



**UNIVERSIDAD DE CARABOBO**  
**FACULTAD EXPERIMENTAL DE CIENCIAS Y TECNOLOGÍA**  
**DIRECCIÓN DE POSTGRADO**  
**ESPECIALIZACIÓN EN DESARROLLO DE SOFTWARE**

---



***INGENIERÍA DE MÉTODOS PARA DESARROLLAR  
APLICACIONES WEB***

AUTOR: Dely Maybel Gil Alvarez  
TUTOR: Judith Barrios Albornoz

Trabajo de Grado presentado  
para optar al título de Especialista

Valencia, 12-12-2020



### ACTA DE VEREDICTO DE TRABAJO DE ESPECIALIZACIÓN

Quienes suscribimos, profesores, Judith Barrios Albornoz, titular de la cédula de identidad número V- 8.003.287, Dinarle Ortega, titular de la cédula de identidad número V- 8.611.660, Yumaira Machuca Cohen, titular de la cédula de identidad número V- 6.437.707, integrantes del jurado designado por el Consejo de Postgrado de la Facultad Experimental de Ciencias y Tecnología de la Universidad de Carabobo, en su reunión ordinaria virtual N° 09/2020 de fecha 16/12/2020, con el oficio N° DPG-005A-2021, para considerar y evaluar el Trabajo de Especialización que lleva por título INGENIERÍA DE MÉTODOS PARA DESARROLLAR APLICACIONES WEB presentado por la Ing. Dely Maybel Gil Alvarez, titular de la cédula de identidad número V-9.476.671, bajo la tutoría académica de la profesora Judith Barrios Albornoz, cédula de identidad número V- 8.003.287, como requisito para optar al título de Especialista en Desarrollo de Software, dejamos constancia de lo siguiente:

Una vez leído el trabajo por cada uno de los integrantes del jurado, se convocó a la mencionada estudiante para la defensa pública virtual de su trabajo el día 23 de marzo 2021 a las 3 pm. a través de la aplicación Zoom. Debido a graves inconvenientes de conexión el día y la hora pautados, el jurado evaluador en pleno decidió que la estudiante grabara su defensa oral en video formato mp4 y lo enviara al correo electrónico de cada uno de los jurados para su evaluación individual. La sesión de preguntas y comentarios sobre la presentación oral se pautó para el día viernes 26 de marzo 2021 a las 9h45 am., a través de una llamada grupal con la aplicación WhatsApp. La sesión se desarrolló satisfactoriamente y la estudiante contestó apropiadamente cada una de las preguntas realizadas por el jurado evaluador.

Finalizada la sesión de preguntas, la deliberación del jurado se apoyó en las siguientes razones:

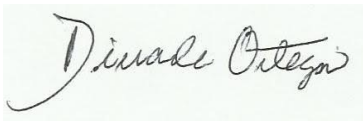
- El método propuesto es innovador, conceptualmente correcto y cubre completamente los requisitos académicos establecidos para este trabajo.
- El método propuesto es completo e integra adecuadamente elementos del contexto técnico y social requerido para la formación y práctica de profesionales en el área de desarrollo de software.
- La propuesta metodológica incluye, además, una guía para apoyar el proceso de instanciación y utilización del método a proyectos reales.
- El método propuesto constituye un aporte original para la Universidad de Carabobo, permitiendo no solo fortalecer y avanzar la investigación en el área de la Ingeniería de Software, sino por su potencial aplicabilidad para la formación de los futuros profesionales del área.

En consecuencia y conforme a lo dispuesto en las normas de Postgrado de la Facultad Experimental de Ciencias y Tecnología de la Universidad de Carabobo, el jurado emite unánimemente el veredicto de APROBADO al trabajo presentado por la Ing. Dely Gil Alvarez.

En fe de lo cual levantamos y firmamos la presente acta veredicto a los 26 días del mes de marzo de 2021.



Prof. Judith Barrios Albornoz  
CI. V- 8.003.287  
Coordinadora del Jurado  
Facultad de Ingeniería - Universidad de Los Andes



Prof. Dinarle Ortega  
CI: V- 8.611.660  
FACYT (UC)



Prof. Yumaira Machuca Cohen  
CI: V- 6.437.707  
UPTVAL-Valencia



UNIVERSIDAD DE CARABOBO  
FACULTAD EXPERIMENTAL DE CIENCIAS Y TECNOLOGÍA



DIRECCIÓN DE POSTGRADO  
ESPECIALIZACIÓN EN DESARROLLO DE *SOFTWARE*

---

---

## CARTA ACEPTACIÓN DEL TUTOR

Bárbula, 06 de Enero del año 2017

Ciudadano Prof. Dinarle Ortega.

Coordinador del Programa de Especialización en Desarrollo de *Software*.

Por este medio de la presente le informo que he aceptado asesorar el trabajo de investigación cuyo autor es: Dely Maybel Gil Álvarez, titular de la Cédula de Identidad N°: V-9.476.671, el cual lleva por título: Ingeniería de Métodos para desarrollar aplicaciones Web y está enmarcado en la línea de investigación: Ingeniería de *Software* y Sistemas de Información del Programa de Especialización en Desarrollo de *Software*

Atentamente,

Dely Maybel Gil Álvarez  
e-mail: [gilmaybel@gmail.com](mailto:gilmaybel@gmail.com)

Judith del Rosario Barrios Albornoz  
C.I. V- 8.003.287  
e-mail: [judith@ula.ve](mailto:judith@ula.ve)

## DEDICATORIA

*Te dedico a ti **Jehová**, este triunfo, porque eres la roca que sustenta mi vida; manantial de sabiduría para lograr esta victoria. Gracias Jehová por elegirme y siempre estar conmigo en el camino que has decidido para mí, dándome fuerzas por más obstáculos que existan y por rodearme de personas necesarias en mi vida.*

***A mis padres** por ser ejemplo de constancia, dedicación y amor.*

***A mi hijo Luis Alejandro**, porque eres mi todo después de Cristo; por el apoyo brindado en cada momento, por tus palabras de aliento cuando creía que no terminaría, por las fuerzas para conseguir este triunfo; triunfo que también es tuyo. Te amo, mi hijo adorado.*

***A toda mi familia** que siempre ha estado presente tanto físicamente como en oración. En especial a mi hermana **Giobely** que partió sin decir adiós y su vida me enseñó que las metas siempre hay que soñarlas y alcanzarlas con esfuerzo y perseverancia.*

## AGRADECIMIENTO

*Primeramente a Dios mi amigo fiel; por su compañía en el caminar diario; sin Él nada.*

*A mis compañeros de la Especialización por la constancia y apoyo incondicional para lograr esta meta.*

*A cada uno de los participantes en el desarrollo de esta investigación. Gracias por su dedicación y paciencia.*

*A los profesores Luis Giménez, Betty Medina y Yumaira Machuca por la valiosa colaboración brindada en asesorías para el desarrollo de este trabajo.*

*A la Profesora Dinarle por sus palabras de aliento, dedicación e impulso constante para culminar este trabajo.*

*Y en especial a mi tutora Profesora Judith Barrios por su persistencia, su paciencia, apoyo incondicional, su interés, su motivación y por impartir tantos conocimientos. Ha sido un honor y privilegio contar con su persona.*

*Y a todos aquellos que de una u otra forma contribuyeron con este logro.*

*Mil bendiciones a todos.*

***Dely Gil***



**UNIVERSIDAD DE CARABOBO**  
**FACULTAD EXPERIMENTAL DE CIENCIAS Y TECNOLOGÍA**  
**DIRECCIÓN DE POSTGRADO**  
**ESPECIALIZACIÓN EN DESARROLLO DE SOFTWARE**

---

---



**TITULO DEL TRABAJO:**

**INGENIERÍA DE MÉTODOS PARA DESARROLLAR APLICACIONES WEB**

**RESUMEN**

Uno de los principales problemas en el desarrollo de aplicaciones web en el ámbito académico venezolano es la selección de una metodología de desarrollo ajustada a las características del grupo, al tiempo disponible para el desarrollo y, que describa las actividades, las técnicas, las herramientas y los productos a generar, en cada una de las diferentes etapas del proceso de desarrollo. En tal sentido, la autora de esta investigación, aplicando el marco de referencia propuesto por Mendoza & Barrios (2004) y los principios de Ingeniería de Métodos a su Metodología “Creación de Sitios Web” (Gil, 2013), busca solventar la problemática antes planteada. Este trabajo está enmarcado dentro del tipo de modalidad de proyecto de Investigación Tecnológica con un diseño de investigación de campo no experimental y documental. El Método fue denominado Método Académico para el desarrollo de Sistemas de Información Web (MASIWeb), y está conformado por el modelo de Actores que propone una estructura ajustada a desarrollos de grupos pequeños; el modelo de Procesos que define, explícitamente, las actividades del desarrollo y, el modelo de Productos que detalla los productos a generar garantizando una guía completa al equipo de trabajo. Las características principales del método son: (1) es balanceado, proporciona un balance entre agilidad y disciplina; (2) cubre todo el ciclo de desarrollo del *software*, desde al análisis hasta la instalación; (3) involucra los procesos técnicos, de gestión y de soporte; (4) se basa en procesos del PMBOK; (5) incluye algunas de las mejores prácticas de ingeniería de *software* como desarrollo iterativo y versionado, gestión de proyectos, gestión de requisitos y modelado visual. El método fue validado mediante la instanciación del mismo en situaciones potenciales de aplicación: (1) Desarrollo de *software* ejecutando procesos técnicos y (2) Desarrollo de *software* ejecutando procesos técnicos y de administración de proyectos.

**Palabras Clave:** Método de desarrollo, Aplicación Web, Ingeniería de Métodos, Ingeniería Web.



UNIVERSIDAD DE CARABOBO  
FACULTAD EXPERIMENTAL DE CIENCIAS Y TECNOLOGÍA  
DIRECCIÓN DE POSTGRADO  
ESPECIALIZACIÓN EN DESARROLLO DE *SOFTWARE*



---

---

**TITULO DEL TRABAJO:**

**METHOD ENGINEERING TO DEVELOP WEB APPLICATIONS**

**ABSTRACT**

One of the main problems in the development of web applications in the Venezuelan academic field is the selection of a development methodology that adjusts to the characteristics of the group, in the time available for development, and that describes the activities, techniques, tools and products to be generated at the different stages of development. In this sense, the author of this research applying the reference framework proposed by Mendoza & Barrios (2004) and the principles of Method Engineering to the Methodology “Creation of Web Sites” (Gil, 2013), seeks to solve the problem mentioned before. This work is framed within the type of Technological Research project modality with a non-experimental and documentary field research design. The Method was called the Academic Method for the Development of Web Information Systems (MASIWeb), which contains the Actor Model that proposes a structure adjusted to small group development; the Process Model that explicitly defines the development activities, and the Product Model that details the products to be generated, guaranteeing a complete guide to the work team. The main characteristics of the method are: (1) it is balanced, because it provides a balance between agility and discipline; (2) it covers the entire *software* development cycle, from analysis to installation; (3) it involves the technical, management and support processes; (4) It is based on PMBOK process; (5) it includes some of the best *software* engineering practices such as iterative and versioned *software* development, project management, requirement management, and visual modeling. The method was validated by instantiating it in potential application situations: (1) *Software* developments executing technical processes and (2) *Software* developments executing technical processes and project management.

Keywords: Development method, Web Applications, Method Engineering, Web Engineering



## ÍNDICE GENERAL

<i>ÍNDICE GENERAL</i> .....	VII
<i>ÍNDICE DE TABLAS</i> .....	XI
<i>ÍNDICE DE FIGURAS</i> .....	XIII
<i>INTRODUCCIÓN</i> .....	15
<i>CAPÍTULO 1. EL PROBLEMA Y MARCO METODOLÓGICO DE LA SOLUCIÓN</i> .....	18
1.1. Planteamiento del Problema.....	18
1.2. Objetivo General.....	22
1.3. Objetivos Específicos .....	22
1.4. Justificación de la Investigación .....	22
1.5. Tipo de la Investigación .....	23
1.6. Diseño de la Investigación.....	25
1.7. Marco de Trabajo .....	26
<i>CAPÍTULO 2. MARCO TEÓRICO</i> .....	29
2.1. Antecedentes.....	29
2.2. Bases Teóricas.....	30
2.2.1. Desarrollo de <i>Software</i> .....	31
2.2.2. Aplicaciones Web.....	32
2.2.3. Ingeniería de Métodos .....	35
<i>CAPÍTULO 3. METODOLOGÍA “CREACIÓN DE SITIOS WEB”</i> .....	38
3.1. Objetivo .....	38
3.2. Usuarios Potenciales.....	38
3.3. Fases.....	39
3.3.1. Fase de Análisis.....	39
3.3.2. Fase de Planificación .....	39

