gross Crane Rate (GCR) de cada buque.
IGB01-RF-002	El sistema debe visualizar el indicador Gross Crane Rate (GCR) y el Ritmo de cada grúa.
IGB01-RF-003	El sistema debe visualizar el valor promedio de la cantidad de grúas que trabajaron en un barco por hora.
IGB01-RF-004	El sistema debe crear alarmas relacionadas al ritmo de cada grúa.
IGB01-RF-005	El sistema debe modificar alarmas relacionadas al ritmo de cada grúa.
IGB01-RF-006	El sistema debe eliminar alarmas relacionadas al ritmo de cada grúa.
IGB01-RF-007	El sistema debe calcular la productividad cada la grúa.
IGB01-RF-008	El sistema debe calcular la productividad proyectada y la productividad esperada de cada grúa, a partir de la productividad de la grúa.
IGB01-RF-009	El sistema debe enviar un mail a una "Lista de distribución" cuando el valor de movimientos por hora es igual al valor de movimientos por hora "rojo".
IGB01-RF-010	El sistema debe visualizar las horas de demoras de un buque agrupadas tipo de demora.
IGB01-RF-011	El sistema debe generar un evento de cierre de la jornada por cada buque.
IGB01-RF-012	El sistema debe emitir el informe "Resumen diario de los indicadores y total de movimientos por buque", después del evento de cierre de la jornada de cada buque.
IGB01-RF-013	El sistema debe emitir el informe "Resumen diario de movimientos por cada grúa asociada al buque cerrado", después del evento de cierre de la jornada de cada buque.
IGB01-RF-014	El sistema debe emitir el informe "Resumen de Productividad de las grúas asociadas al buque, agrupadas por fecha, hora y grúa", después del evento de cierre de la jornada de cada buque.
IGB01-RF-015	El sistema debe enviar un mail a una "Lista de distribución" con tres informes: 1. Resumen diario de los indicadores y total de movimientos por buque, 2. Resumen diario de movimientos por cada grúa asociada al buque cerrado. 3. Resumen de productividad de las grúas asociadas al buque, agrupadas por fecha, hora y grúa.
IGB01-RF-016	El sistema debe guardar información histórica del resumen de productividad de las grúas asociadas al buque, agrupadas por fecha, hora y grúa.

FIGURA 10.2: Listado de Requerimientos Funcionales de la aplicación Web

En la ejecución del proceso de Elicitación de RNF, el ingeniero de requerimientos siguió los lineamientos presentados en el capítulo 6, si bien en el desarrollo del proyecto se relevaron los requerimientos de calidad y restricciones para toda la aplicación Web, para evitar sumar complejidad en la demostración optamos por presentar las actividades y tareas realizadas con ejemplos acotados a la definición de requerimientos específicos y complementar las mismas con los resultados finales obtenidos como consecuencia de su aplicación, para cada tarea indicamos también el momento en que fue realizada de acuerdo al plan de reuniones llevadas a cabo durante el relevamiento.

## Actividades:

### 1. La identificación de todas las fuentes de conocimiento de requerimientos:

#### 1.1 Definir dominio de la aplicación y objetivos:

Quedaron establecidos durante la Reunión 1, en la cual participaron el Gerente de Sistemas y el Ingeniero de requerimientos. La figura 10.3 resume el dominio y el objetivo.

<i>Dominio:</i> Organización portuaria Argentina / Servicio de Carga General.
<i>Objetivo :</i> Disponer de una aplicación Web que utilice los datos generados durante la carga y descarga de barcos, para mostrar indicadores que permitan la toma de decisiones.

FIGURA 10.3: Dominio y objetivos de la aplicación Web

Cabe acotar que actualmente no hay manuales de proceso documentados en el dominio de la aplicación, existen sistemas específicos y herramientas automatizadas que dan soporte a solo a una parte del trabajo que realizan las personas encargadas de producir el servicio de carga general.

No existe una versión anterior de la aplicación Web, ni aplicativo similar en el dominio.

#### 1.2 Identificar y clasificar Stakeholders claves:

Quedaron establecidos durante la Reunión 1, en la cual participaron el Gerente de Sistemas y el Ingeniero de requerimientos. Los Stakeholders claves son:

- Gerente de Sistemas
- Gerente de Área de Carga
- Subgerente de Área de Carga
- Empleado de Área de Carga
- Referente técnico de Aplicaciones
- Referente técnico de Infraestructura

Nota: La descripción de los usuarios potenciales, realizada previamente para obtener los stakeholders claves puede consultarse en la sección Anexo.

#### 1.3 Revisar los Requerimientos Funcionales identificados:

Los RF se definieron durante las reuniones 2 y 3, en la figura 10.1 podemos ver quienes participaron en la definición de los mismos. Los datos obtenidos en la reunión 2 dieron origen a la versión preliminar de los RF, mientras que los datos obtenidos en la reunión 3

permitieron escribir el listado de RF de la figura 10.2. El Ingeniero de requerimientos analizó cada uno de los RF para identificar su relación con las categorías y subcategorías de RNF. La Figura 10.4 muestra tres ejemplos de RF: 001, 002, 003, para los cuales se deberá capturar el conocimiento del dominio vinculado a las categorías de calidad seleccionadas.

<b>Id. Req.</b>	<b>Lista de Requerimientos Funcionales</b>	<b>Tipo</b>	<b>Categoría</b>
IGB01-RF-001	El sistema debe visualizar los indicadores Gross Vessel Rate (GVR) y Gross Crane Rate (GCR) de cada buque.	Calidad	Funcionalidad/Precisión Usabilidad/Atractividad
IGB01-RF-002	El sistema debe visualizar el indicador Gross Crane Rate (GCR) y el Ritmo de cada grúa.	Calidad	Funcionalidad/Precisión Usabilidad/Atractividad
IGB01-RF-003	El sistema debe visualizar el valor promedio de la cantidad de grúas que trabajaron en un barco por hora.	Calidad	Funcionalidad/Precisión Usabilidad/Atractividad

FIGURA 10.4: RF y su relación con tipos de RNF

*1.4 Aplicar los Checklist de RNF de Calidad y Restricciones:*

Esta tarea se divide en tres instancias, que se llevaron a cabo en forma previa a las reuniones 4 y 5.

*a) Mapear los stakeholders:*

El Ingeniero de requerimientos analizó los Stakeholder Claves, para establecer cuál es el rol que ocupan dentro de las categorías de Stakeholders de las plantillas de elicitación de RNF. La Figura 10.5 muestra el resultado de esta tarea.

<b>Stakeholder Clave</b>	<b>Nombre y Apellido</b>	<b>Tipo de Stakeholder de las Plantillas de Elicitación de RNF</b>	<b>Rol de las Plantillas de Elicitación de RNF</b>
Gerente de Sistemas	MC	Cliente	Manager
Gerente de Área de Carga	PS	Usuario	Usuario Operador
Subgerente de Área de Carga	SD	Usuario	Usuario Operador
Empleado de Área de Carga	MF	Usuario	Usuario Operador
		Otros Stakeholders	Experto en la Materia
Referente técnico de Aplicaciones	MG	Otros Stakeholders	Experto en la Materia
Referente técnico de Aplicaciones	AL	Otros Stakeholders	Experto Técnico
		Otros Stakeholders	Inspector
		Otros Stakeholders	Operador de mantenimiento
Referente técnico de Infraestructura	LP	Otros Stakeholders	Operador de mantenimiento

FIGURA 10.5: Rol de los Stakeholders claves en las Plantillas de Elicitación de RNF

*b) Seleccionar las características / subcaracterísticas de RNF de Calidad:*

De las 11 Subcaracterísticas que propone la Plantilla de Elicitación de RNF de Calidad, solamente, 9 fueron consideradas como significativas para el dominio de la aplicación Web.

Las siguientes no fueron seleccionadas.

- Funcionalidad/Interoperabilidad
- Portabilidad/ Adaptabilidad

La figura 10.6, muestra las características seleccionadas. La Plantilla de Elicitación de RNF de Calidad que se completó para el caso de estudio está disponible en el archivo *1. Plantilla Elicitacion RNF de Calidad- Caso v1.0.xls* que complementa este trabajo.

Característica	Subcaracterística	¿Aplica?
<a href="#">Funcionalidad</a> <a href="#">(Functionality)</a>	<a href="#">Idoneidad</a> <a href="#">Suitability</a>	<input checked="" type="checkbox"/>
	<a href="#">Precisión</a> <a href="#">Accuracy</a>	<input checked="" type="checkbox"/>
	<a href="#">Interoperabilidad</a> <a href="#">Interoperability</a>	<input type="checkbox"/>
	<a href="#">Seguridad</a> <a href="#">Security</a>	<input checked="" type="checkbox"/>
<a href="#">Confiabilidad</a> <a href="#">Reliability</a>	<a href="#">Madurez</a> <a href="#">Maturity</a>	<input checked="" type="checkbox"/>
	<a href="#">Tolerancia a fallas</a> <a href="#">Fault tolerance</a>	<input checked="" type="checkbox"/>
	<a href="#">Recuperabilidad</a> <a href="#">Recoverability</a>	<input checked="" type="checkbox"/>
<a href="#">Usabilidad</a> <a href="#">Usability</a>	<a href="#">Comprensibilidad</a> <a href="#">Understandability</a>	<input checked="" type="checkbox"/>
	<a href="#">Atractividad</a> <a href="#">Attractiveness</a>	<input checked="" type="checkbox"/>
<a href="#">Eficiencia</a> <a href="#">Efficiency</a>	<a href="#">Tiempo de comportamiento</a> <a href="#">Time behaviour</a>	<input checked="" type="checkbox"/>
<a href="#">Portabilidad</a> <a href="#">Portability</a>	<a href="#">Adaptabilidad</a> <a href="#">Adaptability</a>	<input type="checkbox"/>
	<a href="#">Capacidad de instalación</a> <a href="#">Installability</a>	<input checked="" type="checkbox"/>

FIGURA 10.6: Elicitación de RNF de Calidad - Características seleccionadas

c) *Seleccionar características / subcaracterísticas de RNF Restricciones:*

De las 14 Subcategorías que propone la Plantilla de Elicitación de RNF Restricciones 13 fueron consideradas como significativas para el dominio de la aplicación Web.

- **Arquitectura/ Sistemas Externos.** No fue seleccionada.

La figura 10.7, muestra las categorías seleccionadas.

La Plantilla de Elicitación de RNF Restricciones que se completó para el caso de estudio está disponible en el archivo *2. Plantilla Elicitacion RNF Restricciones- Caso v-1.0.xls*, que complementa este trabajo.

Categoría	Subcategoría	¿Aplica?
<a href="#">Proceso de desarrollo</a>	<a href="#">Alcance</a>	<input checked="" type="checkbox"/>
	<a href="#">Nivel de detalle</a>	<input checked="" type="checkbox"/>
<a href="#">Lenguaje</a>	<a href="#">Lenguajes de programación</a>	<input checked="" type="checkbox"/>
<a href="#">Ambiente</a>	<a href="#">Implementación</a>	<input checked="" type="checkbox"/>
	<a href="#">Sistema Operativo</a>	<input checked="" type="checkbox"/>
	<a href="#">Base de datos</a>	<input checked="" type="checkbox"/>
	<a href="#">Servidor de Aplicaciones</a>	<input checked="" type="checkbox"/>
<a href="#">Arquitectura</a>	<a href="#">Framework</a>	<input checked="" type="checkbox"/>
	<a href="#">Diseño de Capas</a>	<input checked="" type="checkbox"/>
	<a href="#">Sistemas Externos</a>	<input type="checkbox"/>
	<a href="#">Solicitudes de Mails / Mensajería</a>	<input checked="" type="checkbox"/>
	<a href="#">Accesibilidad</a>	<input checked="" type="checkbox"/>
	<a href="#">Librerías / Componentes</a>	<input checked="" type="checkbox"/>
<a href="#">Navegador</a>	<a href="#">Navegadores</a>	<input checked="" type="checkbox"/>

FIGURA 10.7: Elicitación de RNF - Restricciones

1.5 *Preparar entrevistas y realizar cuestionarios:*

Se planificaron 3 entrevistas, según se detalla a continuación:

**Entrevista 2:**

- **Objetivo:** Capturar Restricciones de Proceso
- **Roles participantes:**

Stakeholder Clave	Nombre y Apellido	Tipo de Stakeholder de las Plantillas de Elicitación de RNF	Rol de las Plantillas de Elicitación de RNF
Gerente de Sistemas	MC	Cliente	Manager

- **Preguntas:** Para realizar la captura de las restricciones de proceso de desarrollo se realizaron las preguntas de la Categoría *Proceso de Desarrollo* y sus subcategorías *Alcance y Nivel de Detalle*.

**Entrevista 4:**

- **Objetivo:** Adquirir conocimiento de RNF de Calidad.
- **Roles participantes:**

Stakeholder Clave	Nombre y Apellido	Tipo de Stakeholder de las Plantillas de Elicitación de RNF	Rol de las Plantillas de Elicitación de RNF
Gerente de Sistemas	MC	Cliente	Manager
Gerente de Área de Carga	PS	Usuario	Usuario Operador
Subgerente de Área de Carga	SD	Usuario	Usuario Operador
Empleado de Área de Carga	MF	Usuario	Usuario Operador
		Otros Stakeholders	Experto en la Materia

- **Preguntas:** En la captura de conocimiento relacionado con las características de calidad se realizaron preguntas extraídas de las subcaracterísticas seleccionadas previamente (ver figura 10.6).

Las preguntas con orientación técnica se realizaron mediante un cuestionario enviado por correo electrónico a los stakeholders correspondientes.

**Entrevista 5:**

- **Objetivo:** Adquirir conocimiento de RNF Restricciones.
- **Roles participantes:**

Stakeholder Clave	Nombre y Apellido	Tipo de Stakeholder de las Plantillas de Elicitación de RNF	Rol de las Plantillas de Elicitación de RNF
Gerente de Sistemas	MC	Cliente	Manager
Referente técnico de Aplicaciones	AL	Otros Stakeholders	Experto Técnico
		Otros Stakeholders	Inspector
		Otros Stakeholders	Operador de mantenimiento
Referente técnico de Infraestructura	LP	Otros Stakeholders	Operador de mantenimiento

- **Preguntas:** En la captura de conocimiento relacionado con las categorías de restricciones se realizaron preguntas extraídas de las subcategorías seleccionadas previamente (ver figura 10.7).

Algunas preguntas, no pudieron responderse durante la entrevista, se envió un cuestionario con las mismas por correo electrónico a los stakeholders correspondientes.

### ***Cuestionarios:***

Se enviaron dos cuestionarios, uno con preguntas de las características de calidad y el otro con preguntas de las categorías de restricciones.

Por ejemplo preguntas relacionadas con *Acceso Seguro*. (*Subcaracterística de Seguridad*) se enviaron por correo electrónico al *Operador de Mantenimiento*

#### *2. Acceso seguro:*

*2.1 ¿Necesita que la aplicación web respete una política de control de acceso corporativa? En caso de que no exista definir:*

*a) Reglas para las claves de usuario (contraseñas débiles)*

*b) Reglas para la caducidad y cambio de la clave de usuario.*

*c) Uso de algoritmos de encriptación de claves.*

*2.2 ¿Necesita que la aplicación web registre el historial de los accesos de usuario a fin de detectar posibles desviaciones de la política de control de acceso y uso?*

*2.3 ¿Se requiere una regla para la eliminación de cookies/ credenciales de autenticación? Definir periodicidad.*

*2.4 ¿Necesita que una cookie / credencial de autenticación se asocie a una dirección IP?*

### ***2. Adquirir el conocimiento:***

#### *2.1. Realizar las entrevistas y cuestionarios:*

El ingeniero de requerimientos realizó las entrevistas en las fechas acordadas. Los cuestionarios se enviaron por correo electrónico.

**2.2. Consolidar la información:**

El ingeniero de requerimientos analizó y clasificó los datos recolectados en las entrevistas 4 y 5 y en las respuestas de los cuestionarios

- Con los datos de usuarios autorizados se completo la "Matriz de Control de Autorización de Accesos". La misma se encuentra disponible en la sección Anexo.
- Durante las entrevistas se realizó un prototipo en papel de la interface gráfica para los dispositivos celulares, quedo pendiente el prototipo para la interface para las notebooks y Pcs.
- Se acordó que los prototipos definitivos se enviarían por correo electrónico. Ver ejemplo de prototipo en la sección Anexo.

Como parte del análisis se identificaron inconsistencias entre las algunas respuestas de los stakeholders, se anotaron las mismas a fin de unificar criterios en una iteración siguiente.

**3. Decidir sobre la relevancia del conocimiento para el dominio del problema:**

*Completar la Matriz de RNF:*

Los datos clasificados y analizados se escribieron en Matriz de RNF (versión 1.0), conformando un total de cuarenta RNF candidatos.

Tabla de totales	
Total de RNF Candidatos	40

La figura 10.8 muestra el resumen de los RNF Candidatos de calidad y restricciones respectivamente. La Matriz de RNF que se completó en esta actividad para el caso de estudio está disponible en el archivo *3. Matriz de RNF- Caso-v1.0.xls* que complementa este trabajo.



Tipo	Categoría	Subcategoría	Id. RNF candida	Total	
1-RNF-Calidad	Confiabilidad	Madurez	PXX01-RNF-014	1	
		Recuperabilidad	PXX01-RNF-016	1	
		Tolerancia a fallas	PXX01-RNF-015	1	
	<b>Total Confiabilidad</b>				<b>3</b>
	Eficiencia	Tiempo de comportamiento		PXX01-RNF-021	1
				PXX01-RNF-022	1
	<b>Total Eficiencia</b>				<b>2</b>
	Funcionalidad	Precisión		PXX01-RNF-005	1
				PXX01-RNF-006	1
				PXX01-RNF-007	1
				PXX01-RNF-008	1
				PXX01-RNF-009	1
		Seguridad		PXX01-RNF-010	1
				PXX01-RNF-011	1
				PXX01-RNF-012	1
				PXX01-RNF-013	1
	<b>Total Funcionalidad</b>				<b>9</b>
	Portabilidad	Capacidad de instalación		PXX01-RNF-023	1
				PXX01-RNF-024	1
	<b>Total Portabilidad</b>				<b>2</b>
Usabilidad	Atractividad		PXX01-RNF-018	1	
			PXX01-RNF-019	1	
			PXX01-RNF-020	1	
		Comprensibilidad	PXX01-RNF-017	1	
<b>Total Usabilidad</b>				<b>4</b>	
<b>Total 1-RNF-Calidad</b>				<b>20</b>	
2-RNF-Restricción	Ambiente	Base de datos		PXX01-RNF-032	1
				PXX01-RNF-026	1
		Implementación		PXX01-RNF-027	1
				PXX01-RNF-028	1
				PXX01-RNF-029	1
				PXX01-RNF-033	1
		Servidor de Aplicaciones	PXX01-RNF-030	1	
		Sistema Operativo	PXX01-RNF-031	1	
	<b>Total Ambiente</b>				<b>8</b>
	Arquitectura	Accesibilidad		PXX01-RNF-037	1
				PXX01-RNF-038	1
		Diseño de Capas	PXX01-RNF-035	1	
		Framework	PXX01-RNF-034	1	
		Librerías / Componentes	PXX01-RNF-039	1	
		Navegador	PXX01-RNF-040	1	
	Solicitudes de Mails / Mensajería	PXX01-RNF-036	1		
	<b>Total Arquitectura</b>				<b>7</b>
	Lenguaje	De programación	PXX01-RNF-025	1	
	<b>Total Lenguaje</b>				<b>1</b>
	Proceso de desarrol	Alcance		PXX01-RNF-001	1
			PXX01-RNF-002	1	
Nivel de detalle			PXX01-RNF-003	1	
			PXX01-RNF-004	1	
<b>Total Proceso de desarrollo</b>				<b>4</b>	
<b>Total 2-RNF-Restricciones</b>				<b>20</b>	
<b>Total general</b>				<b>40</b>	

FIGURA 10.8: RNF Candidatos (Matriz de RNF salida de Elicitación)

**4. Comprender la importancia del conocimiento elicitado y su impacto sobre los requerimientos del software.**

**4.1. Conciliar las necesidades de los Stakeholders:**

Las diferencias que se identificaron entre los RNF candidatos, quedaron registradas en la columna “Observaciones” de la Matriz de RNF versión 1.0. Fue necesario conciliar las mismas con los stakeholders mediante conferencias telefónicas y correos electrónicos, los RNF definitivos se describieron durante el proceso de Especificación. A continuación se muestra un ejemplo particular de inconsistencia entre un RNF de calidad *Categoría Funcionalidad/ Subcategoría Precisión* PXX01-RNF-009 y dos RNF de la *Categoría Usabilidad/ Subcategoría Atractividad*, PXX01-RNF-018 y PXX01-RNF-019.

El RNF candidato dice:

Id. RNF candidato	Descripción	Observaciones
PXX01-RNF-009	Formato para mostrar la hora acordado es: <b>00:00:00hs.</b>	El formato de la hora es diferente en el prototipo recibido que muestra <b>00:00hs.</b> Ver Prototipo asociado a los RNF RNF 018 y 019.

Prototipo asociado a **PXX01-RNF-018** y **PXX01-RNF-019**, muestra:

<b>BUQUE 2</b>	
GVR = 65	CI = 3
FPLP = 55	
P1 GCR = 21	Ritmo 20
P2 GCR = 14	Ritmo 21
P3 GCR = 16	Ritmo 18
P4 GCR = 14	Ritmo 20
Demoras = 1.5 hs +	
Mov. Realizados	1640
Mov. Restantes	405
Hora de Inicio	14:35hs
ETC estimada	21:50hs

#### 4.2. Identificar relaciones entre necesidades de Calidad y las Restricciones:

Las relaciones entre RNF candidatos de calidad y restricciones, se reflejaron en la columna “Id. RNF asociado” de la *Matriz de RNF versión 1.0.xls*.