3.3.3. Fase de Contenido.....	40
3.3.4. Fase de Diseño.....	40
3.3.5. Fase de Programación.....	41
3.3.6. Fase de Testeo.....	41
3.3.7. Fase de Mercadeo y Publicidad.....	41
3.4. Estructura.....	43
3.5. Aplicación de la metodología en proyectos reales.....	43
3.6. Modelos según la Ingeniería de Método.....	43
3.6.1. Modelo de Procesos.....	43
3.6.2. Modelo de Productos.....	46
3.6.3. Modelo de Actores.....	48
<i>CAPÍTULO 4. EVALUACIÓN DE LA METODOLOGÍA “CREACIÓN DE SITIOS WEB”.....</i>	<i>50</i>
4.1. Definición del Marco de Referencia propuesto por Mendoza & Barrios (2004).....	50
4.1.1. La vista de dominio.....	51
4.1.2. La vista de uso.....	51
4.1.3. La vista de producto.....	52
4.1.4. La vista de procesos.....	53
4.2. Aplicación del Marco de Referencia para la evaluación de la Metodología “Creación de Sitios Web”.....	55
4.2.1. La vista de dominio.....	55
4.2.2. La vista de uso.....	56
4.2.3. La vista de producto.....	57
4.2.4. La vista de procesos.....	58
4.3. Análisis de Bondades y Debilidades de la Metodología “Creación de Sitios Web”.....	60
<i>CAPÍTULO 5. MÉTODO DE DESARROLLO PARA APLICACIONES WEB BASADO EN INGENIERÍA DE MÉTODOS.....</i>	<i>62</i>
5.1. Características y herramientas conceptuales que solventan las debilidades.....	62

5.2. Caracterización del método .....	64
5.3. Perspectivas del posicionamiento del método.....	65
5.3.1. Características del método .....	65
5.3.2. Componentes del método .....	66
5.3.3. Modelo de Procesos .....	67
5.3.3.1. Procesos de Gestión .....	70
5.3.3.2. Procesos de Soporte .....	73
5.3.3.3. Proceso Técnico de Análisis y definición de requisitos .....	75
5.3.3.4. Proceso Técnico de Diseño.....	80
5.3.3.5. Proceso Técnico de Construcción & Integración .....	87
5.3.3.6. Proceso Técnico de Pruebas.....	93
5.3.3.7. Proceso Técnico de Lanzamiento .....	95
5.3.4. Modelo de Productos .....	97
5.3.4.1. Modelo de Procesos de Negocios (MPN).....	100
5.3.4.2. Documento Especificación de Requisitos de <i>Software</i> (ERS).....	101
5.3.4.3. Documento de Diseño (DD).....	102
5.3.4.4. Documento de Pruebas (DP).....	105
5.3.4.5. Documento Manual de Usuario del <i>Software</i> (MUS) .....	107
5.3.4.6. Documento de Lanzamiento (DL).....	107
5.3.4.7. Documento de Instalación del <i>Software</i> (DIS).....	108
5.3.4.8. Formato de Entrega del <i>Software</i> (FES) .....	108
5.3.4.9. Documento Visión del Proyecto (DVP).....	108
5.3.4.10. Documento Plan del Proyecto (DPP) .....	109
5.3.4.11. Documento de Seguimiento del Proyecto (DSP).....	110
5.3.4.12. Documento de Gestión de Riesgos (DGR).....	110
5.3.5. Modelo de Actores .....	111
5.3.6. Glosario de términos .....	114
5.4. Directivas para la instanciación .....	116
5.4.1. Situaciones potenciales del método.....	116
5.4.2. Flujos de trabajo de las situaciones potenciales .....	117
<b>CONCLUSIONES</b> .....	<b>123</b>

<i>RECOMENDACIONES</i> .....	128
<i>REFERENCIAS BIBLIOGRÁFICAS</i> .....	129
<i>ANEXOS</i> .....	137
<i>ANEXO A</i> .....	138
Proyectos Socio Tecnológico (PST) disponibles en la Biblioteca “Centro de Información y Documentación Biblioteca Ing. Nelson Rivero” de la UPTVal, Trayecto III, Período 1-2017 .....	138
<i>ANEXO B</i> .....	140
Medias y desviación estándar de los grupos de criterios del QUIS (Shneiderman, 1986b,Chin, Diehl & Norman, 1988). .....	140

## ÍNDICE DE TABLAS

Tabla 2.1. <i>Caracterización de las aplicaciones web</i> .....	34
Tabla 3.1. <i>Modelo de Productos intermedios de la metodología Gil (2013)</i> .....	47
Tabla 3.2. <i>Productos intermedios de la metodología Gil (2013)</i> . ....	48
Tabla 4.1. <i>Vista de dominio</i> . ....	51
Tabla 4.2. <i>Vista de uso</i> .....	52
Tabla 4.3. <i>Vista del producto</i> .....	53
Tabla 4.4. <i>Vista del proceso</i> .....	54
Tabla 4.5. <i>Evaluación de la vista de dominio</i> . ....	55
Tabla 4.6. <i>Evaluación de la vista de uso</i> . ....	56
Tabla 4.7. <i>Evaluación de la vista del producto</i> . ....	57
Tabla 4.8. <i>Evaluación de la vista del proceso (a)</i> .....	58
Tabla 4.9. <i>Evaluación de la vista del proceso (b)</i> .....	59
Tabla 4.10. <i>Análisis Fortalezas y Debilidades de la metodología Gil (2013) (a)</i> . ....	60
Tabla 4.11. <i>Análisis Fortalezas y Debilidades de la metodología Gil (2013) (b)</i> . ....	61
Tabla 5.1. <i>Clasificación de los PST</i> .....	64
Tabla 5.2. <i>Descripción del proceso de Gestión del Proyecto</i> .....	71
Tabla 5.3. <i>Entradas y salidas del proceso de Gestión del Proyecto</i> . ....	72
Tabla 5.4. <i>Descripción del proceso de soporte: Gestión de Riesgos</i> .....	74
Tabla 5.5. <i>Entradas y salidas del proceso de Gestión de Riesgos</i> . ....	75
Tabla 5.6. <i>Descripción del proceso de Análisis y definición de requisitos (a)</i> .....	76
Tabla 5.7. <i>Descripción del proceso de Análisis y definición de requisitos (b)</i> . ....	77
Tabla 5.8. <i>Descripción del proceso de Análisis y definición de requisitos (c)</i> . ....	78
Tabla 5.9. <i>Entradas y salidas del proceso de Análisis y definición de requisitos (a)</i> .79	
Tabla 5.10. <i>Entradas y salidas del proceso de Análisis y definición de requisitos (b)</i> . .....	80
Tabla 5.11. <i>Descripción del proceso de Diseño (a)</i> .....	81
Tabla 5.12. <i>Descripción del proceso de Diseño (b)</i> .....	82
Tabla 5.13. <i>Los 10 principios heurísticos de Nielsen (1994)</i> .....	84
Tabla 5.14. <i>Aplicación de la evaluación heurística de Shneiderman (1986a)</i> .....	85
Tabla 5.15. <i>Aplicación de los principios heurísticos de autor (año)</i> . ....	86

Tabla 5.16. <i>Entradas y salidas del proceso de Diseño.</i> .....	86
Tabla 5.17. <i>Descripción del proceso de Construcción &amp; Integración (a)</i> .....	87
Tabla 5.18. <i>Descripción del proceso de Construcción &amp; Integración(b)</i> .....	88
Tabla 5.19. <i>Criterios del cuestionario de facilidad de uso de Travis (2012)</i> .....	89
Tabla 5.20. <i>Criterios del cuestionario de facilidad de uso SIRIUS (2011)</i> .....	90
Tabla 5.21. <i>Criterios del cuestionario de facilidad de uso QUIS (Shneiderman, 1986b,Chin, Diehl &amp; Norman, 1988)</i> .....	91
Tabla 5.22. <i>Cuestionario para la satisfacción de la interfaz de usuario (QUIS; Shneiderman, 1986b,Chin, Diehl &amp; Norman, 1988).</i> .....	91
Tabla 5.23. <i>Entradas y salidas del proceso de Construcción &amp; Integración.</i> .....	93
Tabla 5.24. <i>Descripción del proceso de Pruebas.</i> .....	94
Tabla 5.25. <i>Entradas y salidas del proceso de Pruebas.</i> .....	95
Tabla 5.26. <i>Descripción del proceso de Lanzamiento (a).</i> .....	96
Tabla 5.27. <i>Descripción del proceso de Lanzamiento (b).</i> .....	97
Tabla 5.28. <i>Entradas y salidas del proceso de Lanzamiento.</i> .....	97
Tabla 5.29. <i>Productos Intermedios Entregables del documento ERS.</i> .....	102
Tabla 5.30. <i>Productos Intermedios Entregables del Documento de Diseño.</i> .....	103
Tabla 5.31. <i>Productos Intermedios Entregables del Documento de Pruebas.</i> .....	105
Tabla 5.32. <i>Productos Finales Entregables del proceso de Pruebas.</i> .....	105
Tabla 5.33. <i>Productos Intermedios Entregables del documento de Lanzamiento.</i> .108	
Tabla 5.34. <i>Productos Finales Entregables del documento de Lanzamiento.</i> .....	108
Tabla 5.35. <i>Roles y responsabilidades de los actores.</i> .....	113
Tabla 5.36. <i>Glosario de Términos (a)</i> .....	114
Tabla 5.37. <i>Glosario de Términos (b)</i> .....	115

## ÍNDICE DE FIGURAS

<i>Figura 3.1.</i> Componentes de la metodología “Creación de sitios Web”.....	42
<i>Figura 3.2.</i> Componentes de la metodología Gil (2013) según Ingeniería de Métodos.....	45
<i>Figura 3.3.</i> Modelo de Procesos de la metodología Gil (2013). .....	46
<i>Figura 3.4.</i> Modelo de Actores de la metodología Gil (2013). .....	49
<i>Figura 5.1.</i> Modelo de Procesos del método MASIWeb.....	67
<i>Figura 5.2.</i> Flujo de trabajo del Modelo de Procesos del método MASIWeb. ....	69
<i>Figura 5.3.</i> Estructura del modelo de procesos del método MASIWeb. ....	70
<i>Figura 5.4.</i> Diagrama de jerarquía del proceso de Gestión del Proyecto.....	70
<i>Figura 5.5.</i> Diagrama de jerarquía del proceso de Gestión de Riesgos.....	73
<i>Figura 5.6.</i> Diagrama de jerarquía del proceso Análisis y definición de requisitos. ...	76
<i>Figura 5.7.</i> Diagrama de jerarquía del proceso de Diseño.....	80
<i>Figura 5.8.</i> Diagrama de jerarquía del proceso de Construcción & Integración.....	87
<i>Figura 5.9.</i> Diagrama de jerarquía del proceso de Pruebas.....	93
<i>Figura 5.10.</i> Diagrama de jerarquía del proceso de Lanzamiento. ....	96
<i>Figura 5.11.</i> Modelo de Productos del método MASIWeb. ....	99
<i>Figura 5.12.</i> Modelo de Procesos de Negocios (MPN). ....	100
<i>Figura 5.13.</i> Documento de Especificación de Requisitos de <i>Software</i> (ERS). ....	101
<i>Figura 5.14.</i> Documento de Diseño (DD). ....	104
<i>Figura 5.15.</i> Documento de Pruebas (DP). ....	106
<i>Figura 5.16.</i> Documento de Lanzamiento (DL). ....	107
<i>Figura 5.17.</i> Documento Visión del Proyecto (DVP). ....	109
<i>Figura 5.18.</i> Documento Plan del Proyecto (DPP). ....	109
<i>Figura 5.19.</i> Documento Seguimiento del Proyecto (DSP). ....	110
<i>Figura 5.20.</i> Documento Gestión de Riesgos (DGR). ....	111
<i>Figura 5.21.</i> Modelo de Actores del método MASIWeb. ....	112
<i>Figura 5.22.</i> Proceso de instanciación del método MASIWeb. ....	116
<i>Figura 5.23.</i> Modelo principal de situación potencial de procesos técnicos. ....	118
<i>Figura 5.24.</i> Modelo principal de situación potencial de administración de proyectos. ....	119

*Figura 5.25.*Diagrama de flujo de Situación Potencial de Procesos Técnicos. ....120

*Figura 5.26.*Diagrama de flujo de Situación Potencial de Administración de  
Proyectos. ....120

*Figura 5.27.*Diagrama de flujo de continuidad de los procesos potenciales (a). ....121

*Figura 5.28.*Diagrama de flujo de continuidad de los procesos potenciales (b). ....122



## INTRODUCCIÓN

Estamos ante una sociedad de la información global emergente, que depende cada vez más de la creación, administración y distribución de información a través de internet. Es así, como la demanda de aplicaciones Web se hace cada vez más imprescindible y compleja, debido a lo vertiginoso de los cambios en un ambiente cada vez más exigente.

Las organizaciones están adoptando metodologías para el desarrollo de *software* existentes, las adecuan a sus exigencias, empleando procesos que puedan ajustarse a los recursos de desarrollo, es decir, construyen una metodología propia.

En tal sentido, existen metodologías orientadas al desarrollo de proyectos de pequeña envergadura, que responden a requerimientos muy cambiantes denominadas metodologías ágiles y otras, diseñadas para proyectos grandes, con requerimientos resistentes a cambios denominadas metodologías tradicionales. La selección y adaptación del tipo de metodología depende en gran parte de la experiencia y del conocimiento del equipo de trabajo, además, de la validez y la confiabilidad de la documentación que sustenta la metodología. Tal situación es más compleja cuando se refiere a su aplicación en el ámbito académico, la cual está orientada al aprendizaje del estudiante, quien no cuenta con una experiencia sólida en el área de desarrollo de *software*.

La mayoría de las metodologías disponibles para el desarrollo de *software* de mediana y pequeña escala, carecen de un modelo de proceso y de métodos que posean las siguientes características: fácil de aprender, que permita una aplicación rápida, eficaz y adaptable; y que reúna o contemple

los conceptos más recientes de la Ingeniería del *Software* (Montilva, Hamzan & Gharawi, 2000).

En atención a lo antes mencionado, surge la necesidad de diseñar un método fundamentado en la Ingeniería de Métodos (Brikkemper, 1995), que incluya pertinentemente las prácticas de Ingeniería de *Software*; este método debe guiar al equipo de trabajo el desarrollo de aplicaciones web mediante procedimientos, técnicas y herramientas en cada una de las fases del ciclo.