A continuación la figura 10.9 muestra un ejemplo particular que muestra la relación entre la necesidad de calidad de la *Categoría Usabilidad* PXX01-RNF-019 con dos RNF Restricciones de las *Categorías Ambiente y Arquitectura* PXX01-RNF-026 y PXX01-RNF-040.

Id. RNF candidato	Tipo	Categoría	Descripción	Id. RNF asociado
PXX01-RNF-019	1-RNF-Calidad	Usabilidad	En la versión para PC, se incluirán íconos de barcos y grúas. Ver prototipo.	PXX01-RNF-026 PXX01-RNF-040
PXX01-RNF-026	2-RNF-Restricciones	Ambiente	La aplicación web será desarrollada para PC y notebook.	PXX01-RNF-019 PXX01-RNF-020
PXX01-RNF-040	2-RNF-Restricciones	Arquitectura	Compatibilidad con los browsers google chrome e IE 8 o superior.	PXX01-RNF-019 PXX01-RNF-020

FIGURA 10.9: Relaciones entre RNF Candidatos

#### Salida:

- La Matriz de RNF que se generó para el caso de estudio, en base a las actividades precedentes, está disponible en el archivo *3. Matriz de RNF- Caso-v1.0.xls* que complementa este trabajo.

## 10.2.2 Aplicación del proceso de Especificación de RNF

#### Entradas:

- Clientes, Usuarios, Otros Stakeholders.
- Matriz de RNF versión 1.0 (Salida del proceso de Elicitación).

#### Actividades:

### 1. Análisis del conocimiento asociado a los requerimientos No Funcionales:

#### 1.1. Derivar los RNF de Calidad y Restricciones:

Los RNF se escribieron de acuerdo a los lineamientos del patrón para especificación de RNF que propone esta tesis. Algunos RNF candidatos dieron origen a nuevos RNF. Se escribieron un total de 44 RNF, para lo cual se realizaron las siguientes actividades.

a) *Identificar la estabilidad de RNF de Calidad y Restricciones:*

Se identificó la estabilidad para 33 de los 44 RNF, estableciendo 26 de estabilidad Alta y 7 de estabilidad Baja. Quedaron 11 RNF para los cuales no es obligatorio este atributo.

b) *Identificar la prioridad de RNF de Calidad y Restricciones:*

La priorización de los RNF candidatos quedó establecida según el cuadro siguiente

Clasificación según prioridad	
Indispensables	33
Necesarios	10
Preferibles	1

c) *Identificar la versión y documentación de soporte:*

Si bien al inicio de la especificación los 44 RNF candidatos se solicitaron para la versión 1.0, el resultado de la interacción entre los procesos de Elicitación y Especificación finalmente reflejó que 36 RNF, se realizarán en la versión 1.0. Se completo la columna “Documentación de soporte” para permitir la trazabilidad de los RNF con su origen.

1.2. *Identificar conflictos entre los RNF de Calidad y Restricciones:*

Los conflictos entre los RNF de calidad y restricciones se registraron en la Matriz de RNF versión 1.1, en la columna estado con el valor “En negociación” y los comentarios correspondientes se agregaron en la columna “Observaciones”.

Por ejemplo para la pregunta relacionada con *Acceso Seguro (Subcaracterística de Seguridad)*

2.1 *¿Necesita que la aplicación web respete una política de control de acceso corporativa?*

Se obtuvieron respuestas y necesidades diferentes:

- a) Un *usuario operador*, quiere un control de acceso independiente a la política control de acceso corporativo y describió sus propias reglas.

- b) El *manager*, respondió que debe respetar la política de control de acceso corporativa.
- c) El *operador de mantenimiento*, respondió que debe adecuar la política de control de acceso corporativa.

que dieron origen a tres RNF candidatos diferentes: PXX01-RNF-015, PXX01-RNF-016 y PXX01-RNF-017, los cuales están en conflicto entre sí, a su vez con el RNF de la *Categoría Arquitectura/ Subcategoría/accesibilidad* PXX01-RNF-042 que solicita utilizar LDAP (disponible en la organización).

En la Matriz de RNF versión 1.1 puede verse el resumen de la clasificación de RNF candidatos por estado que muestra cantidad de RNF “En negociación”:

Clasificación según estado	
En negociación	7

1.3. *Identificar si los RNF requieren análisis de factibilidad técnica:*

Se identificaron RNF que necesitan análisis de factibilidad técnica, los cuales se registraron en la Matriz de RNF versión 1.1, en la columna estado con el valor “En Análisis” y los comentarios correspondientes se agregaron en la columna “Observaciones”.

Por ejemplo, se realizó un prototipo de arquitectura en HTML 5, en base al diseño para celulares del RNF PXX01-RNF-022, a fin de determinar si será necesario customizar el desarrollo para cada modelo de celular requerido por los stakeholders en el RNF o se podrá contar con un diseño genérico en HTML 5.

En la Matriz de RNF versión 1.1 puede verse el resumen de la clasificación de RNF candidatos por estado que muestra cantidad de RNF “En Análisis”:

Clasificación según estado	
En análisis	9

1.4 *Identificar los RNF postergados:*

Como producto de las iteraciones entre los procesos de elicitación y especificación se decidió postergar algunos requerimientos. Estos se registraron en la Matriz de RNF versión 1.2, en la columna estado con el valor “Pospuesto” y los comentarios correspondientes se agregaron en la columna “Observaciones”.

La figura 8.10 muestra dos ejemplos de RNF de la *Característica/ Usabilidad, Subcaracterística/Atractividad* PXX01-RNF-023 y PXX01-RNF-024 que pasaron del estado “En Analisis” al estado “Pospuesto”, por razones de presupuesto disponible. Esta decisión se tomó en la reunión 6, que se llevo a cabo para negociación de RNF.

Id. RNF candidato	Descripción	Estado	Observaciones
PXX01-RNF-023	La interface gráfica de la aplicación para pc debe seguir los lineamientos establecidos en el prototipo de diseño versión 2.1. Los indicadores se visualizan en forma plana (Números, letras, se incluirán íconos de barcos y grúas)	3-Pospuesto	24/01/2013: Se posterga el RNF por razones de presupuesto. Será desarrollado en la versión 1.1 de la aplicación.
PXX01-RNF-024	La aplicación debe generar un gráfico de líneas sobre actividad de un "Pórtico" en base a los lineamientos establecidos en el prototipo de diseño versión 3.1.	3-Pospuesto	24/01/2013: Se posterga el RNF por razones de presupuesto. Será desarrollado en la versión 1.1 de la aplicación.

FIGURA 10.10: RNF Candidatos - Pospuestos

En la Matriz de RNF versión 1.2 puede verse el resumen de la clasificación de RNF candidatos por estado que muestra cantidad de RNF “Pospuestos”

Clasificación según estado	
Pospuestos	4

1.5 *Identificar los RNF rechazados:*

En la reunión 6, producto de la negociación, se decidió rechazar algunos requerimientos. Estos se registraron en la Matriz de RNF versión 1.2, en la columna estado con el valor

“Rechazado” y los comentarios correspondientes se agregaron en la columna “Observaciones”.

La figura 8.11 muestra un ejemplo de RNF de la *Característica/ Funcionalidad, Subcaracterística/Seguridad* el requerimiento PXX01-RNF-015 pasó del estado “En Negociación” al estado “Rechazado”, debido a que se decidió implementar otro requerimiento en su lugar, el PXX01-RNF-016 que muestra la figura.

Id. RNF candidato	Descripción	Estado	Observaciones
PXX01-RNF-015	La aplicación debe implementar las reglas de acceso seguro: a) Reglas para las claves de usuario (contraseñas débiles). - Las claves deben ser de 8 caracteres. - Las claves deben tener letras y números. Una de las letras mayúscula. - La clave no debe permitir las palabras grua, buque, ritmo.  b) Reglas para la caducidad y cambio de la clave de usuario. - El cambio de clave debe ser bimestral. - Controlar que la clave nueva sea diferente de las cuatro últimas claves.  c) Uso de algoritmos de encriptación de claves. - Se debe utilizar el algoritmo de encriptación disponible en la organización.	4-Rechazado	Estaba en conflicto con los RNF 16, 17 y 42. El 24/01/2013 se decidió rechazar este RNF, en su lugar se implementará el RNF15.
PXX01-RNF-016	La aplicación web debe respetar la política de control de acceso seguro que especifica el "Estándar de control de acceso corporativo v.1.1"	5-Pendiente Validación	

FIGURA 10.11: RNF Candidato - Rechazado

En la Matriz de RNF versión 1.2 puede verse el resumen de la clasificación de RNF candidatos por estado que muestra cantidad de RNF “Rechazados”

Clasificación según estado	
Rechazados	4

1.6. *Actualizar la Matriz de RNF*: La matriz de RNF, es la herramienta que brindó soporte a la especificación, en la ejecución de este proceso se generaron las versiones 1.1 y 1.2 de matriz de RNF. Las figura 10.12 y 10.13 muestran el gráfico de estado de los RNF en dos momentos diferentes.