Esta propuesta se fundamenta en la adaptación de la metodología “Creación de Sitios Web” (Gil, 2013), a las necesidades planteadas en el ámbito académico, basándose en la tendencia de Ingeniería de Métodos; tal como lo señala Ralyté (2001) en referencia a reutilizar o extender un método existente.

Para el desarrollo de ésta investigación se desarrolla el siguiente esquema estructurado en capítulos, para su mejor presentación:

Capítulo I: está referido al planteamiento del problema, los objetivos y las razones que justifican la propuesta. Además, se especifica el tipo de investigación, en el cual está enmarcado el proyecto y se definen las fases del marco de trabajo.

Capítulo II: contiene algunos de los antecedentes que permiten abordar y profundizar en el área de Ingeniería de Métodos. Luego, se definen los términos más relevantes en cuanto al tema de investigación.

Capítulo III: señala los aspectos generales de la metodología “Creación de sitios Web” Gil (2013), sus objetivos, los usuarios potenciales,

las fases, la estructura y la aplicación en proyectos reales; luego se construyen los tres (3) modelos según la Ingeniería de Métodos (modelo de productos, modelo de procesos y modelo de actores).

Capítulo IV: describe el proceso de evaluación de la metodología de Gil (2013) utilizando el marco de referencia de Mendoza & Barrios (2004) y, como resultado de la evaluación anterior, se realiza un análisis detallando las debilidades y fortalezas de dicha metodología.

Capítulo V: presenta la propuesta del método basada en Ingeniería de Métodos, especificando en detalle los modelos de procesos, de productos y de actores, en los cuales está fundamentada.

Por último, se presentan las conclusiones, recomendaciones y anexos del trabajo realizado.

## **CAPÍTULO 1. EL PROBLEMA Y MARCO METODOLÓGICO DE LA SOLUCIÓN**

El presente capítulo describe la problemática existente, el planteamiento de los objetivos y las razones que justifican el desarrollo de la propuesta. También se define el tipo de investigación en el cual está enmarcado el estudio y las actividades de acción para alcanzar los objetivos planteados.

### **1.1. Planteamiento del Problema**

El objetivo de la Ingeniería de *Software* es la aplicación de un enfoque sistemático y disciplinado al desarrollo, operación y mantenimiento de *software*, mediante el uso de técnicas, métodos, metodologías, herramientas y estándares (IEEE, 1993).

En tal sentido, la selección de una metodología es una de las actividades más importantes en la ejecución exitosa del desarrollo de *software*, ya que la aplicación de la misma implica mejoras en el proceso, en el producto y en la satisfacción para el cliente. Al respecto, Brito (2009) presenta las siguientes mejoras:

- Todos los integrantes del equipo del proyecto trabajan bajo un marco común, el cual es apoyado por procedimientos, guías, actividades y tareas definidas.
- Se asegura que los productos cumplan con los objetivos de calidad propuestos y con las expectativas del cliente.

- Se obtiene la detección temprana de errores.
- El cliente percibe el orden en los procesos, permitiéndole hacerle seguimiento a través de la evolución del proyecto.

A pesar de los beneficios que genera el uso de una metodología, Sommerville (2005) afirma que éstas suelen ser artesanales. Por otro lado, se tiene conocimiento, según estudios realizados en el proyecto METHODIUS (2009), que la mayoría de las empresas venezolanas dedicadas al desarrollo de *software*, utilizan métodos propios de una manera empírica o inapropiada.

En este contexto, Harmsen & Saeki (1996) afirman que las metodologías contienen un conjunto de métodos, procedimientos y pautas que en realidad nunca se pueden seguir literalmente. Los procesos o las pautas a veces se agregan, se modifican o se ignoran para adaptarse a las circunstancias especiales, como tecnología, experiencia de desarrollo, la aplicación de factores internos y de factores externos.

Ahora bien, en el ámbito académico y de modo particular en los sistemas de información web, la selección y aplicación de la metodología adquiere mayor complejidad ya que es necesario que los estudiantes de pregrado posean sólidos conocimientos, y considerable experiencia para emplear la que se adecue a las características de este tipo de sistemas.

Sin embargo, si se aplica una metodología y no existe ningún enfoque sistemático y coordinado para establecer los métodos de trabajo, difícilmente

puede considerarse una metodología. Para que tal enfoque cumpla con esas exigencias requiere de la Ingeniería de Métodos (Odell, 1996).

La Ingeniería de Métodos es la disciplina para mejorar o construir nuevos métodos (Harmsen & Saeki, 1996). A su vez, un método indica cómo crear técnicamente un *software* (Pressman, 2005), a través de pasos necesarios, junto con las herramientas requeridas y los productos anticipados (Odell, 1996).

Por consiguiente, existe una necesidad apremiante de crear un método basado en los principios de Ingeniería de Métodos, que guíe al estudiantado de una manera ordenada y disciplinada para obtener éxito en el desarrollo, empleo y mantenimiento de los sistemas web.

En este orden de ideas, la investigadora como observadora participativa, está en la búsqueda de adaptar su metodología “Creación de Sitios Web” (Gil, 2013), en función de la situación antes mencionada; la misma fue producto de la experiencia de un grupo de egresados de la Universidad Politécnica Territorial de Valencia (UPTVal) en el desarrollo de sus trabajos especiales de grado, cuando comenzó el auge de las aplicaciones web. Según este estudio, los alumnos seleccionaban las orientadas a los sistemas tradicionales o de hipermedia, las cuales no se adecuan a las características particulares de los sistemas web. Por lo tanto, surgió la necesidad de proponer una metodología ajustada a estos sistemas y al ámbito académico venezolano.

Las fases de la metodología son: Análisis, Planificación, Contenido, Diseño, Testeo, Mercadeo y Publicidad.

Las actividades en cada fase son:

- Fase I. Fase de Análisis: definición de objetivos del sitio, selección de usuarios, expectativas de los usuarios, expectativas de la organización.
- Fase II. Fase de Planificación: selección del *hardware*, selección del *software*, selección del equipo de trabajo, *bechmarking*, estructura de navegación, costos de inversión, beneficios a obtener.
- Fase III. Fase de Contenido.
- Fase IV. Fase de Diseño: usabilidad, accesibilidad.
- Fase V. Programación: base de datos, programación intermedia, interfaz
- Fase VI. Fase de Testeo: comprobación en navegadores, detectar los vínculos rotos, comprobar tiempo de descarga.
- Fase VII. Fase de Mercadeo y Publicidad: derechos internacionales del autor, registro de dominio, selección de servidor de alojamiento y publicidad.

Según la descripción de las fases y sus respectivas actividades, se formulan las siguientes interrogantes: La metodología “Creación de Sitios Web” ¿está bien definida, estructurada y documentada?, ¿cubre el ciclo de desarrollo de *software*?, ¿puede ser aplicada en el ámbito académico y empresarial?, ¿es independiente del entorno donde será aplicada?, ¿incluye o prescribe técnicas, herramientas, estándares y procedimientos, relacionados con el desarrollo de *software*?

De aquí deriva la necesidad de redefinir dicha metodología para solventar la problemática antes señalada.

## **1.2. Objetivo General**

Diseñar un Método Académico para el desarrollo de Sistemas de Información Web, que cubra el ciclo de desarrollo completo, fundamentado en la Ingeniería de Métodos y que incluya pertinentemente las mejores prácticas de Ingeniería del *Software*. Para lograr dicho objetivo se toma como base la metodología Gil (2013).

## **1.3. Objetivos Específicos**

- Evaluar la metodología de Gil (2013), en términos del marco de referencia propuesto por Mendoza & Barrios (2004) y definido según la Ingeniería de Métodos (Brikkemper, 1995).
- Caracterizar el método para el desarrollo de aplicaciones web, incluyendo como base una caracterización presentada por Mendoza & Barrios (2004).
- Diseñar un método para el desarrollo de aplicaciones web atendiendo los resultados de la evaluación anterior y de los conceptos de la Ingeniería de Métodos según Brikkemper (1995).
  - Definir los modelos de producto, proceso y actores del Método de desarrollo de Aplicaciones Web.

## **1.4. Justificación de la Investigación**

El trabajo de investigación genera distintos beneficios en diferentes áreas:

- Nivel Institucional: la creación de un método ajustado al nivel académico, garantiza a cualquier universidad venezolana de un banco de herramientas para el desarrollo de proyectos web.



- Nivel Social: el uso de este método permitirá que los estudiantes resuelvan los diferentes problemas de la comunidad desde el punto de vista informático, con el fin de garantizar un producto de alta calidad.
- Nivel Operativo: Desarrollar proyectos eficientemente e involucrar al personal docente y estudiantes en el campo de la investigación para la innovación y mejora de los métodos existentes para el desarrollo de aplicaciones web.
- Nivel Económico: El uso de este método permitirá al equipo de trabajo tener control sobre el desarrollo del mismo y no involucrar a terceros que generen algún tipo de inversión del soporte que se requiera.
- Nivel Tecnológico: Este método genera una contribución a cualquier investigador y/o desarrollador que quiera hacer uso del método o agregar valor al mismo para alcanzar los objetivos deseados.

### **1.5. Tipo de la Investigación**

La presente investigación se orienta a la modalidad de investigación denominada Investigación Tecnológica, la cual tiene como fin:

Obtener un conocimiento para lograr modificar la realidad en estudio, vinculando la investigación y la transformación, la cual trata de ir de las ideas a las acciones para generar bienes o servicios para facilitar la vida del hombre. Persigue un conocimiento práctico, que sea más un conjunto de instrucciones a seguir para transformar el objeto, que explicaciones teóricas respecto a las cualidades del mismo (García, 2007).

La investigación tecnológica parte del conocimiento de la Investigación Científica para transformar la realidad y así aportar soluciones incluyendo el proceso de invención, innovación, diseño o de desarrollo tecnológico.

Las etapas del proceso de Investigación Tecnológica según García (2007), corresponden a un conjunto de fases que no son excluyentes, ni necesarias y se encuentran en constante realimentación; éstas corresponden más a un diseño de carácter didáctico-explicativo que a un procedimiento que hay que seguir al pie de la letra. A continuación se describen las fases:

1. Observar: representa el punto de partida inevitable de cualquier proceso que tenga como preocupación la realidad, donde intervienen los conocimientos previos, la experiencia adquirida y la intuición del observador.
2. Determinar el problema: se determina qué es lo que ocurre o está presente y requiere atención.
3. Documentar: se refiere a reunir toda la información referente al problema a solucionar, para razonar con claridad y tomar decisiones acertadas.
4. Reflexionar: una vez recabados los datos de la fase anterior, ocurre la revisión, comprensión y análisis de los problemas; con el dominio del conocimiento es probable que se generen algunas posibles respuestas al problema.
5. Elaborar el Proyecto de Intervención: es la formulación del plan para llevar a cabo una intervención en la realidad, donde se indican las acciones, los recursos, los participantes, los responsables, los compromisos del investigador y de la organización, entre muchas cosas más, que se proponen para lograr la transformación deseada.

6. Valorar: es la acción que permite conocer la efectividad del proyecto que se desarrolla, mediante la presentación y discusión de la propuesta a todos los implicados o la experimentación de la propuesta.
7. Comunicar: se comunica cuál será la participación de los involucrados, especificando las responsabilidades y compromisos de cada persona.
8. Implementar: consiste en especificar cuándo se podrán en acción las diferentes actividades contempladas en el plan de trabajo, es recomendable dar a conocer la fecha de inicio, lo que permitirá valorar los logros, así como determinar si lo previsto se realiza en el tiempo acordado.
9. Seguimiento: se asegura que las actividades definidas en el plan de trabajo se están ejecutando como se tiene previsto, evitando de esta manera cualquier desviación que se pueda presentar a lo largo del desarrollo del proyecto.
10. Evaluación: permite determinar con precisión si los objetivos se logran convenientemente; generalmente tal acción deberá documentarse en un informe final, en el que se señalen las condiciones iniciales, el plan que se siguió y los logros obtenidos.

## **1.6. Diseño de la Investigación**

Según Balestrini (2006) define el diseño de la investigación como:

el plan o la estrategia global en el contexto de estudio propuesto, que permite orientar desde el punto de vista técnico, y guiar todo proceso de investigación, desde la recolección de los primeros datos hasta el análisis e interpretación de los

mismos en función de los objetivos definidos en la investigación (p.120).

Para el caso de la investigación se toma a consideración el tipo de diseño no experimental partiendo de la idea de que no se pretende llevar a cabo ningún tipo de experimento, y por consiguiente, no hay manipulación de variables, se busca sólo observar el fenómeno en estudio, elaborar un diagnóstico y posteriormente analizarlos para diseñar una propuesta viable; tal como lo afirma Hernández, Fernández y Baptista (2006), lo que se hace en la investigación no experimental es observar los fenómenos tal y como son, en su entorno natural para después analizarlos.

Con el objeto de emprender el proceso de observar al fenómeno en su propio ambiente es necesario definir el diseño en función del tipo de datos a ser recogidos; definiendo así para esta investigación, el diseño de campo y documental. Cuando se hace referencia al diseño de campo los datos de interés se recogen en forma directa de la realidad, mediante el trabajo concreto del investigador y su equipo, denominados estos datos como primarios; mientras que cuando se hace referencia a diseños documentales los datos son recolectados en otras investigaciones definiéndolos como datos secundarios(Sabino, 1995).

### **1.7. Marco de Trabajo**

En la investigación tecnológica se ejecuta un plan de intervención definido por acciones, recursos, responsables, entre otros. En esta investigación no se elabora dicho plan ya que la actividad de elaborar el proyecto es el desarrollo del trabajo de postgrado.

Tomando en cuenta las fases del tipo de investigación tecnológica y según la necesidad y características de este estudio, se describe el marco de trabajo seguido:

1. Determinar el problema: En esta fase se utiliza la técnica de observación de la autora como evaluadora participativa en los proyectos generados de la materia Ingeniería del *Software* para determinar qué es lo que ocurre o está presente y que requiere de una investigación para solventar la problemática.
2. Documentar: se realiza una revisión documental de lo que se ha escrito o investigado sobre el objeto de estudio y de un compendio de términos conceptuales que sirven de base para el entendimiento de este trabajo.
3. Elaborar el proyecto de investigación: es definido según las siguientes actividades de acción:
  - Realizar la uniformidad de términos para la representación de la metodología de Gil (2013) en los modelos de la Ingeniería de Métodos.
  - Evaluar la Metodología a través del marco de referencia de Mendoza & Barrios (2004).
  - Describir las debilidades y fortalezas de la metodología a través de la selección de una notación gráfica; como resultado de la evaluación en la actividad anterior.
  - Definir características y prácticas para solventar las debilidades y, producir mejoras.
  - Desarrollar la propuesta de la investigación.
  - Generar las directivas para la aplicación e instanciación del método.

4. Evaluación: se desarrolla el informe final donde se describirá si se cumplieron los objetivos de la investigación planteados inicialmente; detallando la situación actual, el procedimiento realizado y los logros obtenidos.

En este capítulo se presentó el planteamiento del problema, los objetivos y las fases del marco de trabajo según el tipo de investigación tecnológica.

## CAPÍTULO 2. MARCO TEÓRICO

Este capítulo se centra en la revisión de algunos antecedentes relacionados a la presente investigación, razón que permite abordar y profundizar el área en estudio contribuyendo a su enriquecimiento. Luego, se presenta definiciones, tales como desarrollo de *software*, aplicaciones web e Ingeniería de Métodos.

### 2.1. Antecedentes

El desarrollo de la presente investigación tiene fundamentos definidos en base a trabajos previos y en la literatura existente en relación al tema, razón que permite evaluar el grado de completitud, cobertura y pertinencia de la citada metodología de Gil (2013) y de la propuesta. Entre los trabajos presentados se destacan los siguientes:

Castillo, Barrios, Montilva & Rivero (2010) elaboraron el artículo: Conceptualización del proceso de Implementación de *software*: Perspectiva Ágil y Disciplinada. Este artículo presenta una propuesta conceptual de especificación de un modelo híbrido; especificando un proceso de implementación de *software* ágil y disciplinado. La propuesta integra, de modo complementario, las prescripciones típicas de modelos tradicionales o disciplinados con prácticas ágiles, buscando un equilibrio entre formalidad y agilidad que, mejore el desempeño y la productividad en el proceso de desarrollo, sin atentar contra la calidad del producto que se elabora.

Este trabajo se utiliza como marco de referencia para la conceptualización de la propuesta aquí presentada.

Lizardo (2013) elaboró la Tesis Especial de Grado titulada: Reutilización de Componentes de la suite *WATCH* a través del *Eclipse Process Framework (EPF) Composer*. Este trabajo tiene como finalidad representar los métodos de la suite *WATCH* con el *EPF Composer*, a través de la definición de meta-modelos. El aporte del trabajo de grado a la presente investigación es el conocimiento de los conceptos de Ingeniería de Métodos, la suite del método *WATCH* y los meta-modelos de producto, proceso y actores.

Mendoza (2004) realizó la Tesis de maestría intitulada: "Método Situacional para el desarrollo de Aplicaciones Web". En este trabajo se seleccionó un grupo de métodos, utilizados para el desarrollo de portales, comercio electrónico, negocios inteligentes, sistemas de información web (SIW) y gobiernos electrónicos; generando un modelo de proceso y un modelo de producto para cada dominio y, un modelo genérico de proceso y de productos. Para la evaluación de dichos métodos se definieron y utilizaron marcos de referencia propios de la Ingeniería de Métodos. Este trabajo contribuye con un marco de referencia para evaluar la metodología de Gil (2013).

## **2.2. Bases Teóricas**

En esta sección se presenta la perspectiva teórica que fundamenta esta investigación, la cual tiene como propósito suministrar un sistema coordinado y coherente de conceptos y proposiciones, que permitan abordar



el problema, a manera de establecer un marco teórico para la interpretación de los resultados del estudio planteado.

### **2.2.1. Desarrollo de *Software***

Según Pressman (2005) es el proceso mediante el cual se realizan actividades tanto estructurales como de sombrilla para obtener un producto de calidad que satisfaga a los usuarios. Este debe ser desarrollado bajo un enfoque adaptable al problema, al proyecto, al equipo y a la cultura organizacional.

Las actividades estructurales son: comunicación, planeación, modelado, construcción y despliegue. Estas son complementadas por las actividades sombrillas, las cuales se aplican a lo largo del desarrollo del proyecto y ayudan a administrar y controlar el avance, la calidad, el cambio y el riesgo.

Para Weitzenfeld (2004) el desarrollo de *software* es el conjunto de diferentes actividades comenzando desde la formulación del problema, seguido por la especificación de requisitos, análisis, diseño, implementación y codificación, integración, pruebas del *software*, documentación y como actividad final el mantenimiento del sistema; con el objetivo de producir *software* según las necesidades del cliente.

Sommerville (2005), afirma que el desarrollo de un *software* tiene como objetivo la elaboración de un producto que reúna los requisitos del cliente y/o usuario a través de un conjunto de actividades coherentes que son llevadas a cabo por los ingenieros de *software*. Existen cuatro (4) actividades fundamentales de procesos, las cuales son la especificación del *software*, desarrollo del *software*, validación del *software* y evolución del *software*.

En conclusión, el desarrollo de *software* consiste en la realización de un conjunto de actividades, con la finalidad de obtener un producto de *software*.

### **2.2.2. Aplicaciones Web**

Según Conallen (1999) una aplicación Web es un sistema Web conformado por un servidor Web, un protocolo HTTP y un navegador en el cual el usuario, a través de navegación y entrada de datos, afecta el estado del negocio o el estado de los datos del sistema, entendiéndose esto como el cambio de la información que se le presenta al usuario mediante las tres (3) operaciones básicas para el manejo de datos: crear (*insert*), modificar (*update*) y borrar (*delete*).

Existen varias características que ubican a las aplicaciones Web en una posición especial dentro de la Ingeniería de *Software*. Algunas de ellas son:

- La importancia del contenido y de la presentación en un sitio Web hacen necesaria una estrecha colaboración dentro de un equipo en el que coexisten programadores, especialistas en contenido y diseñadores gráficos.
- Existen, además de las tradicionales aplicaciones centradas en documentos muchas otras. Por citar sólo algunas, se puede mencionar las aplicaciones interactivas, transaccionales, sociales y colaborativas.
- Los requerimientos de calidad se vuelven más complejos, ya que hay que considerar la calidad del contenido (consistencia, confiabilidad, actualización, relevancia entre otros.) Asimismo, es necesario tener en cuenta la calidad de la estructura de hipertexto que generalmente subyace en un sitio Web.

- El contexto social se vuelve extremadamente complejo, ya que el perfil de usuarios es casi imposible de determinar. Hay también diferencias educativas y culturales que hacen que los parámetros de usabilidad se vuelvan relativos. El aspecto de “retención” de un usuario es especialmente relevante dada la facilidad con la que un usuario puede dejar un sitio Web en el caso de que, por ejemplo, algo no sea de su agrado.
- El contexto técnico es también complejo. La multiplicidad de plataformas desde las cuales se puede acceder una aplicación Web cambia tanto que es difícil para un usuario normal tener un panorama general. Esto hace que el tradicional “mantra” que se suele invocar cuando se habla de requerimientos (hay que concentrarse en “qué” y no en “cómo”) se vuelve impracticable en el caso de aplicaciones Web por la especial interacción que hay entre los requerimientos y la arquitectura.
- Finalmente, la localización geográfica del usuario requiere de las aplicaciones capacidad para adaptarse a diversos contextos locales (por ejemplo, diferencias de hora, idioma, cultura, etc.)

Las aplicaciones web se clasifican en diferentes categorías, dependiendo de su uso, entre los autores podemos citar la clasificación de Ginige & Murugesan (2001), Pressman (2005), Kappel, Pröll, Reich & Retschitzegger (2003) y Mendoza (2004). Esta clasificación o caracterización se puede observar en la tabla 2.1.

Tabla 2.1. *Caracterización de las aplicaciones web.*

<b>Ginige &amp; Murugesan,2001</b>	<b>Pressman, 2005</b>	<b>Kappel, Pröll, Reich &amp; Retschitzegger, 2003</b>	<b>Mendoza, 2004</b>
Informativas Interactivas Transaccionales Sistemas de workflow Ambientes de trabajo Colaborativo Portales Web Comunidades en línea	Informativas De descarga Personalizable Interactivas Entradas de usuario Orientadas a transacciones Orientadas a servicios Portales Con acceso a bases de datos Almacenes de datos	Sitios Web centrados en documentos Interactivas Transaccionales Flujos de trabajos Colaborativo Web social Portales de negocios, de mercados o de comunidades Aplicaciones de Web ubicua Web semántica	Informativas Instruccionales Portales de comercio electrónico (e-commerce), de negocios inteligentes (e-business) Gobiernos electrónicos (e-government) Sistemas de Información web (SIW)

Fuente: Ginige & Murugesan (2001), Pressman (2005), Kappel, Pröll, Reich & Retschitzegger (2003) y Mendoza (2004).

### 2.2.3. Ingeniería de Métodos

Para entender detalladamente este tema, es importante comenzar a definir qué es un método de desarrollo.

Según Harmsen & Saeki (1996), un método de ingeniería de sistemas de información es: "Una colección de procedimientos, técnicas, descripciones de productos y herramientas para el soporte efectivo, eficiente y consistente del proceso de ingeniería de un Sistema de Información".

Un método de desarrollo debe comprender según lo expuesto por Cheesman & Daniels (2001) los siguientes elementos:

- Elementos de modelado, los cuales son módulos conceptuales básicos.
- Una notación o lenguaje de modelado cuyo objetivo es asegurar la coherencia visual de los elementos del modelado.
- Un proceso que describa las etapas a seguir en el desarrollo.

Una de las muchas definiciones del concepto de método es la de Booch (1991): "Un método de ingeniería de sistemas es un proceso riguroso para generar un conjunto de modelos que describe diversos aspectos de un *software* que se está construyendo usando alguna notación definida".

Para efectos de este trabajo, el concepto de Harmsen & Saeki (1996) es el más indicado; ya que describe los pasos o procedimientos que deben realizarse en un orden dado; las técnicas, las herramientas a utilizar y los entregables o productos a generar en cada una de las fases del proceso de desarrollo.

Ahora bien, Según Brikkemper (1995), la Ingeniería de Métodos (IM) es “una disciplina de conceptualización, de construcción y adaptación de métodos, técnicas y herramientas para el desarrollo de sistemas de información”. Del mismo modo que la Ingeniería del *Software* se refiere a todos los aspectos de la producción del *software*, también lo es la ingeniería de métodos tratar con todas las actividades de ingeniería relacionadas con la generación de métodos, técnicas y herramientas.

Según Ralyté (2001) existen dos formas mediante la cual, se pueden definir nuevos métodos; estos son:

1. A través de métodos existentes siendo ésta la nueva tendencia de IM. También se le denomina reutilización de métodos. Una adaptación puede consistir en extender un método con un nuevo enfoque, si el método no es lo suficientemente rico; mejorar la calidad del método mediante nuevas propiedades (gestión del tiempo, por ejemplo); completar un método con un nuevo modelo.
2. “De *scratch*” o “Desde cero” la cual es una actitud clásica en la construcción de métodos. Algunas razones pueden ser: que los ingenieros presenten mejores ideas para mejorar las actividades en el desarrollo de sistemas; y/o los diseñadores apliquen el lenguaje de modelado existente para describir el método, ejemplo de esto son los métodos *ObjectOriented Software Engineering* (OOSE) o Ingeniería de *Software* Orientado a Objetos (Jacobson, 1992) y *ObjectModelingTechnique* (OMT) o Técnica de Modelación Orientada a Objetos (Rumbaugh, 1991).

La ingeniería de métodos se fundamenta en los principios de metamodelado, reutilización, modularidad y flexibilidad. En este contexto, un

método aporta los conceptos para describir el producto a elaborar, las reglas de conducción metodológica para realizar aquellas actividades que conllevan a construir un producto (Brinkkemper, 1995).

Otro principio según Odell (1996) establece que un método debe estar definido en términos de dos elementos: un modelo de producto y un modelo de proceso. El modelo de producto representa el conjunto de conceptos que se pueden utilizar para construir un producto o sistema y el modelo de proceso permite construir el producto o sistema dando directivas y/o prescripciones que guían las actividades y tareas de desarrollo del producto. Montilva & Barrios (2003) adicionan un tercer elemento; el modelo de actores, que se refiere al equipo de personas involucradas en el desarrollo de los componentes.

Los antecedentes y la teoría definida es la base fundamental para el entendimiento de esta investigación, esencialmente los principios de Ingeniería de Métodos, los cuales se aplican para la evaluación y posterior redefinición de la metodología Gil (2013), presentada en los siguientes capítulos.