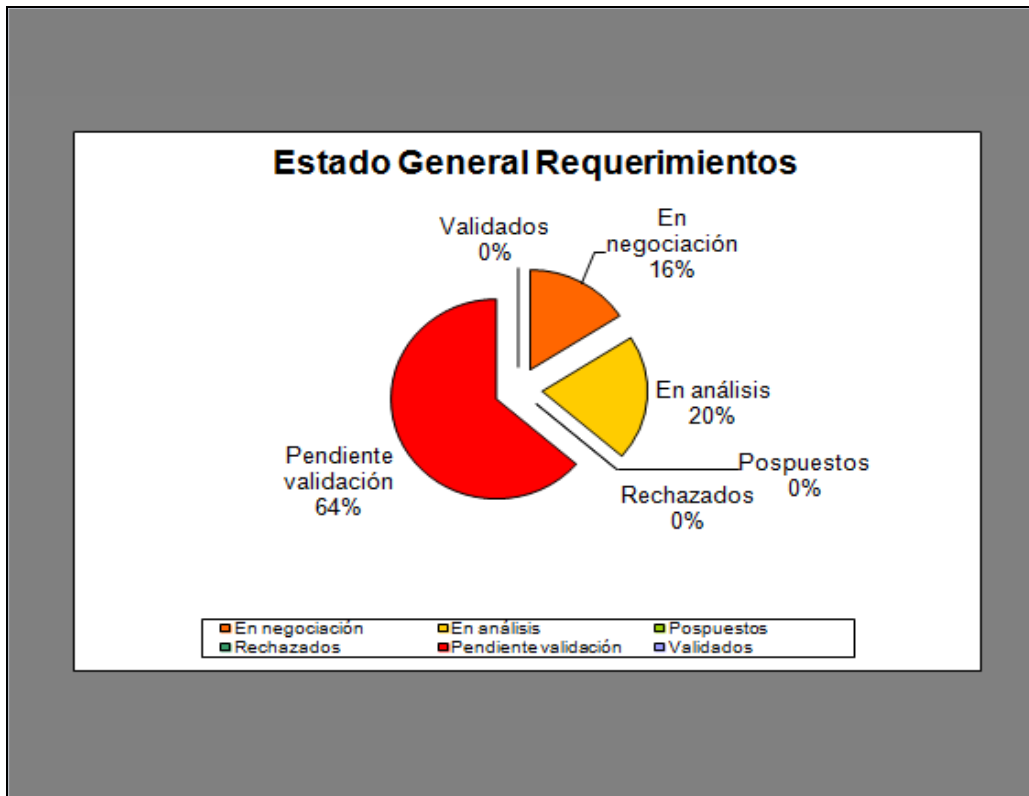


FIGURA 10.12: Gráfico de estado de RNF – Inicio especificación



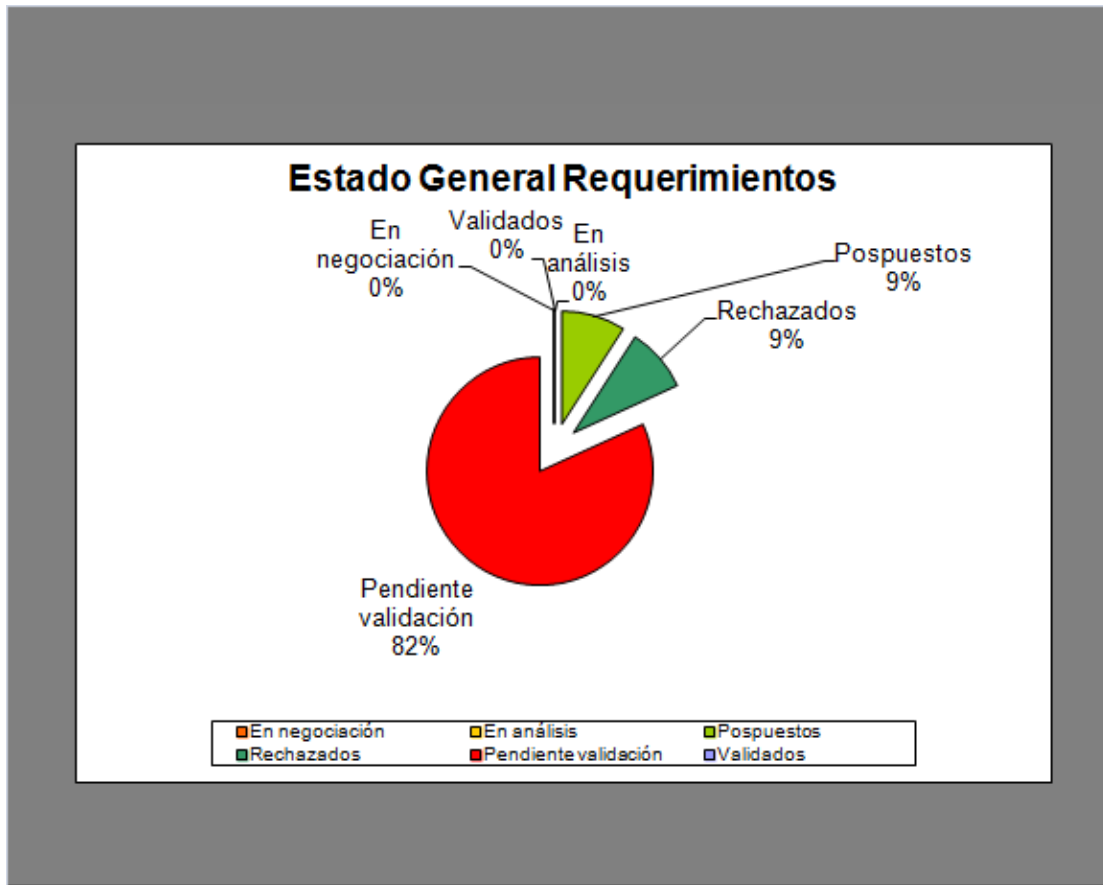


FIGURA 10.13: Gráfico de estado de RNF – Fin de especificación

**2. Organizar y documentar los Requerimientos No Funcionales:** Esta actividad se realizaron las siguientes tareas.

*2.1. Controlar que los RNFs identificados estén en condiciones de ser validados:* El ingeniero de requerimientos revisó todos los RNF de la Matriz que da soporte al proceso, se aseguró que los RNF de la Matriz tengan los estados: Pospuesto, Rechazado, Pendiente de Validación. Lo cual implica que aquellos RNF en estado “En análisis” o “En negociación”, fueron resueltos en iteraciones con el proceso de Elicitación previamente.

En la Matriz de RNF versión 1.2 puede verse el resumen de la clasificación de RNF candidatos por estado, que se reflejan en la figura 10.14.

Clasificación según estado	
En negociación	0
En análisis	0
Pospuestos	4
Rechazados	4
Pendiente validación	36

FIGURA 10.14: Clasificación de RNF según su estado – Fin de especificación

2.2. *Elaborar la especificación de Requerimientos No Funcionales*: El ingeniero de requerimientos completó la *Plantilla de especificación de requerimientos no funcionales Web* que propone este trabajo, solamente se consideraron los 36 RNFs que quedaron con estado “Pendiente de Validación” en la Matriz de RNF versión 1.2.

En la reunión 7, se realizó una revisión final de los RNF en conjunto con los stakeholders referentes en el proyecto, a fin de repasar los motivos por los cuales se rechazaron o postergaron requerimientos y revisar la prioridad establecida para cada uno de los 36 RNF que se volcaron a la Plantilla de especificación de RNF Web 1.0, dado que esta Plantilla, salida del proceso de especificación, será de entrada a la **Validación**, último proceso de la IR.

#### Salida:

- La Plantilla de especificación de requerimientos no funcionales Web 1.0 que se generó para el caso de estudio, en base a las actividades precedentes, está disponible en el archivo 6. *Plantilla de Especificacion RNF Web-v1.0.xls* que complementa este trabajo.

### 10.3 Resultados

La figura 10.15 refleja una vista de los 36 RNF identificados y especificados para la aplicación Web “Indicadores de Grúas y Barcos”, los cuales deben ser validados por los stakeholders en el proceso de Validación de la IR.

Tipo	Categoría	Subcategoría	Id. Requerimiento	Prioridad	Total	
1-RNF-Calidad	Confiabilidad	Madurez	PXX01-RNF-018	2-Necesario	1	
		Recuperabilidad	PXX01-RNF-020	1-Indispensable	1	
		Tolerancia a fallas	PXX01-RNF-019	1-Indispensable	1	
	Total Confiabilidad					3
	Eficiencia	Tiempo de comportamiento		PXX01-RNF-025	2-Necesario	1
				PXX01-RNF-026	1-Indispensable	1
	Total Eficiencia					2
	Funcionalidad	Precisión		PXX01-RNF-009	1-Indispensable	1
				PXX01-RNF-010	1-Indispensable	1
				PXX01-RNF-011	1-Indispensable	1
				PXX01-RNF-012	1-Indispensable	1
				PXX01-RNF-013	1-Indispensable	1
			Seguridad	PXX01-RNF-014	1-Indispensable	1
			PXX01-RNF-016	2-Necesario	1	
	Total Funcionalidad					7
Portabilidad	Capacidad de instalación		PXX01-RNF-028	2-Necesario	1	
Total Portabilidad					1	
Usabilidad	Atractividad		PXX01-RNF-022	1-Indispensable	1	
	Comprensibilidad		PXX01-RNF-021	1-Indispensable	1	
Total Usabilidad					2	
<b>Total 1-RNF-Calidad</b>					<b>15</b>	
2-RNF-Restricciones	Ambiente	Base de datos	PXX01-RNF-036	1-Indispensable	1	
		Implementación	PXX01-RNF-032	1-Indispensable	1	
			PXX01-RNF-033	1-Indispensable	1	
		Servidor de Aplicaciones	PXX01-RNF-037	1-Indispensable	1	
		Sistema Operativo	PXX01-RNF-035	1-Indispensable	1	
	Total Ambiente					5
	Arquitectura	Accesibilidad		PXX01-RNF-041	1-Indispensable	1
				PXX01-RNF-042	1-Indispensable	1
		Diseño de Capas	PXX01-RNF-039	1-Indispensable	1	
		Framework	PXX01-RNF-038	1-Indispensable	1	
		Librerías / Componentes	PXX01-RNF-043	1-Indispensable	1	
		Navegador	PXX01-RNF-044	1-Indispensable	1	
		Solicitudes de Mails / Mensajería	PXX01-RNF-040	1-Indispensable	1	
	Total Arquitectura					7
	Lenguaje	De programación		PXX01-RNF-029	1-Indispensable	1
	Total Lenguaje					1
	Proceso de desarrollo	Alcance		PXX01-RNF-001	1-Indispensable	1
				PXX01-RNF-002	1-Indispensable	1
Nivel de detalle			PXX01-RNF-003	1-Indispensable	1	
			PXX01-RNF-004	1-Indispensable	1	
			PXX01-RNF-005	1-Indispensable	1	
			PXX01-RNF-006	1-Indispensable	1	
			PXX01-RNF-007	1-Indispensable	1	
	PXX01-RNF-008	1-Indispensable	1			
Total Proceso de desarrollo					8	
<b>Total 2-RNF-Restricciones</b>					<b>21</b>	
<b>Total general</b>					<b>36</b>	

FIGURA 10.15: Vista de la Plantilla de especificación de RNF Web

Debido al volumen de RNF candidatos identificados a lo largo de los procesos de Elicitación y Especificación de RNF Web, la eficiencia de la aplicación de las actividades de cada proceso quedó demostrada mediante ejemplos puntuales relacionados a los tipos de stakeholders del caso de estudio, la matriz de control de autorización de accesos de usuario y sobre todo con ejemplos particulares de los RNF de calidad y restricciones. Como

complemento a este trabajo se proporcionan todas las plantillas que se completaron para el caso de estudio como parte de la ejecución de los procesos de Elicitación y Especificación de RNF Web:

1. Plantilla de Elicitación de RNF de Calidad 1.0
2. Plantilla de Elicitación de RNF de Restricciones 1.0
3. Matriz de RNF 1.0 – Versión Inicial
4. Matriz de RNF 1.1 – Versión Intermedia
5. Matriz de RNF 1.2 – Versión Definitiva
6. Plantilla de Especificación de RNF Web 1.0

Si bien el objetivo de este capítulo fue alcanzado al demostrar la aplicación de los procesos de Elicitación y Especificación de RNF Web, como parte de los resultados podemos mencionar también el análisis realizado a partir de la experiencia de aplicar los procesos en un caso real y que va más allá del seguimiento de los lineamientos establecidos en los procesos precedentes. A continuación se describen los ítems relevantes de este análisis:

- Cantidad de RF Vs cantidad de RNF: La captura de requerimientos involucró un total de **16 RF** y **36 RNF** que a su vez se dividen en 15 requerimientos de calidad y 21 restricciones. Esto demuestra que cuando existen herramientas y lineamientos que ayudan al ingeniero de requerimientos a capturar RNF, estos pueden ser identificados y documentados correctamente.
- RNF en Negociación: Se identificaron **7 RNF** “En Negociación”, lo cual significa no solo que los stakeholders tienen intereses diferentes, sino también es un indicador de que en el dominio de los RNF muchas veces existe más de una solución para un problema. Los diferentes RNF en Negociación merecen ser expuestos y analizados hasta decidir por el más conveniente para la aplicación Web y la organización.
- RNF en Análisis: Se identificaron **9 RNF** “En Análisis”, y los mismos fueron resueltos mediante la interacción entre los procesos de Elicitación y Especificación. El análisis de la factibilidad técnica ayudó a definir las restricciones de arquitectura, por ejemplo: *Al codificar el prototipo diseñado para celulares en html5 con ajax para celulares con*

*android versión 2.1*, permitió tomar una restricción de Arquitectura importante y más económica respecto de la idea de personalizar los desarrollos de interface de acuerdo a los diferentes modelos de celulares con los cuales los usuarios accederán a la aplicación.