## **CAPÍTULO 3. METODOLOGÍA “CREACIÓN DE SITIOS WEB”**

En este capítulo se definen los aspectos generales de la metodología “Creación de sitios Web” (Gil, 2013), sus objetivos, los usuarios potenciales, las fases, la estructura y la implementación en proyectos reales para una mayor comprensión de la misma. Luego, se definen los tres modelos según la Ingeniería de Métodos: modelo de procesos, modelo de productos y modelo de actores.

### **3.1. Objetivo**

La metodología tiene como objetivo principal guiar al equipo de desarrollo de una manera estructurada a través de fases o etapas para así obtener un producto de *software* funcional que cumpla con las necesidades del cliente.

Esta metodología está orientada a aplicaciones web, por lo tanto da importancia a aspectos diferentes a desarrollos tradicionales tales como: identificar las necesidades y expectativas de los diferentes usuarios, el diseño de la interfaz, la presentación del contenido, la estructura de navegación, las pruebas en diferentes navegadores web, entre otros.

### **3.2. Usuarios Potenciales**

Fue creada principalmente para ser utilizada en el ámbito académico en el contexto venezolano. Ajustado específicamente a estudiantes del último



semestre de la carrera Técnico Superior Universitario en Informática de la Universidad Politécnica Territorial de Valencia (UPTVal).

Por lo tanto, el tamaño del equipo es pequeño, menos de tres (3) participantes, el tiempo de desarrollo es de corta duración (menos de un año) y las aplicaciones a desarrollar son de baja complejidad sin muchas restricciones de diseño o seguridad en la web.

### **3.3. Fases**

La metodología comprende siete (7) fases, las cuales son: análisis, planificación, contenido, diseño, programación, testeo y su fase final mercadeo y publicidad. A continuación, se describen las mismas:

#### **3.3.1. Fase de Análisis**

Se definen los objetivos en relación a las expectativas de la organización y los usuarios. Se establecen algunas interrogantes que permitirán aclarar y definir de manera básica las perspectivas de la aplicación web.

#### **3.3.2. Fase de Planificación**

La planificación del desarrollo de un sitio web debe abarcar aspectos como beneficios a adquirir con el funcionamiento del mismo. Antes de establecer el desarrollo de un sitio web se debe estimar el tiempo de retorno de la inversión, definir cuáles son los requerimientos técnicos, seleccionar la estructura del equipo de desarrollo, planear un sitio bien organizado y realizar estudios de mercadeo comparativo.

Se sugiere una lista de miembros que pueden ser parte del equipo: jefe del proyecto, productor, editor, escritor, diseñador de base de datos, diseñador de *scripts*, diseñador gráfico, técnico de *hardware*, técnico de prueba. No indica el formato ni modelo para definir el equipo de desarrollo.

### **3.3.3. Fase de Contenido**

Es conveniente presentar un contenido acorde a los objetivos planteados para desarrollar el sitio, y que la información a plasmar cubra las necesidades e intereses del público a quien va dirigido.

En esta fase, no se señala ningún formato ni textual, ni en diagramas para indicar el contenido de la aplicación.

### **3.3.4. Fase de Diseño**

El diseño web supone un trabajo extenso y detallado, puesto que abarca no sólo la interacción de múltiples elementos, como tecnología multimedia (audio, sonido, imágenes, animaciones, entre otros); sino, que abarca también su integración con una estructura lógica basada en el propósito del sitio. Representando una labor que va más allá de la simple presentación visual de la información.

Al momento de diseñar sitios web, es necesario tener en cuenta aspectos como la usabilidad y accesibilidad. La usabilidad es la facilidad de uso de una interfaz, en relación al tiempo, esfuerzo y capacidad del usuario; y la accesibilidad significa que mayor número de usuarios con limitaciones derivadas del contexto de uso y del dispositivo de acceso empleado (*hardware/software*) puedan interactuar en un entorno web.

En esta fase no se especifica ninguna prueba de usabilidad y en relación a las pruebas de accesibilidad, se toma en cuenta sólo las limitaciones de hardware y *software*:

- Pruebas de la aplicación en diferentes computadores y navegadores web.
- Pruebas del diseño de las páginas web, desactivando algunas funcionalidades de los navegadores web.

### **3.3.5. Fase de Programación**

En esta fase se lleva a cabo la creación de la base de datos, el diseño de la interfaz y la codificación de los programas que se ejecutarán del lado del cliente y del servidor.

### **3.3.6. Fase de Testeo**

Se realizan las siguientes pruebas: comprobación en navegadores, la detección de vínculos rotos y la comprobación del tiempo de descarga.

### **3.3.7. Fase de Mercadeo y Publicidad**

En esta fase se debería tomar en cuenta los derechos internacionales del autor; luego se define el dominio del sitio, se transfieren los archivos al servidor remoto por *FTP* y, finalmente, se realiza la publicación de la aplicación web en algunos buscadores.

En la figura 3.1 se aprecian los componentes con sus aspectos contenidos en cada fase; se les denomina aspectos porque no representan una acción para definirlos como actividad.

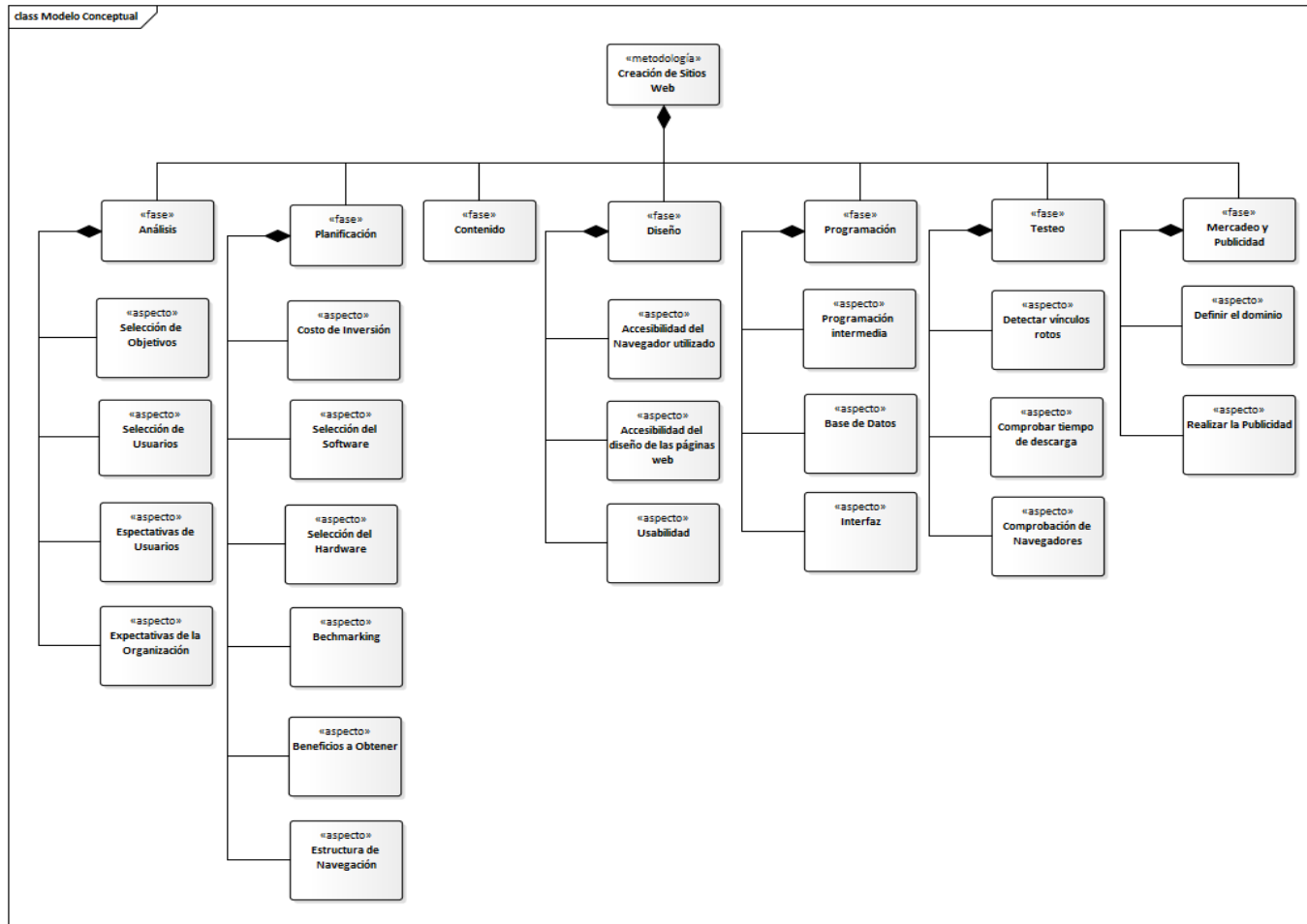


Figura 3.1. Componentes de la metodología “Creación de sitios Web”.

Este modelo se construyó con los términos usados en la metodología. Ejemplo en la Fase de Planificación se definen los aspectos: Costo de Inversión, Estructura de Navegación, Selección del *Software*, Equipo de Desarrollo y estudios de mercadeo comparativo (*bechmarking*).

### **3.4. Estructura**

La metodología se estructura por un conjunto de fases implementadas de una manera lineal sistemática, la cual no permite avanzar a la siguiente fase sin completar la anterior.

### **3.5. Aplicación de la metodología en proyectos reales**

La metodología se aplica según el nivel de entendimiento de la misma por parte del grupo de desarrollo; y según el conocimiento del tutor para guiar e indicar los pasos y los productos a generaren cada fase.

### **3.6. Modelos según la Ingeniería de Método**

A continuación, se detalla la representación de la metodología según los tres modelos: Modelo de Procesos, Modelo de Productos y Modelo de Actores.

#### **3.6.1. Modelo de Procesos**

Este modelo identifica los procesos de Gestión de Proyectos y los procesos Técnicos que el equipo de trabajo debería seguir durante la ejecución de un proyecto de desarrollo.

Para poder generar el modelo de proceso es necesario evaluar la descripción de los aspectos en cada fase. La mayoría no están nombrados adecuadamente como actividad. Por ejemplo, el aspecto “interfaz” representa un vacío de identidad ya que no es identificado con un verbo; el mismo deberá ser redefinido como: diseñar la estructura de la interfaz. Todos estos cambios se visualizan en el modelo de la figura 3.2. En relación a la clasificación de los procesos gerenciales, de soporte y técnicos se visualiza la no correspondencia de los aspectos en cada fase.

Al realizar dichos cambios la estructura del modelo estará representada por procesos y estos a su vez por actividades. La metodología de Gil (2013) no especifica procedimientos para ejecutar la actividad, tampoco describe formatos o productos que genera cada actividad.

El modelo de procesos de la metodología está formado por ocho (8) procesos, los cuales se visualizan en la cadena valor. Los procesos técnicos constituyen los procesos claves para el desarrollo de la aplicación; éstos se ubican en la parte superior de la cadena de valor; y los procesos de gestión se llevan a cabo en la parte inferior (ver figura 3.3).

En los procesos Análisis y Planificación se definen algunas actividades de gestión de proyectos, tales como:

- Seleccionar los objetivos del producto
- Establecer las expectativas tanto de los clientes como las de la organización.

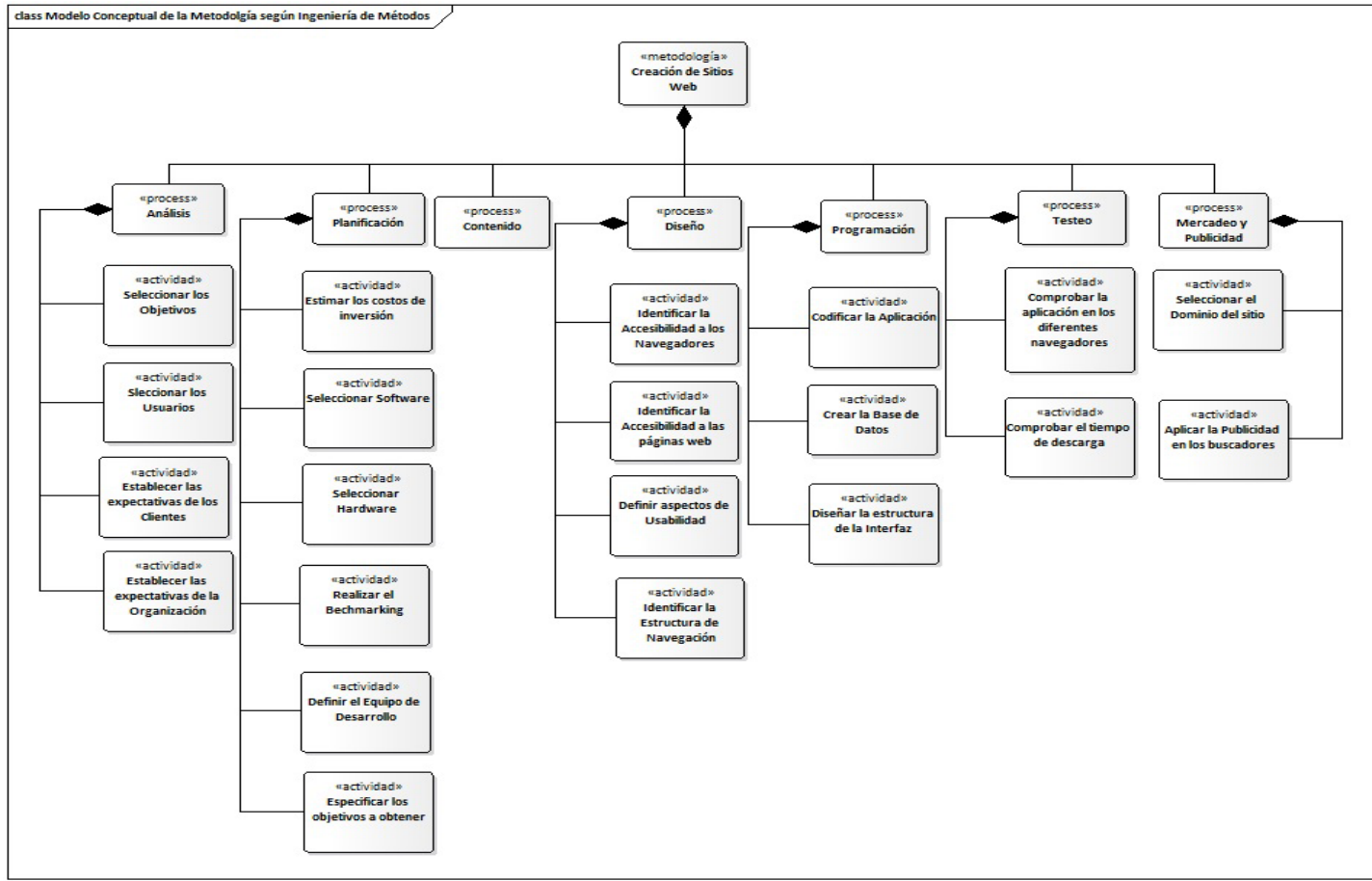


Figura 3.2. Componentes de la metodología Gil (2013) según Ingeniería de Métodos.

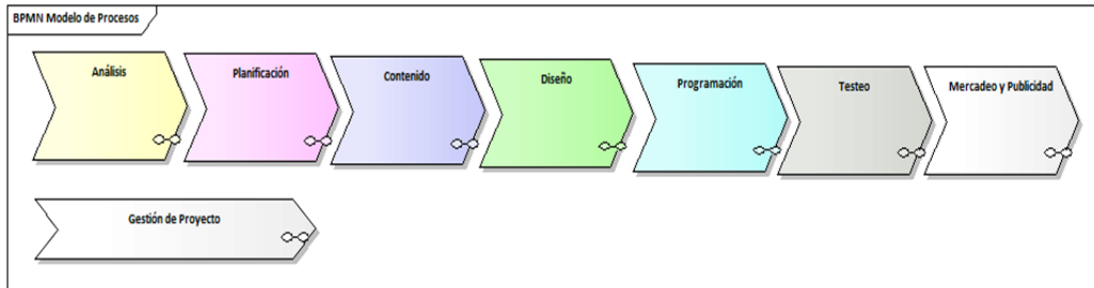


Figura 3.3. Modelo de Procesos de la metodología Gil (2013).

- Estimar costos.
- Especificar los objetivos a obtener con el desarrollo.

La metodología no indica un proceso de seguimiento de los cambios que se ejecute en las fases subsiguientes; solo hace referencia de las actividades mencionadas anteriormente en las fases iniciales de análisis y planificación.

### 3.6.2. Modelo de Productos

Para un método es importante el modelo de productos ya que éste define los tipos de productos que se deben generar durante el desarrollo de la aplicación de *software*. Este modelo identifica los productos intermedios y los productos finales que el equipo de desarrollo debería producir durante la ejecución de un proyecto de desarrollo. Dentro de los productos intermedios se definen los productos técnicos, de gestión y de soporte.

Los productos intermedios son los usados como insumo en la producción de otros productos terminados, ejemplo: el modelo de datos, diseño de interfaces, prototipos; y, los productos finales son los que se entregan al usuario una vez que finaliza el proceso de desarrollo de la



aplicación. El modelo se representa en la tabla 3.1, donde se aprecian los productos generados en las actividades de cada fase.

Tabla 3.1. *Modelo de Productos intermedios de la metodología Gil (2013).*

<b>Fase</b>	<b>Actividad</b>	<b>Producto(s)</b>
Análisis	Seleccionar objetivos	Texto informal
	Seleccionar usuarios	
	Establecer las expectativas del usuario	Texto informal
	Establecer las expectativas de la organización	Texto informal
Planificación	Seleccionar el <i>software</i>	Requisitos de <i>Software</i>
	Seleccionar el <i>hardware</i>	Requisitos de <i>Hardware</i>
	Definir el equipo de desarrollo	
	Realizar el <i>bechmarking</i>	Lista de páginas inherentes al dominio de la aplicación y de los elementos de diseño seleccionados de dichas páginas para el desarrollo de la interfaz de la aplicación.
	Identificar la estructura de navegación	Gráfico de la Estructura de Navegación.
	Estimar los costos de inversión	Estudio de Viabilidad
	Especificar los beneficios a obtener	Texto informal
Contenido		No se especifica
Diseño	Definir los aspectos de usabilidad	Texto informal
	Identificar la accesibilidad a los navegadores	Gráficos de la puesta en marcha de la aplicación en los diferentes navegadores.
	Identificar la accesibilidad del diseño de las páginas web	Gráficos de la puesta en marcha con la opción de <i>javascript</i> desactivada.
Programación	Crear la Base de Datos	Modelo Entidad Relación, Modelo Físico y Lógico de la Base de Datos.
	Diseñar la estructura de la Interfaz	Boceto de la interfaz con las secciones de la aplicación. Interfaz de la aplicación.
	Codificar la aplicación	Aplicación
Testeo	Comprobar la aplicación en diferentes navegadores	Gráficos de la aplicación en los diferentes navegadores.
	Comprobar vínculos rotos	No se especifica
	Comprobar el tiempo de descarga	No se especifica
Mercadeo y Publicidad	Incluir los derechos de autor	Licencia
	Aplicar la publicidad en los buscadores	No se especifica
	Seleccionar el dominio y servidor de alojamiento	Dominio y servidor de alojamiento definidos

Los productos intermedios se clasifican en los productos técnicos y de gestión, no específica procesos de soporte (ver tabla 3.2).

Tabla 3.2. *Productos intermedios de la metodología Gil (2013).*

<b>Productos Técnicos</b>	<b>Productos de Gestión</b>
Requerimientos de <i>Software</i>	Objetivos
Requerimientos de <i>Hardware</i>	Expectativas de la Organización
Listas de páginas inherentes al dominio	Expectativas del usuario
Gráfico de la estructura de navegación	Equipo de Desarrollo
Aspectos de usabilidad	Estudio de Viabilidad
Gráficos de la accesibilidad	Beneficios
Base de Datos	
Boceto de la interfaz	
Gráficas de las pruebas en los navegadores	

El producto final es la aplicación, la cual es entregada al usuario al culminar el proceso de desarrollo sin ningún formato definido, ni manual de usuario y/o sistema.

### **3.6.3. Modelo de Actores**

En la figura 3.4 se ilustra el Modelo de Actores conformado por un equipo de desarrollo donde se identifican diferentes roles, los cuales se pueden ajustar dependiendo del alcance del proyecto y el tamaño del grupo de desarrollo.

A pesar de la definición del modelo de actores, la metodología no indica ningún producto donde se especifique la estructura del equipo de desarrollo y sus responsabilidades.

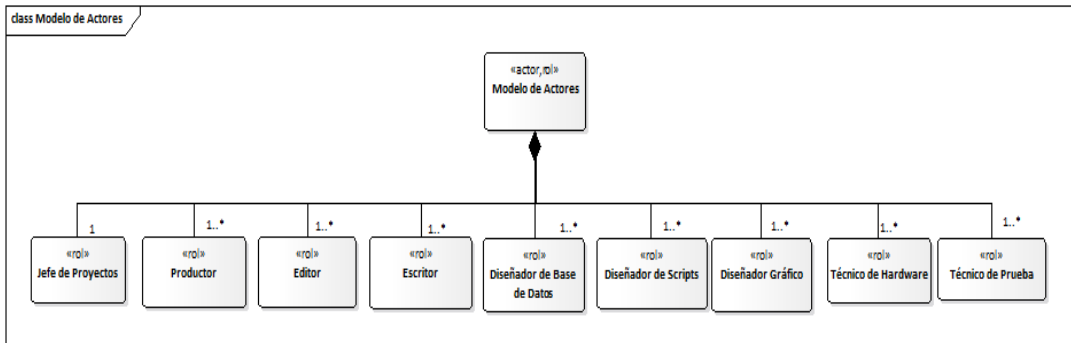


Figura 3.4. Modelo de Actores de la metodología Gil (2013).

Una vez descrita las fases de la metodología y su representación según la Ingeniería de Métodos, ésta se evaluará en el siguiente capítulo utilizando el marco de referencia Mendoza & Barrios (2004).

## **CAPÍTULO 4. EVALUACIÓN DE LA METODOLOGÍA “CREACIÓN DE SITIOS WEB”**

El propósito de este capítulo es evaluar la metodología Gil (2013), presentada en el capítulo anterior, usando el marco de referencia de Mendoza & Barrios (2004). Para esto en la primera sección, se describe el marco de referencia, sus principios y los aspectos fundamentales para la evaluación de métodos; en la segunda sección se evalúa la metodología utilizando el marco definido y, en la última sección, se realiza un análisis de las Debilidades y Fortalezas (D-F) según los resultados de la evaluación anterior.

### **4.1. Definición del Marco de Referencia propuesto por Mendoza & Barrios (2004)**

El marco de referencia está basado en los trabajos de Jarke (1992) y Barrios & Nurcan (2002) y usa los principios de ingeniería de métodos. El objetivo principal del marco de referencia es la evaluación de métodos. Esta evaluación considera cuatro (4) perspectivas relevantes, los cuales se le denominan vistas: Vista de dominio: ¿Para qué sirve y para qué es concebido?; Vista de uso: objetivos del que lo usa; Vista de producto; resultado de aplicar el método y, la Vista de proceso: desarrollo del producto.

En cada vista se consideran facetas, las cuales describen las diferentes características asociadas a dicha vista. Y cada faceta está compuesta por un conjunto de atributos con sus correspondientes valores.

#### 4.1.1. La vista de dominio

Está relacionada con el dominio de aplicación del método a evaluar, la cual está descrita en una faceta: el Alcance; la misma contempla todos los atributos relacionados con diferentes áreas de aplicación y los aspectos tanto técnicos, gerenciales y aquellos que están relacionados específicamente al dominio de aplicación. La estructura de esta vista se muestra en la tabla 4.1.

Tabla 4.1. *Vista de dominio.*

Faceta	Atributos	Valores
Alcance	Área de aplicación	Hipermedia
		Educativo
		<i>e-commerce</i>
		<i>e-business</i>
		Portales
		Sistema de Información Web
	Aspectos	<i>e-government</i>
		Técnicos
		Gerenciales
		Inherentes al área de aplicación

#### 4.1.2. La vista de uso

Esta vista está relacionada con las facetas de aplicabilidad, uso y participación del usuario en el desarrollo de la aplicación web. La faceta de aplicabilidad evalúa las fases que cubren el ciclo de vida y la orientación que se le dará a la aplicación web. El uso describe las características relacionadas con el uso del método: visibilidad, estandarización y eficiencia de uso y, la faceta participación del usuario, especifica los tipos de usuario

esperado y su grado de participación en el desarrollo de aplicaciones web. En la tabla 4.2 se muestra la estructura de esta vista.

Tabla 4.2. *Vista de uso.*

Faceta	Atributos	Valores
Aplicabilidad	Ciclo de vida	Análisis
		Diseño
		Desarrollo
		Evaluación
		Mantenimiento
	Orientación	Académico
		Comercial
		Publicitario
Uso	Visibilidad	Siempre muestra qué hacer
		Siempre muestra cómo hacerlo
	Estandarización	Uso de técnicas estándares
		Uso de notaciones estándares
	Eficiencia de uso	Adaptable y expandible
		Simple y fácil de usar
Participación del usuario	Tipo de usuario	Desarrolladores de <i>software</i>
		Diseñadores de páginas
		Publicistas
		Personal directivo de grandes organizaciones
	Grado de participación	Total
		Parcial
		Ninguno

#### 4.1.3. La vista de producto

Esta vista evalúa el modelo de producto usado por el método; y está definida por dos facetas: la descripción del producto y la descripción conceptual.

La faceta descripción del producto contiene los atributos modelo del producto, su notación, orientación y perspectiva; y la faceta de descripción conceptual describe los conceptos que son usados por el modelo de producto. La estructura de esta vista se detalla en la tabla 4.3.

Tabla 4.3. *Vista del producto.*

Faceta	Atributos	Valores
Descripción del Producto	Modelo del producto	Explicito
		Implícito
		No existe
	Notación del modelo	Formal (Fórmulas matemática)
		Semiformal (Gráficos, imágenes)
		Informal (Textual)
	Orientación del modelo	Orientado a objetos
		Orientado a procesos
		Basado en ejemplos
		Indefinido
	Perspectiva del producto	Perspectiva funcional
		Perspectiva tecnológica
Perspectiva estructural		
Descripción Conceptual	Conceptos usados por el modelo	Conceptos Técnicos
		Conceptos estéticos

#### 4.1.4. La vista de procesos

Esta vista contiene un conjunto de facetas y atributos que describen el proceso de desarrollo de la aplicación que se desea producir; allí, se evalúa el modelo de proceso usado por el método, a través de cinco facetas: modelo de proceso de desarrollo, característica del proceso, gerencia de proyecto, cobertura del ciclo de desarrollo y, del proceso de post-desarrollo. En la tabla 4.4, se muestra la descripción de la vista.

Tabla 4.4. Vista del proceso.