- RNF Pospuestos: Al finalizar el proceso de especificación, de los **44 RNF** candidatos, solamente quedaron **36 RNF** para la aplicación Web, **4 RNF** fueron Pospuestos y otros **4** fueron Rechazados, según lo muestra el cuadro.

Clasificación según estado	
En negociación	0
En análisis	0
Pospuestos	4
Rechazados	4
Pendiente validación	36

Lo curioso es que al iniciar el proceso de Elicitación, los stakeholders querían una aplicación Web solo para PC: *La aplicación web debe ser desarrollada para PC y notebook*, y, a lo largo del proceso de elicitación, producto de las preguntas realizadas a partir de las plantillas de Elicitación de RNF de Calidad y Restricciones, y la interacción con el proceso de especificación, este requerimiento terminó formando parte de los **RNF Pospuestos** ya que otro RNF surgió como relevante para la versión 1.0 de la aplicación Web *Contar con una Aplicación Web compatible con teléfonos celulares*.

- RNF Indispensables Vs RNF Necesarios: Se identificaron 4 RNF **Necesarios** y 32 RNF **Indispensables**. La priorización de RNF servirá de guía para los integrantes de equipo de trabajo involucrados en las fases siguientes del ciclo de vida de desarrollo de la aplicación Web.
- RNF de Calidad: Los RNF relacionados a la Precisión, Seguridad, Madurez y Tolerancia a Fallas y capacidad de instalación surgieron a partir de las preguntas proporcionadas en las Plantillas de Elicitación de RNF de calidad. En la práctica este tipo de requerimientos o no son identificados o suelen surgir en las fases finales del ciclo de vida de desarrollo.

- RNF restricciones de Proceso de desarrollo: Se identificaron 8 RNF de proceso, las cuales fueron relevantes para seleccionar el equipo de trabajo y organizar las actividades del ciclo de vida de desarrollo de la aplicación Web en base a las restricciones establecidas para documentar, codificar, probar e implementar la aplicación, como así también para la gestión de configuración de los productos generados como parte del proceso.
- RNF Restricciones: Los RNF relacionados a la Arquitectura, Ambiente, surgieron a partir de las preguntas proporcionadas en las Plantillas de Elicitación de RNF de restricciones. Por lo general este tipo de información no suele identificarse durante el relevamiento, en el común de los casos, surgen en la fase de diseño y codificación del ciclo de vida de desarrollo.
- Patrón para especificar los RNF: Contar con el Patrón para escribir los RNF, ayudo a obtener requerimientos, correctos, factibles, priorizables, no ambiguos.
- Matriz de RNF: Dio soporte a los procesos de Elicitación y Especificación. Es una herramienta valiosa en sí misma, dado que la versión definitiva representa la trazabilidad del proceso de captura, análisis, especificación, negociación de los RNF candidatos. Se generaron tres versiones de la Matriz de RNF, la versión inicial Matriz de RNF 1.0 (salida del proceso de Elicitación) sirvió para registrar respuestas de Stakeholders, Reglas de negocio, Políticas de la organización. La Matriz de RNF versión 1.1 generada al inicio del proceso de especificación, en la cual surgieron los RNF candidatos con sus diferentes estados “En Negociación”, “en Análisis”, “Pendiente de validación” y finalmente la Matriz de RNF 1.2 al cerrar el proceso de especificación donde los RNF quedaron registrados con los estados “Pospuesto”, “Rechazado” y “Pendiente de validación”.

Para resumir y concluir, la aplicación de los lineamientos y técnicas definidas en este trabajo al caso real descrito en este capítulo, muestra que es posible la captura y especificación de los RNF de aplicaciones Web cuando se cuenta con un proceso y técnicas adecuadas para tal fin.

## Capítulo: 11

---

### 11. Conclusiones y trabajos futuros

El presente capítulo ofrece un resumen de las conclusiones de esta tesis, como así también de los aportes. Al final del capítulo se describen los trabajos futuros.

El objetivo de este trabajo fue desarrollar técnicas y lineamientos para los procesos de Elicitación y Especificación de la Ingeniería de Requerimientos, que sean aplicables al tratamiento de los requerimientos no funcionales en un marco de desarrollo de aplicaciones Web. Para ello, en el capítulo 2 se presentó el estado de arte de los conceptos que definen a los RNF dentro de la literatura existente en la Ingeniería de Requerimientos, quedó expuesta la carencia de consenso que surge de la diversidad de términos utilizados en las definiciones, con significados poco claros, que dan lugar a la ambigüedad sobre su alcance o lo que representan los mismos. Las discrepancias conceptuales también se encontraron en las propuestas de clasificaciones de los Requerimientos No Funcionales, que incluyen árboles o categorías de conceptos que se muestran en forma separada en otras definiciones. Luego se establecieron las bases conceptuales a partir de lo expuesto por Cysneiros [7], se considera que los Requerimientos No Funcionales son requerimientos de calidad y son restricciones.

En el capítulo 3 se analizaron en forma completa los procesos de desarrollo de WSDM, HFPM, UWE, WebML, NDT, DDDP, se identificaron los tipos de Requerimientos No Funcionales contemplados por cada uno y las técnicas de ingeniería de requerimientos propuestas por estos para su tratamiento y gestión. Además se establecieron la fase o fases del ciclo de vida de desarrollo de software en las que son tratados los Requerimientos No Funcionales. En el capítulo 4 se revisaron un conjunto de técnicas que la Ingeniería de Requerimientos ofrece para la elicitación, especificación y validación de requerimientos. En el capítulo 5 se desplegaron los resultados del estudio comparativo de los seis enfoques metodológicos estudiados, con tres puntos de vista: a) *Tipos de RNF identificados en cada metodología.* b) *Técnicas de elicitación, especificación y validación de la IR que utilizan para el tratamiento de los RNF* y c) *RNF identificados por cada metodología y el ciclo de vida.*

Los resultados del análisis permiten apreciar que:

- Las metodologías de aplicaciones Web consideran RNF a pesar de la carencia de consenso en su significado.
- En los enfoques estudiados no hay técnicas específicas para la elicitación de Requerimientos No Funcionales tampoco hay consenso sobre como especificarlos ni lineamientos de como validarlos.
- Existen herramientas para tratarlos pero las mismas son particulares del proceso metodológico al que soportan.
- Los procesos de desarrollo de los enfoques estudiados no brindan técnicas que soporten el tratamiento de los Requerimientos No Funcionales a lo largo del ciclo de vida.
- Hay carencia de consenso para establecer la fase del ciclo de vida de desarrollo en que se identifican los Requerimientos No Funcionales. De hecho hay Requerimientos No Funcionales que no son identificados durante la fase de relevamiento y análisis.

La complejidad de los requerimientos en el desarrollo de aplicaciones Web demanda que el ingeniero de requerimientos utilice herramientas y técnicas acordes a la naturaleza de los mismos, particularmente los RNF necesitan ser identificados, clasificados, analizados para recién ser documentados apropiadamente. Si bien actualmente la IR proporciona numerosas técnicas y herramientas para capturar, describir, validar y gestionar requerimientos, a menudo estos no son aplicados por los profesionales de sistemas, particularmente en el desarrollo de aplicaciones Web. La madurez del proceso de IR parece ser insuficiente y demanda nuevos enfoques o la evolución de los enfoques existentes para el tratamiento de los RNF en particular.

Por su importancia, las actividades para captura, especificación y validación de los RNF Web, merecen contar con técnicas y lineamientos específicos.

Los capítulos 6, 7, 8 y 9 representan los aportes más relevantes de esta tesis. En el capítulo 6 se presentaron los procesos para la Elicitación y Especificación de RNF Web. La simplicidad de los mismos permite que estos puedan ser utilizados por los enfoques metodológicos estudiados en esta tesis y por otras metodologías de desarrollo de aplicaciones



Web, puesto que ofrecen un conjunto de técnicas y lineamientos específicos para dar soporte a la captura y especificación de los RNF Web.

En los capítulos 8 y 9 se describieron en detalle cada una de las plantillas que dan soporte a los procesos de Elicitación y Especificación de RNF web, las mismas fueron diseñadas especialmente a partir de las bases conceptuales expuestas en el capítulo 2, donde se establece que los Requerimientos No Funcionales son *requerimientos de calidad* y *son restricciones* y las consideraciones particulares explicadas en el capítulo 7. Para el proceso de Elicitación se diseñaron una: *Plantilla de Elicitación de RNF de Calidad* y una *Plantilla de Elicitación de RNF-Restricciones*, las cuales están dirigidas a un conjunto de stakeholders específicos, al comienzo del capítulo 7 se presentaron los stakeholders considerados para las plantillas elicitación de RNF.

La *plantilla para la elicitación de RNF de Calidad* para aplicaciones Web que propone este trabajo, reúne 5 de las 6 características y 11 de las 27 subcaracterísticas del estándar de calidad internacional ISO/IEC 9126-1[34]. Cabe acotar que al momento de concluir esta tesis, este estándar está en proceso de revisión, esperándose como resultado su aprobación y su inclusión en la nueva serie de estándares ISO/IEC 25000.

La Mantenibilidad y sus subcaracterísticas no están incluidas en la *Plantilla de Elicitación de RNF de Calidad*, aunque la Mantenibilidad es un factor asociado a la calidad de la aplicación final. Al igual que se expresa en [22], consideramos que esta característica está relaciona con el proceso de desarrollo. Si se cuenta con un proceso de desarrollo completo, organizado, documentado y con lineamientos que den soporte a las actividades de todo el ciclo de vida de desarrollo, el mantenimiento de la aplicación Web y la gestión de cambio serán facilitadas por disponer de modelos de diseño, un software modular y la documentación asociada que permite la trazabilidad de los requerimientos para la corrección de errores o para su evolución.

Cada una de las características/ subcaracterísticas consideradas, fueron analizadas desde el punto de vista de las aplicaciones Web, para justificar su inclusión en la plantilla de Elicitación. Esta fue una tarea ardua debido a que los conceptos del estándar ISO/IEC 9126-1 están enfocados particularmente a sistemas tradicionales.

La *Planilla de Elicitación de RNF Restricciones*, agrupa restricciones de proceso de desarrollo, de lenguaje, ambiente, arquitectura, y navegadores, las que contienen 14 subcategorías que, de acuerdo a nuestra experiencia, consideramos relevantes para el

desarrollo de aplicaciones Web. Las restricciones de la plantilla no solo son significativas para evaluar la factibilidad técnica de la solución, sino que además el conocimiento asociado a las mismas servirá de base para las actividades de las fases siguientes del ciclo de vida de desarrollo.

Para el proceso de Especificación se diseñó un *Patrón para especificar RNF* y una *Plantilla de Especificación de RNF Web*, estos se diseñaron en base a las prácticas recomendadas en el estándar IEEE 830-1998 [36] para la especificación de requerimientos de software. Finalmente se presentó una *Matriz de RNF*, la cual fue diseñada para brindar soporte a los procesos de Elicitación y Especificación de RNF Web respectivamente.

La aplicación de los procesos y plantillas propuestos por esta tesis, se demostraron en un caso real: Elicitación y especificación de RNF de la aplicación Web “Indicadores de barcos y grúas”; la ejecución paso a paso descrita en detalle y las conclusiones particulares se describieron en el capítulo 10. A continuación se desprenden conclusiones generales a partir de la experiencia aplicada y la propuesta en sí:

- Las plantillas que se proponen en este trabajo pueden considerarse no solo como una herramienta para capturar el conocimiento, sino que además sirven como una guía que da soporte al ingeniero de requerimientos durante la búsqueda y adquisición del conocimiento relevante del dominio del problema relacionado a los RNF.
- Las preguntas diseñadas para las plantillas de Elicitación de RNF de calidad y restricciones, sirven como un marco de referencia para el ingeniero de requerimientos al momento de iniciar un relevamiento, sobre todo si carece de experiencia en desarrollo de aplicaciones Web.
- El ingeniero de requerimientos debe identificar la característica del proceso de desarrollo durante la elicitación, debido a que el *alcance* y el *nivel de detalle del proceso de desarrollo* establecen el marco de trabajo a seguir para el desarrollo de la aplicación por lo tanto tienen impacto directo en la calidad de los productos intermedios y la aplicación Web resultante.
- La tipificación de RNF de calidad y restricciones propuesta en las plantillas de Elicitación y Especificación, permiten analizar, obtener y documentar un modelo de RNF completo en la fase inicial del ciclo de vida de desarrollo,

ya que de lo contrario y como se vio en el resultado del análisis comparativo realizado en el capítulo 5, existe carencia de consenso para establecer en qué fase del ciclo de vida de desarrollo se identifican los RNF, y peor aún, hay RNF que no son identificados durante la fase de relevamiento y análisis.

- Contar con un *patrón para escribir los RNF*, permite obtener requerimientos RNF, correctos, factibles, priorizables, no ambiguos, que serán evaluados en el proceso de Validación.
- La *matriz de RNF* es una herramienta valiosa en sí misma, la versión final, actualizada en el proceso de especificación permite la trazabilidad del proceso de captura, análisis, especificación, tiene registro de los RNF candidatos que quedaron pospuestos o rechazados a raíz de la negociación y permite también trazar cada RNF con su origen.
- La *Planilla de especificación de RNF Web*, salida del proceso de especificación representa un modelo de los RNF de una aplicación Web, la cual una vez validada, servirá de entrada a las fases siguientes del ciclo de vida de desarrollo.

Con relación a la comunicación de los resultados, además de la publicación ya realizada se han planificado tres publicaciones para comunicar los resultados de esta tesis:

1. Una primera publicación en una conferencia conteniendo la visión global de los procesos de elicitación y especificación de Requerimientos No Funcionales para aplicaciones Web.
2. Una publicación enfocada en los documentos de soporte del proceso de elicitación de Requerimientos No Funcionales para aplicaciones Web. También orientada a una conferencia.
3. Una descripción extensa y detallada del proceso en su conjunto y sus documentos de soporte, orientada a una revista.

Los trabajos a futuro, a partir de esta tesis podrían enfocarse en:

- Proponer un conjunto de técnicas y lineamientos específicos para un

proceso de validación de RNF, acorde a los procesos definidos en esta tesis para aplicación de Ingeniería de Requerimientos Web al tratamiento de los RNF.

- Estudiar la aplicación los procesos y plantillas definidas en este trabajo en desarrollos de aplicaciones Web con los enfoques WSDM, HFPM, UWE, WebML, NDT, DDDP.
- Estudiar la aplicación los procesos y plantillas definidas en este trabajo en desarrollos de aplicaciones Web de diferentes categorías: Sitios Web centrados en documentación, aplicaciones Web interactivas, aplicaciones web transaccionales, aplicaciones web basadas en workflow, Web sociales, aplicaciones web orientadas a portales, aplicaciones web basados en la localización, web semánticas entre otras.
- Extensión de las categorías y/o preguntas de la plantilla de elicitación de RNF-Restricciones.
- Extensión de la plantilla de RNF de calidad, una vez que se publique la versión ISO/IEC 25000. Generar nuevas preguntas para las categorías actuales y nuevas.
- Otro flujo de acción podría orientarse a automatizar los procesos de Elicitación y Especificación de RNF para aplicaciones Web presentados en este trabajo. Por ejemplo, la generación automática de entrevistas en base a los stakeholders y las preguntas seleccionadas para las subcaracterísticas y subcategorías propuestas en las plantillas de elicitación de RNF.

## 12. Bibliografía

1. Escalona, M. J., Koch, N.: *Requirements Engineering for Web applications – A Comparative Study*. Journal of Web Engineering, Vol. 2 No. 3, pp. 193-212 (2004).
2. Escalona, M. J., Koch, N.: *Metamodeling the Requirements of Web Systems*. J. Filipe, J. Cordeiro, and V. Pedrosa (Eds): WEBIST 2005/2006, LNBIP 1, pp. 267-280. Springer, Heidelberg (2007).
3. IEEE Standard 610. *IEEE Standard Glossary of Software Engineering Terminology*. Std 610.12-1990, IEEE, New York. (1990).
4. Cysneiros, L.M.: *Requerimientos No Funcionales desde la Elicitación al Modelo Conceptual*. Tesis Doctoral. Departamento de Informática. Politécnica Universidad Católica de Río de Janeiro (2001).
5. Kappel, G., Pröll, B., Reich, S., Retschitzegger, W.: *Web Engineering. The Discipline of Systematic Development of Web Applications*. G. Kappel, B. Pröll, S. Reich, & W. Retschitzegger (eds). John Wiley & Sons Inc. (2006).
6. Rossi, G., Pastor, O., Schwabe, D., & Olsina, L. *Web Engineering: Modelling and Implementing Web Applications*. London: Springer (2008).
7. Cysneiros L.M. and Yu E.: *Non-Functional requirements Elicitation*. “Book” Chapter #, pp. 1-24 (2004).
8. Chung, L., Nixon, B.: “*Dealing with Non-Functional Requirements: Three Experimental Studies of a Process-Oriented Approach*” Proc. 17th Int. Con. on Software Eng. Seattle, Washington, April pp: 24-28, 1995.
9. De Troyer, O., Leune, C.: *WSDM: A User Centered Design Method for Web Sites*. Technical Report of Tilburg University, Infolab. Belgium (1997).
10. Olsina, L.: *Building a Web-based Information System applying the Hypermedia Flexible Process Modeling Strategy*. 1st International Workshop on Hypermedia Development, Hypertext’98, Pittsburg, USA. (1998).
11. Schwabe D., Rossi G: *Developing Hypermedia Applications using OOHDM*. Workshop on Hypermedia Development Process, Methods and Models, Hypertext’98, Pittsburg, USA. (1998).
12. Lange D.B: *An Object-Oriented Design Approach for Developing Hipermedia Information Systems*. Research Report RT00112, IBM Research, Tokyo Research Laboratory, Japón, (1995).
13. Isakowitz, T.; Stohr, E.; Balasubramanian, P.: *RMM: a methodology for structured hypermedia design*. Comm (D.B. 1995). ACM 38, 8; pp. 34-48. (1995).

14. Hennicker, R., Koch, N.: *A UML-based Methodology for Hypermedia Design*. Lecture Notes in Computer Science. Proc. UML'2000. York, England. (2000).
15. Koch, N., Wirsing, M.: *Software Engineering for Adaptive Hypermedia Applications ?* Poster Presentation, Third Workshop on Adaptive Hypertext and Hypermedia at the 8<sup>th</sup> International Conference on User Modeling. Sonthofen, Germany; pp 13-17. July (2001).
16. Koch N.: *Classification of Model Transformation Techniques used in UML-based Web Engineering*. Ludwig-Maximilians-Universität. Oettingenstr. 67, 80538. And FAST GmbH Arabellastr. 17, 81925 München, Germany.
17. Jacobson I., Booch G., Rumbaugh J.: *The Unified Software Development Process*. Addison Wesley. (1999).
18. Booch G., Rumbaugh J., Jacobson I.: *The Unified Modeling Language: A User Guide*. Addison Wesley (1999).
19. <http://argouml.tigris.org/>
20. Koch, N., Kraus, A., Hennicker, R.: *The Authoring Process of the UML-based Web Engineering Approach*. Institute of Computer Science. Ludwig-Maximilians University of Munich - Oettingenstr. 67, D-80538. And F.A.S.T. Applied Software Technology GmbH Arabellastr. 17, D-81925 München, Germany.
21. Ceri, S., Fraternali, P., Bongio, A.: *Web Modeling Language (WebML): a modeling language for designing Web sites*. Dipartimento di Elettronica e Informazione, Politecnico di Milano. Piazza L. da Vinci, 32-20133, Milano, Italy. (2000).
22. Ceri, S. Fraternali, P., Bongio, A., Brambilla M., Comai S., Matera, M.: *Designing Data-Intensive Web Applications*. Morgan Kaufman. (2003).
23. [http://www.txtgroup.com/newsletter/attachment/w3i3\\_paper.pdf](http://www.txtgroup.com/newsletter/attachment/w3i3_paper.pdf)
24. <http://www.webratio.com/sv1.do>
25. Escalona, M.J., Mejías, M., Torres, J., Jurado, M.C.: *NDT: Navigational Development Techniques*. Departamento de Lenguajes y Sistemas Informáticos. Escuela Técnica Superior de Ingeniería Informática. Universidad de Sevilla. Avenida Reina Mercedes, S/N 41012. Sevilla.
26. Escalona, M.J., Mejías, M., Torres, J.: *Developing Systems with NDT and NDT-Tool*. Department of Computer Languages and Systems University of Seville. ETS Ingeniería Informática. AV. Reina Mercedes S/N 41012 Sevilla, Spain.
27. Escalona, M., J.: *Modelos y técnicas para la especificación y el análisis de la navegación en sistemas software – Tesis Doctoral*. Departamento de Lenguajes y Sistemas Informáticos. Escuela Técnica Superior de Ingeniería Informática. Universidad de Sevilla. Avenida Reina Mercedes, S/N 41012. Sevilla. Anexo A Manual de referencia NDT p. 188-236. (2004).