Faceta	Atributos	Valores
Modelo de proceso de desarrollo	Orientación	Orientada al producto
		Orientada a la actividad
		Orientada a la decisión
	Enfoque	<i>Ad-hoc</i>
		Predictivo (cascada) Evolutivo (adaptativo)
Características del proceso	Claridad	Estructura del proceso bien definido
	Estructura del proceso (qué hacer)	Fases/pasos/areas
	Modo de ejecución del proceso	Iterativo
		Secuencial estricto
		Incremental
	Visibilidad	Definición clara del flujo de trabajo
		Progreso visible externamente
	Dependencia tecnológica	Independiente de las herramientas
	Adaptabilidad	Modularidad
		Procesos modificables
Guía	Explícita	
	Completa	
	Implícita	
Gerencia de proyecto	Planificación y control	Definición del plan del proyecto
		Estimación de recursos
		Monitoreo y evaluación del progreso
	Tamaño del equipo	Individual
		Equipos pequeños (<= 5 personas)
		Equipos Grandes (> 5 personas)
	Aseguramiento de calidad	Calidad en la planificación y control
		Calidad del equipo
		Revisión de técnicas de validación y verificación
	Gestión de la configuración	Planificación y ejecución de la gestión de configuración del <i>software</i>
Documentación	Documentación de plantillas	
Entrenamiento	Entrenamiento equipo	
	Entrenamiento de usuario	
Cobertura del ciclo de desarrollo	Análisis	Representación del conocimiento del dominio
		Análisis de objetos del dominio.
	Definición de requerimientos	Definición de requerimientos informal
		Técnicas de especificación
	Diseño	Diseño de arquitecturas
		Diseño de interfaces
Implementación	Producción de multimedia	
	Creación del sitio web	
Proceso de post-desarrollo	Instalación	Implantación de la aplicación Web.
	Mantenimiento	Soporte de mantenimiento.



## 4.2. Aplicación del Marco de Referencia para la evaluación de la Metodología “Creación de Sitios Web”

Para la evaluación de la metodología “Creación de Sitios Web” (Gil, 2013) la autora utiliza el marco de referencia propuesto por Mendoza y Barrios (2004); descrito en la sección anterior, con sus cuatro (4) perspectivas fundamentales: la vista del dominio, la vista de uso, la vista del producto y la vista del proceso. Con el objeto de visualizar mejor los resultados de la evaluación, se añade una columna en cada vista para indicar con una tilde que ese valor de ese atributo, específicamente, si es cumplido en la metodología evaluada.

### 4.2.1. La vista de dominio

El resultado de la evaluación se muestra en la tabla 4.5.

Tabla 4.5. *Evaluación de la vista de dominio.*

Faceta	Atributos	Valores	Metodología Gil(2013)
Alcance	Área de aplicación	Hipermedia	✓
		Educativo	✓
		<i>e-commerce</i>	✓
		<i>e-business</i>	✓
		Portales	✓
		Sistema de Información Web	✓
		<i>e-government</i>	✓
	Aspectos	Técnicos	✓
		Gerenciales	
		Inherentes al área de aplicación	

Al analizar el resultado de la evaluación de la vista de dominio, en el atributo área de aplicación, la metodología puede ser utilizada en cualquiera de los tipos de aplicación web definidos en la vista.

En relación al atributo Aspectos, la metodología Gil (2013) solo considera los técnicos, obviando así, los gerenciales lo que dificulta el desarrollo del *software* de una manera organizada y sistemática.

#### 4.2.2. La vista de uso

El resultado de la evaluación se muestra en la tabla 4.6.

Tabla 4.6. *Evaluación de la vista de uso.*

Faceta	Atributos	Valores	Metodología Gil(2013)
Aplicabilidad	Ciclo de vida	Análisis	
		Diseño	
		Desarrollo	
		Evaluación	
		Mantenimiento	
	Orientación	Académico	✓
		Comercial	✓
		Publicitario	✓
Servicios		✓	
Uso	Visibilidad	Siempre muestra qué hacer	
		Siempre muestra cómo hacerlo	
	Estandarización	Uso de técnicas estándares	
		Uso de notaciones estándares	
	Eficiencia de uso	Adaptable y expandible	
		Simple y fácil de usar	
		Bien documentado	
Participación del usuario	Tipo de usuario	Desarrolladores de <i>software</i>	
		Diseñadores de páginas	✓
		Publicistas	
		Personal directivo de grandes organizaciones	
	Grado de participación	Total	
		Ninguno	✓

Al analizar el resultado de la evaluación de la vista de uso, en la faceta Aplicabilidad, la metodología no cubre el ciclo de vida sino el ciclo del proceso de desarrollo (análisis, diseño, desarrollo y evaluación). En relación al atributo orientación no posee una definida ya que no está diseñada para ser utilizada en el desarrollo de alguna aplicación específica.

En la faceta de uso, la metodología no está bien documentada, muestra de una manera ligera y superficial las actividades a desarrollar en cada fase, no incluye técnicas ni estándares a utilizar.

Y en relación con la faceta participación del usuario, se sugiere un equipo de desarrollo con una lista de miembros y sus funciones respectivas; pero no especifica ningún formato para definir la estructura seleccionada; Tampoco integra al usuario con las actividades de evaluación en las fases de diseño y pruebas, sólo en la fase de análisis y planificación.

#### 4.2.3. La vista de producto

El resultado de la evaluación se detalla en la tabla 4.7.

Tabla 4.7. Evaluación de la vista del producto.

Faceta	Atributos	Valores	Metodología Gil (2013)
Descripción del producto	Modelo del producto	Explicito	
		Implícito	
		No existe	✓
	Notación del modelo	Formal (Fórmulas matemática)	
		Semiformal (Gráficos, imágenes)	✓
		Informal (Textual)	✓
	Orientación del modelo	Orientado a objetos	
		Orientado a procesos	
		Basado en ejemplos	
		Indefinido	
Perspectiva del producto	Perspectiva funcional		
	Perspectiva tecnológica		
	Perspectiva estructural		
Descripción Conceptual	Conceptos usados por el modelo	Conceptos Técnicos	
		Conceptos estéticos	

A partir de la evaluación, se observa que no existe un modelo de producto, es decir, no define explícitamente los productos a generar en cada fase.

#### 4.2.4. La vista de procesos

El resultado de la evaluación de la metodología se puede observar en las tablas 4.8 y 4.9.

Tabla 4.8. *Evaluación de la vista del proceso (a).*

Faceta	Atributos	Valores	Metodología Gil (2013)
Modelo de proceso de desarrollo	Orientación	Orientada al producto	
		Orientada a la actividad	✓
		Orientada a la decisión	
	Enfoque	<i>Ad-hoc</i>	
		Predictivo (cascada)	✓
		Evolutivo (adaptativo)	
Características del proceso	Claridad	Estructura del proceso bien definido	
	Estructura del proceso (qué hacer)	Fases/pasos/tareas	
	Modo de ejecución del proceso	Iterativo	
		Secuencial estricto	
		Incremental	
	Visibilidad	Definición clara del flujo de trabajo	
		Progreso visible externamente	
	Dependencia tecnológica	Independiente de las herramientas	✓
	Adaptabilidad	Modularidad	
		Procesos modificables	
Guía	Explícita		
	Completa		
	Implícita		
Gerencia de proyecto	Planificación y control	Definición del plan del proyecto	
		Estimación de recursos	✓
		Monitoreo y evaluación del progreso	
	Equipo de trabajo	Estructura del equipo	✓
	Tamaño del equipo	Individual	
		Equipos pequeños (<= 5 personas)	
		Equipos Grandes (> 5 personas)	
		Calidad en la planificación y control	
		Calidad del equipo	
		Revisión de técnicas de validación y verificación	
		Planificación y ejecución de la gestión de configuración del <i>software</i>	
		Documentación de plantillas	
		Entrenamiento equipo	
Entrenamiento de usuario			

Tabla 4.9. *Evaluación de la vista del proceso (b).*

Faceta	Atributos	Valores	Metodología Gil (2013)
Cobertura del ciclo de desarrollo	Análisis	Representación del conocimiento del dominio	
		Análisis de objetos del dominio.	
	Definición de requerimientos	Definición de requerimientos informal	✓
		Técnicas de especificación	
	Diseño	Diseño de arquitecturas	
		Diseño de interfaces	✓
	Implementación	Producción de multimedia	
		Creación del sitio web	
		Construcción de la Base de Datos	✓
		Implementación de las reglas del negocio	
		Programación del lado del cliente	✓
		Programación del lado del servidor	✓
		Implementación de políticas de seguridad (SSL, etc).	
	Pruebas	Comunicación de objetos distribuidos.	
		Prueba de cada uno de los componentes	
		Pruebas de integración	
Prueba de la aplicación web.		✓	
Proceso de post-desarrollo	Validación y verificación de la aplicación web.		
	Instalación	Implantación de la aplicación web	
	Mantenimiento	Soporte de mantenimiento	

Al analizar, en el resultado obtenido para la vista de proceso, se observa que el modelo de desarrollo corresponde con el enfoque tipo cascada. No indica una estructura en pasos/actividades y tareas, las fases están descritas muy superficialmente; no indica ninguna técnica ni herramientas de apoyo en los procesos.

En relación a los procesos de gestión de proyecto, solo considera aspectos de estimación de costos, definición del equipo de desarrollo y beneficios a obtener.

En cuanto a la faceta de cobertura del ciclo de desarrollo, la metodología Gil (2013), solo considera de forma muy ligera y simple los atributos: definición de requerimientos, diseño, implementación y pruebas. El proceso de post-desarrollo no se define, lo que no garantiza que las aplicaciones web estén siempre actualizadas.

### 4.3. Análisis de Bondades y Debilidades de la Metodología “Creación de Sitios Web”

Una vez obtenido los resultados de la evaluación de la metodología Gil (2013) utilizando el marco de referencia de Mendoza & Barrios (2004), se procedió a adaptar, parcialmente, el esquema de la técnica FODA. En este caso, solo se determinan los principales elementos de las Fortalezas (F) y Debilidades (D) resaltadas a partir de la evaluación anterior; dicha matriz D-F se observa en las tablas 4.10 y 4.11.

Tabla 4.10. Análisis Fortalezas y Debilidades de la metodología Gil (2013) (a).

<b>Fortalezas (F)</b>	<b>Debilidades (D)</b>
<p>F1. Hay independencia tecnológica con las herramientas CASE.</p> <p>F2. Es abierta y permite que el estudiante decida “cómo” usarla.</p>	<p>D1. No está definida para un tipo específico de aplicación web.</p> <p>D2. No está bien documentada.</p> <p>D3. No cubre la mayoría de aspectos gerenciales.</p> <p>D4. Algunas actividades de gestión de proyectos se realizan solo en las dos primeras fases.</p> <p>D5. No tiene una orientación específica en el uso.</p> <p>D6. No indica el modelo de desarrollo en el cual se basa el proceso.</p> <p>D7. No hay claridad en el proceso.</p> <p>D8. Las actividades del proceso no están bien definidas, no se describen con un verbo.</p> <p>D9. La estructura del proceso no está definida en pasos/tareas.</p> <p>D10. No detalla el procedimiento para lograr las actividades.</p>

Tabla 4.11. *Análisis Fortalezas y Debilidades de la metodología Gil (2013) (b).*

Fortalezas (F)	Debilidades (D)
	<p>D11. No especifica notación gráfica o textual para representar o documentar la aplicación.</p> <p>D12. No indica los productos a generar en cada fase.</p> <p>D13. No define un plan de desarrollo de <i>software</i>.</p> <p>D14. No hay formato de entrega final de la aplicación.</p> <p>D15. Se describe el equipo de trabajo pero no se especifica ningún formato o diagrama para que definan el mismo.</p> <p>D16. No indica el tipo de usuario (Desarrollador de <i>software</i>, diseñador de páginas, publicista).</p> <p>D17. No especifica el grado de participación del usuario.</p> <p>D18. Nombra un equipo de trabajo con muchos roles, que no corresponde a un pequeño grupo de trabajo de máximo 2 estudiantes por proyectos académicos.</p> <p>D19. La actividad de estimación de costos, recursos y tiempo están definidos en la fase de análisis.</p> <p>D20. Los requisitos están en la fase de planificación en conjunto con las actividades de gestión de proyectos.</p> <p>D21. No indica requerimientos de calidad.</p> <p>D22. La estructura de navegación está en la fase de planificación.</p> <p>D23. El <i>bechmarking</i> está fuera de contexto.</p> <p>D24. El concepto de facilidad de uso (<i>usability</i>) no se desarrolla.</p> <p>D25. Incluye actividades de diseñar en la fase de programación.</p> <p>D26. No indica tipos de pruebas.</p> <p>D27. El nombre de la metodología está mal definida en el término: Creación de sitios Web.</p> <p>D28. Como es tan abierta, no guía ni contribuye a la formación de los estudiantes en los proceso técnicos de desarrollo de <i>software</i>.</p>

Es de notar que, en Gil (2013), hay muchas fallas o vacíos metodológicos lo que conlleva a una redefinición de la metodología basándose en los principios de Ingeniería de Métodos (Barrios, 2002) y, en las mejores prácticas de Ingeniería del *Software* (Montilva, Montilva, Barrios, 2011) las cuales se describen en el siguiente capítulo.

## **CAPÍTULO 5. MÉTODO DE DESARROLLO PARA APLICACIONES WEB BASADO EN INGENIERÍA DE MÉTODOS**

El contenido de este capítulo está centrado en la descripción del método propuesto. Se explican las características y herramientas conceptuales utilizadas para solventar las debilidades y se detallan los tres modelos tomando en cuenta los hallazgos producto de la evaluación de la metodología Gil (2013) descritos en el capítulo anterior.