28. Lowe, D.: *Web System Requirements: an overview*. Requirements Eng. Vol. 8. 102–113. - 10.1007/s00766-002-0153-x, DOI (2003).
29. Lowe, D., Eklund, J.: *Client Needs and the design process in web projects*. Journal of Web Engineering, Vol. 1, No.1 - 023-036. © Rinton Press. (2002).
30. Cysneiros, L.M., Leite, J.C. S.: *NonFunctional Requirements: From Elicitation to Conceptual Models*. Published by the IEEE Computer Society, (2004).
31. Glinz, M.: *On-Non-Functional Requirements*. 15<sup>th</sup> IEEE International Requirements Engineering Conference. 21-26. DOI 10.1109/RE.2007.45. IEEE Computer Society, (2007).
32. Chung, L., Leite, J.C. S.: *On-Non-Functional Requirements in Software Engineering*. A.T.Borgida et al. (Eds.): Mylopoulos Festschrift, LNCS 5600, 363–379. © Springer – Verlag Berlin Heidelberg. (2009).
33. Sutcliffe, A. *On the Inevitable Interwining of Requirements and Architecture*. Design Requirements Workshop (págs. 168-185). Cleveland: Springer. (2007).
34. ISO/IEC 9126-1:2001: Software Engineering - Product Quality - Part 1: Quality Model (2001).
35. Loucopoulos, P.; Karakostas, V.: *System Requirements Engineering*. Copyright© McGraw-Hill Company (UK) Limited –London (1995).
36. IEEE, *IEEE Recommended Practice for Software Requirements Specifications*. IEEE Std 830-1998 (Revision of IEEE Std 830-1993).
37. Rojo, S: *Requerimientos No Funcionales para aplicaciones Web*. Trabajo Integrador para obtener el grado de especialista en Ingeniería de Software. Departamento de Informática. Universidad Nacional de La Plata (2012).
38. Antonelli, L., Rossi, G., Sampaio do Prado Leite, J. C., Oliveros, A.: *Deriving requirements specifications from the application domain language captured by Language Extended Lexicon*. Departamento de Informática. Universidad Nacional de La Plata (2012).
39. Hadad, G.D.S: *Uso de Escenarios en la Derivación de Software* . Tesis Doctoral. Departamento de Informática. Universidad Nacional de La Plata (2008).
40. Rojo, S., Oliveros, A.,: *Requerimientos No funcionales para aplicaciones Web*. ASSE 2012 - 13th Argentine Symposium on Software Engineering, publicado en *Anales de las 41 JAIIO*. ISSN 1850-2776, Buenos Aires (2012).
41. Robertson S., Robertson J.: *Mastering the Requirements Proccess Second Edition*”. Addison Wesley Professional. ISBN-13: 978-0-321-41949-1. Marzo 2006.

42. Oliveros, A.: *Calidad del Software*. Material de Cátedra Administración de Proyectos (pág. 3). Universidad Nacional de La Plata (2012).
43. Alaeddine, N., Tian J.: *Analysis of Anomalies and Failures in Dynamic Web Applications*, Department of Computer Science and Engineering Southern Methodist University, Dallas, Texas, USA.
44. Guo, Y., Sampath, S.: *Web Application Fault Classification— An Exploratory Study*. Department of Information Systems. University of Maryland Baltimore, Country, Baltimore, Maryland 21250, USA (2008).
45. Políticas relating to Web Accessibility: <http://www.w3.org/WAI/Policy/>.
46. Nielsen, J.: *Writing for the Web*. <http://www.nngroup.com/topic/writing-web/> [última visita: 2012-11-25].
47. Nielsen, J. *10 Usability Heuristics*: <http://www.nngroup.com/articles/ten-usability-heuristics/> [última visita: 2012-11-25].
48. Leavitt, M., O., Shneiderman, B.: *Research-Based Web Design & Usability Guidelines*. ISBN 0-16-076270-7. [http://www.usability.gov/guidelines/guidelines\\_book.pdf](http://www.usability.gov/guidelines/guidelines_book.pdf) [última visita: 2012-12-07].
49. Solinas, M, A. *Elicitación y Trazabilidad de Requerimientos utilizando Patrones de Seguridad*. Tesis para Obtener el grado de Magister en Ingeniería de Software. Universidad Nacional de La Plata (2012).
50. Losavio, F., Matteo, A., Rahamut, R.: *Quality view of Web Services domain*. Sixth LACCEI International Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2008). *Partnering to Success: Engineering, Education, Research and Development*. Tegucigalpa, Honduras, (2008).
51. Mairiza, D., Zowghi, D., Nurmuliani N.: *An Investigation into the Notion of Non-Functional Requirements*. Faculty of Engineering and Information Technology. University of Technology Sydney. Australia. Copyright, (2010).
52. Wiegers, K. E.: *Software Requirements*. 2nd Edition by Microsoft Press. ISBN 0-7356-1879-8. (2003).
53. Wiegers, K. E.: *Writing Quality Requirements* . Process Impact [www.processimpact.com](http://www.processimpact.com). Copyright, (1999).
54. *Guía de pruebas de OWASP. Versión 3.0*. [https://www.owasp.org/index.php/Projects/OWASP\\_Testing\\_Project/Releases/Testing\\_Guide\\_V\\_3.0](https://www.owasp.org/index.php/Projects/OWASP_Testing_Project/Releases/Testing_Guide_V_3.0) [última visita: 2012-11-05].



## 13. Anexos

### 13.1 Plantilla de Elicitación de RNF de Calidad

El archivo Excel *Plantilla Elicitacion RNF de Calidad- v1.0.xls* cuenta con 4 solapas:

- *La Plantilla de Elicitación de RNF de Calidad:* Contiene la plantilla propuesta en esta tesis, la cual contiene hipervínculos a las definiciones de las características y subcaracterísticas del estándar ISO 9126-1 y de los tipos de Stakeholders.
- *Las características del estándar ISO 9126-1:* Contiene las definiciones de las características y subcaracterísticas del estándar, cada definición es referenciada desde la Plantilla de Elicitación de RNF de Calidad.
- *Los tipos de Stakeholders:* Contiene las definiciones de los tipos de stakeholders, estas son referencias desde la Plantilla de Elicitación de RNF de Calidad
- *La Matriz de Control de Autorización:* Contiene la plantilla propuesta para dar soporte a la definición de roles autorizados. Es referenciada desde la Plantilla de Elicitación de RNF de Calidad (Subcaracterística Seguridad).

### 13.2 Plantilla de Elicitación de RNF de Restricciones

El archivo Excel *Plantilla Elicitacion RNF Restricciones v1.0.xls* cuenta con 3 solapas:

- *La Plantilla de Elicitación de RNF Restricciones:* Contiene la plantilla propuesta en esta tesis, la cual contiene hipervínculos a las definiciones de las categorías y subcategorías de restricciones propuestas en esta tesis y de los tipos de Stakeholders.
- *Categorías de Restricciones:* Contiene las definiciones de las categorías y subcategorías, cada definición es referenciada desde la Plantilla de Elicitación de RNF Restricciones.
- *Los tipos de Stakeholders:* Contiene las definiciones de los tipos de stakeholders, estas son referencias desde la Plantilla de Elicitación de RNF Restricciones.

### **13.3 Plantilla de Especificación de RNF Web**

El archivo Excel *Plantilla de Especificacion RNF Web-v1.0.xls* cuenta con 2 solapas:

- *El Patrón para especificación de RNF*: Contiene las secciones del patrón propuesto en esta tesis.
- *Plantilla de Especificación de RNF Web*: Contiene la Plantilla descrita en el capítulo 9.

### **13.4 Matriz de RNF**

El archivo Excel *Matriz de RNF- v1.0.xls* cuenta con 2 solapas:

- *Matriz de RNF*: Contiene la Plantilla propuesta en esta tesis, descrita en el capítulo 9.
- *Gráfico - Estado de RNF*: Contiene el gráfico que muestra el estado de los RNF, se genera a partir de la Matriz de RNF.

### 13.5 Aplicación del procesos de Elicitación y Especificación Web

Se adjuntan las Planillas que corresponden al caso práctico descrito en el capítulo 10.

#### 13.5.1 Stakeholders claves de la aplicación Web del caso de estudio

La figura 13.1 muestra el listado de los usuarios potenciales de la aplicación web “Indicadores de Grúas y Barcos”, definidos con la información que proponen en [41] Robertson and Robertson.

<b>Nombre de usuario:</b>	<i>PS</i>
<b>Rol de usuario:</b>	<i>Gerente de Área de Carga</i>
<b>Experiencia en el conocimiento del dominio:</b>	<i>Media. Tiene conocimiento de los sistemas y procedimientos que soportan las actividades del área que coordina.</i>
<b>Experiencia tecnológica:</b>	<i>Si.</i>
<b>Otras características:</b>	<i>Hombre. Es Ingeniero Industrial. Actitud positiva para el trabajo. Le gusta la tecnología especialmente los celulares. Habla Español e Inglés.</i>
<b>Nombre de usuario:</b>	<i>SD</i>
<b>Rol de usuario:</b>	<i>Subgerente de Área de Carga</i>
<b>Experiencia en el conocimiento del dominio:</b>	<i>Alta tiene conocimiento de los sistemas y procedimientos que soportan las actividades del área. Opera los sistemas disponibles para generar informes.</i>
<b>Experiencia tecnológica:</b>	<i>Si.</i>
<b>Otras características:</b>	<i>Hombre. Es ingeniero Industrial. Actitud positiva para el trabajo. Le gusta la tecnología especialmente los celulares. Habla Español.</i>
<b>Nombre de usuario:</b>	<i>MF</i>
<b>Rol de usuario:</b>	<i>Empleado de Área de Carga</i>
<b>Experiencia en el conocimiento del dominio:</b>	<i>Alta tiene conocimiento de los sistemas y procedimientos que soportan las actividades del área. Opera y parametriza los sistemas disponibles.</i>
<b>Experiencia tecnológica:</b>	<i>Si.</i>
<b>Otras características:</b>	<i>Hombre. Estudiante de Ingeniería en Informática. Actitud positiva para el trabajo. Le gusta la tecnología. Habla Español e Inglés.</i>

FIGURA 13.1: Usuarios potenciales de la aplicación Web del caso de estudio.

La figura 13.2 presenta el listado de otros Stakeholders candidatos de la aplicación Web “Indicadores de Grúas y Barcos”

<b>Identificación del Stakeholder</b>	<i>Gerente de Sistemas. MC.</i>
<b>Conocimiento necesario por el proyecto.</b>	<i>Tiene conocimiento de la plataforma de sistemas automatizados que dan soporte a las actividades de la organización.</i>
<b>Grado de influencia.</b>	<i>Alto. Es el responsable del seleccionar la empresa que desarrolla la aplicación Web y quien determina los criterios de aceptación de calidad.</i>
<b>Identificación del Stakeholder</b>	<i>Referente técnico de Aplicaciones. MG</i>
<b>Conocimiento necesario por el proyecto.</b>	<i>Conoce el modelo y las bases de datos de los sistemas que dan soporte al dominio de la aplicación.</i>
<b>Grado de influencia.</b>	<i>Bajo.</i>
<b>Identificación del Stakeholder</b>	<i>Referente técnico de Infraestructura. LP</i>
<b>Conocimiento necesario por el proyecto.</b>	<i>Conoce la plataforma de hardware, comunicación y redes de la organización.</i>
<b>Grado de influencia.</b>	<i>Bajo.</i>
<b>Identificación del Stakeholder</b>	<i>Referente técnico de Aplicaciones. AL</i>
<b>Conocimiento necesario por el proyecto.</b>	<i>Arquitecto de aplicaciones JAVA.</i>
<b>Grado de influencia.</b>	<i>Alto</i>
<b>Identificación del Stakeholder</b>	<i>Responsable de Infraestructura. PR</i>
<b>Conocimiento necesario por el proyecto.</b>	<i>Conoce la plataforma de hardware, comunicación y redes de la organización que coordina.</i>
<b>Grado de influencia.</b>	<i>Bajo.</i>
<b>Identificación del Stakeholder</b>	<i>Especialista en Sistema AVI. GG</i>
<b>Conocimiento necesario por el proyecto.</b>	<i>Realiza el mantenimiento evolutivo de la aplicación</i>
<b>Grado de influencia.</b>	<i>Bajo.</i>

FIGURA 13.2: Otros Stakeholders de la aplicación Web del caso de estudio.

### **13.5.2 Plantilla de Elicitación de RNF de Calidad del caso de estudio**

El archivo *1. Plantilla Elicitacion RNF de Calidad- Caso v1.0.xls* es el producto del paso *1.4 Aplicar los Checklist de RNF de Calidad y Restricciones* del proceso de Elicitación de RNF Web, en el caso de estudio.

### **13.5.3 Plantilla de Elicitación de RNF Restricciones del caso de estudio**

El archivo *2. Plantilla Elicitacion RNF Restricciones- Caso v-1.0.xls* es el producto del paso *1.4 Aplicar los Checklist de RNF de Calidad y Restricciones* del proceso de Elicitación de RNF Web, en el caso de estudio.

### **13.5.4 Matriz de RNF 1.0 - Versión Inicial del caso de estudio**

El archivo *3. Matriz de RNF- Caso-v1.0.xls* es el producto del proceso de Elicitación de RNF Web aplicado al caso de estudio.

### **13.5.5 Matriz de Control de Autorización de Acceso del caso de estudio**

La figura 11.3 muestra la Matriz de control de autorización de accesos del caso de estudio, resultado de aplicar proceso de Elicitación de RNF Web, de la subcaracterística de calidad *Seguridad*.

Funcionalidades	Tipos de usuarios		
	Administrador	Consultante de Indicadores	Receptor de Reportes
Visualizar los indicadores Gross Vessel Rate (GVR) y Gross Crane Rate (GCR) de		Si	
Visualizar el indicador Gross Crane Rate (GCR) y el Ritmo de cada grúa.		Si	
Visualizar el valor promedio de la cantidad de grúas que trabajaron en un barco por		Si	
Crear alarmas relacionadas al ritmo de cada grúa.	Si		
Modificar alarmas relacionadas al ritmo de cada grúa.	Si		
Eliminar alarmas relacionadas al ritmo de cada grúa.	Si		
Visualizar las horas de demoras de un buque agrupadas tipo de demora.		Si	
Consultar Resumen diario de los indicadores y total de movimientos por buque.			Si
Consultar Resumen diario de movimientos por cada grúa asociada al buque cerrado.			Si
Consultar Resumen de productividad de las grúas asociadas al buque, agrupadas por fecha, hora y grúa.			Si

FIGURA 13.3: Matriz de control de autorización de accesos del caso de estudio.

### 13.5.6 Prototipos para Smartphone y PC del caso de estudio

Las figuras 13.4 y 13.5 presentan los prototipos para dispositivos celulares y PC diseñados al aplicar el proceso de Elicitación de RNF Web, de la subcaracterística de calidad *Atractividad*.

#### Smartphone

BUQUE 1	BUQUE 2
GVR = <b>65</b> CI = <b>3</b>	GVR = <b>65</b> CI = <b>3</b>
FPLP = <b>55</b>	FPLP = <b>55</b>
P1 GCR = <b>21</b> Ritmo <b>20</b>	P1 GCR = <b>21</b> Ritmo <b>20</b>
P2 GCR = <b>14</b> Ritmo <b>21</b>	P2 GCR = <b>14</b> Ritmo <b>21</b>
P3 GCR = <b>16</b> Ritmo <b>18</b>	P3 GCR = <b>16</b> Ritmo <b>18</b>
P4 GCR = <b>14</b> Ritmo <b>20</b>	P4 GCR = <b>14</b> Ritmo <b>20</b>
Demoras = <b>1.5 hs</b> +	Demoras = <b>1.5 hs</b> +
Mov. Realizados <b>1640</b>	Mov. Realizados <b>1640</b>
Mov. Restantes <b>405</b>	Mov. Restantes <b>405</b>
Hora de Inicio <b>14:35hs</b>	Hora de Inicio <b>14:35hs</b>
ETC estimada <b>21:50hs</b>	ETC estimada <b>21:50hs</b>

FIGURA 13.4: Prototipos para dispositivos celulares del caso de estudio.

**PC**

**BUQUE 1 (Icono de Buque)**

GVR = **65**    CI = **3**  
 FPLP = **55**  
 P1 GCR = **21** Ritmo **20**  
 P2 GCR = **14** Ritmo **21**  
 P3 GCR = **16** Ritmo **18**  
 P4 GCR = **14** Ritmo **20**  
 Demoras = **1.5 hs** +  
 Mov. Realizados **1640**  
 Mov. Restantes **405**  
 Hora de Inicio **14:35hs**  
 ETC estimada **21:50hs**

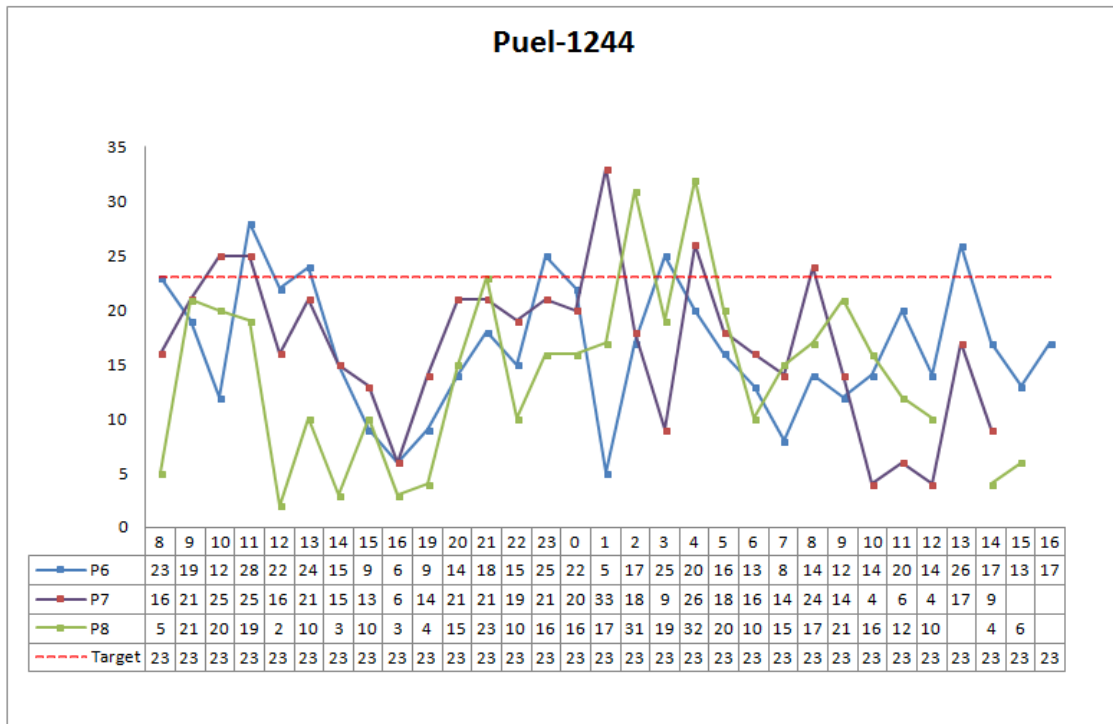
<b>BUQUE 1 (Icono de Buque)</b>				
	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>
Mov. Restantes	50	25	40	35
<b>Total</b>	<b>238</b>	<b>250</b>	<b>246</b>	
18	14	20	14	
17	12	18	12	
16	18	18	18	
15	22	22	22	
14	19	19	19	
13	21	21	25	
12	22	22	26	
11	22	22	22	
10	25	25	25	
9	22	22	22	
8	21	21	21	
7	20	20	20	

FIGURA 13.5: Prototipos para PC del caso de estudio.

**13.5.7 Prototipo de gráfico de actividad de pórticos del caso de estudio**

La figuras 11.6 muestra un prototipo para diseñado para visualizar la actividad de pórticos, el cual surgió al aplicar el proceso de Elicitación de RNF Web, de la subcaracterística de calidad *Atractividad*.

**Grafico (Ejemplo)**



**Observaciones :** Los colores de la serie que representan a los Porticos deben ser los mismos que utiliza Sparcs.

FIGURA 13.6: Prototipo para gráfico de actividad de pórticos del caso de estudio.

### 13.5.8 Matriz de RNF 1.1 - Versión intermedia del caso de estudio

El archivo 4. *Matriz de RNF- Caso-v1.1.xls* muestra el estado de la matriz al inicio del proceso de Especificación de RNF Web aplicado al caso de estudio.

### 13.5.9 Matriz de RNF 1.2 – Versión Definitiva del caso de estudio

El archivo 5. *Matriz de RNF- Caso-v1.2.xls* muestra el estado de la matriz al finalizar el proceso de Especificación de RNF Web aplicado al caso de estudio.

### 13.5.10 Plantilla de Especificación de RNF Web

El archivo 6. *Plantilla de Especificacion RNF Web-v1.0.xls*, muestra la plantilla obtenida al finalizar el proceso de Especificación de RNF Web aplicado al caso de estudio.