### **5.1. Características y herramientas conceptuales que solventan las debilidades**

A continuación, se visualizan las características y las herramientas conceptuales utilizadas para solucionar las debilidades (identificadas con la notación Dx, donde x es el número correspondiente en la tabla 4.11 o 4.12):

- A través de un estudio estadístico en la Universidad Politécnica Territorial de Valencia (UPTV)(Anexo A) se establece el dominio del método (D1).
- La documentación del método se realiza al representar los tres modelos: procesos, productos y actores (D2).
- En el Modelo de Procesos: se incluyen algunos aspectos gerenciales (D3), los cuales se identifican en la parte inferior de la cadena valor y se ejecutan en todo el proceso de desarrollo (D4); se indica la orientación de uso: académico (D5); se especifica el modelo de desarrollo (D6); se explican los procesos técnicos, gerenciales y de apoyo (D7); las actividades se describen con verbos; los procesos se definen con



pasos/tareas a través de la estructura del proceso definido (D9); se especifican las técnicas y herramientas en las actividades (D10); se indica una notación (D11); se detallan los productos a generar en cada proceso (D12); en el proceso de gestión de proyecto se define el subproceso plan del proyecto.

- En el Modelo de Productos: se incluye el entregable Plan del Proyecto (PP) (D13), el formato para la entrega de la aplicación (D14) y, el formato para describir el equipo de desarrollo (D15); los productos intermedios y finales se representan a través de diagramas de clases (D12).
- En el Modelo de Actores: se indica el tipo de usuario (D16); el grado de participación del mismo (D17) y se describen los roles de los miembros del equipo de trabajo (D18).
- En el proceso de planificación se define solo la actividad de estimar el tiempo de desarrollo (D19); las otras dos, en relación a los costos y recursos, no se incluirán en esta investigación ya que se definirán en una versión posterior.
- En el proceso de análisis y definición de requisitos (D20) se incluyen los requerimientos (funcionales y no funcionales) (D20) y los de calidad ISO 9126 (D21).
- En el proceso de diseño: se define la estructura de navegación (D22); se elimina el concepto de *bechmarking* y, se incluye la técnica de análisis competitivo para generar la Lista de Sitios Similares (LSS) (D23); se especifican técnicas para la facilidad de uso (*usability*) (D24); se ubica la actividad de realizar el diseño, eliminándola del proceso de programación (D25).
- Los tipos de pruebas se especifican en el producto Documento de Pruebas (DP) y se realizan en el Proceso de Pruebas (D26).

- A través de la revisión documental de la definición de sitios Web se modifica al término a Aplicaciones Web (D27). En este contexto, según Conallen (1999) hace la diferencia entre sitio y aplicación Web desde el punto de vista de la modificación de la lógica del negocio. Según este autor, una aplicación Web es un sistema Web (servidor Web, red, protocolo HTTP, y browser) donde el usuario, a través de navegación y entrada de datos, afecta el estado del negocio. De este modo, un sitio Web es un sistema Web donde la navegación del usuario no modifica lógica de negocio, o un sistema Web donde no hay lógica del negocio.
- La estructura sistemática y ordenada de la representación del método mediante los tres modelos, permite guiar y aportar conocimientos al estudiante (D28).

## 5.2. Caracterización del método

Uno de los principios de la Ingeniería de Métodos es que un método debe tener un propósito específico; por tal motivo se realizó una revisión de los Proyectos Socio Tecnológico (PST) de la Universidad Politécnica Territorial de Valencia(UPTV), clasificándolos según tabla 2.2 (Caracterización de las aplicaciones Web). De los veintitrés (23) proyectos desarrollados sólo nueve (9) son sistemas de escritorio y los catorce restantes (14) son aplicaciones Web (ver tabla 5.1).

Tabla 5.1. *Clasificación de los PST.*

<b>Tipo de Aplicación</b>	<b>Cantidad</b>
Sistemas de Información Web	14
Sistemas de Escritorio	9

Una vez obtenido los resultados, el método se orienta a Sistemas de Información Web (SIW); los cuales se definen como sistemas de información

tradicionales que emplean las tecnologías de Internet con el acceso de los usuarios a través de navegadores Web (Schewe, Thalheim, & Zhao, 2005).

### **5.3. Perspectivas del posicionamiento del método**

El Método se denomina Método Académico para el desarrollo de Sistemas de Información Web (MASIWeb).

MASIWeb se inspira en los métodos *Gray WATCH* (Montilva, Barrios & Rivero, 2008), *Blue WATCH* (Montilva, Montilva & Barrios, 2011) y *White WATCH* (Barrios & Montilva, 2010).

El método describe los procesos técnicos, de gestión y de soporte para equipos pequeños (máximo dos participantes) en el desarrollo de Sistemas de Información Web (SIW) y orientado al ámbito académico.

#### **5.3.1. Características del método**

- Es un proceso balanceado: combina disciplina con agilidad. En este sentido, la disciplina está asociada a la ejecución de un plan de trabajo y a la creación de productos de una manera sistemática y organizada; y la agilidad a los cambios de los requisitos y la facilidad de generar prototipos y versiones previas a la entrega final del producto.
- El enfoque es iterativo y versionado: permite que el cliente evalúe el *software* de forma regular hasta obtener una versión más completa. Cada ciclo de desarrollo, que implica la ejecución de los procesos técnicos, produce una versión de la aplicación.

- Es de propósito específico: está orientado a Sistemas de Información Web (SIW).
- Es extensible: su estructura puede ser mejorada incorporando procesos, productos, técnicas, herramientas dependiendo de las necesidades de desarrollos y de contenidos académicos en el área Web.
- El método emplea prácticas aceptadas y utilizadas en la industria del *software*. A continuación se identifican las prácticas ágiles (A) y disciplinadas (D) que se utilizan en métodos balanceados (Montilva, Montilva, Barrios, 2011):
  - Desarrollo de *software* iterativo (A) y versionado.
  - Gestión de requisitos (D).
  - Modelo visual de la aplicación (D).
  - Código colectivo (A)
  - Desarrollo guiado por pruebas (A).
  - Gestión del riesgo (D).
  - Gestión de requisitos (D).
  - Planificación de iteraciones (A).

### **5.3.2. Componentes del método**

El método se define a través de tres modelos: el Modelo de Procesos, el Modelo de Productos y el Modelo de Actores.

### 5.3.3. Modelo de Procesos

El objetivo de este modelo es describir los procesos técnicos, de gestión y de soporte que el equipo de trabajo debería ejecutar para el desarrollo de la aplicación. Los procesos del método se observan en la cadena de valor, la cual identifica los procesos y los relaciona pero no indica el orden de ejecución de los mismos (ver figura 5.1).

Los procesos técnicos se ubican en la parte superior de la cadena de valor, éstos se encargan de organizar las actividades tecnológicas que caracterizan el desarrollo de una aplicación cualquiera; y, los procesos de apoyo se visualizan en la parte inferior de la cadena de valor, los cuales se ejecutan a lo largo de todos los procesos técnicos del método.

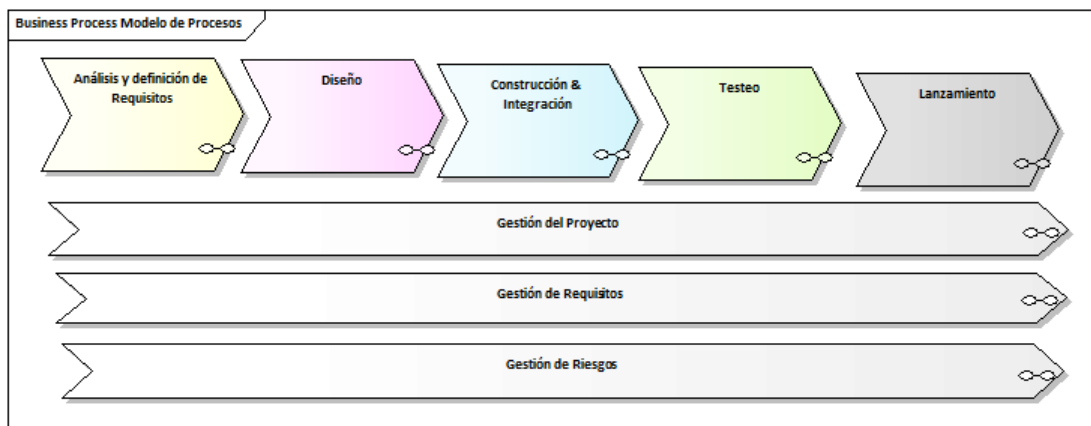


Figura 5.1. Modelo de Procesos del método MASIWeb.

El grupo de procesos técnicos del método MASIWeb está conformado por los siguientes procesos:

- Análisis y definición de requisitos: comprende los subprocesos: modelado del proceso de negocio, descubrimiento, análisis y especificación de requisitos y gestión de requisitos. Se realizan las

actividades de construir el modelo de procesos de negocio, recolectar, clasificar, identificar, priorizar, especificar y gestionar los requisitos funcionales y no-funcionales de la aplicación. Para generar el modelo de proceso de negocio se utiliza la herramienta de modelado visual UML y, las técnicas de elicitación de requisitos para identificar los requisitos y documentarlos en el producto: Especificación de Requisitos de *Software* (ERS).

- Diseño: agrupa las actividades de diseño de la arquitectura de *software*, diseño de la arquitectura de contenido, diseño de componentes, diseño de la interfaz, diseño de navegación y diseño de la base de datos.
- Construcción & Integración: proceso en el cual el programador genera la aplicación Web, cumpliendo con los requisitos definidos en el documento ERS. Se deben ejecutar las pruebas unitarias y de integración. Además, se elabora el Manual de Usuario del Sistema (MUS).
- Pruebas: se ejecutan las actividades de pruebas unitarias, de integración, de aceptación y de sistema en función del plan de pruebas. Este proceso no se ejecuta por el programador, sino por el probador para obtener resultados confiables.
- Lanzamiento: organiza todas las actividades de implantación de la aplicación en la plataforma de producción, las pruebas de instalación y la entrega final del producto.

El grupo de los procesos de gestión apoya a todos los procesos técnicos y, está relacionado con la gestión de proyectos; este grupo tiene un solo proceso:

- Planificación del Proyecto: establece las actividades para definir el informe inicial, el equipo de trabajo, el plan y las iteraciones del proyecto.

Y el grupo de soporte, complementa los procesos técnicos y gerenciales con procesos de:

- Gestión de Riesgos: contempla las actividades de identificar, clasificar y mitigar los probables riesgos que amenazan al plan del proyecto.
- Gestión de Requisitos: organiza las actividades de identificar, controlar y ejecutar los cambios de los requisitos a lo largo del desarrollo de la aplicación.

A continuación, se muestra el diagrama de flujo de trabajo del Modelo de Proceso del método (ver figura 5.2).

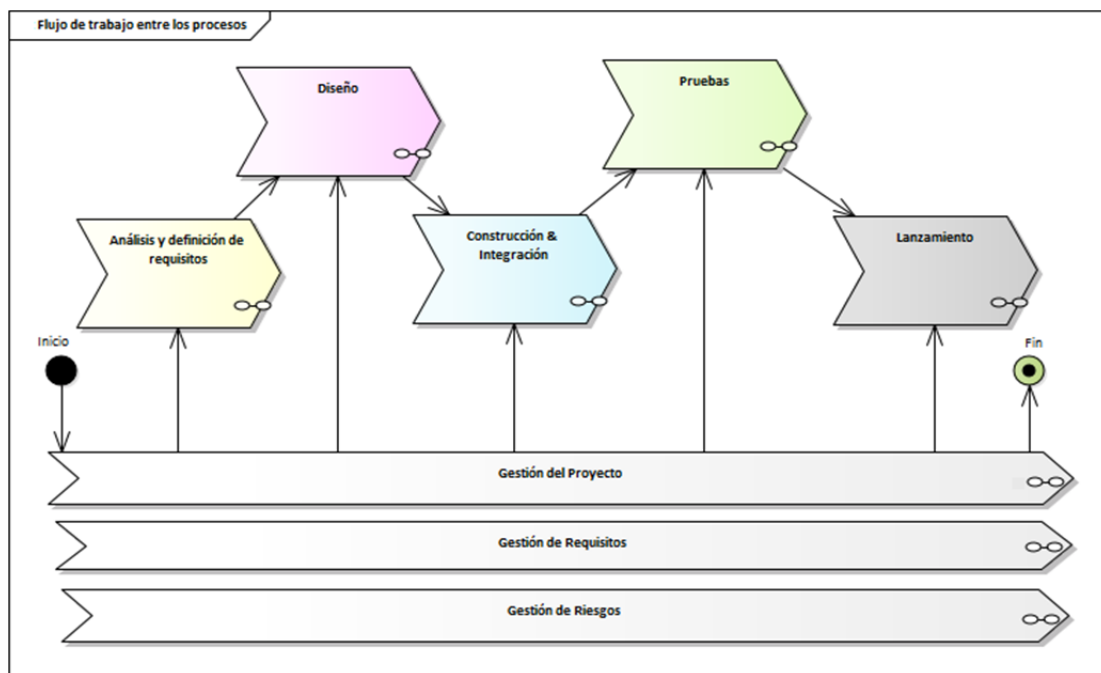


Figura 5.2. Flujo de trabajo del Modelo de Procesos del método MASIWeb.

Para especificar en detalle cada proceso del método y así, guiar al grupo de desarrollo, se tomó como referencia la estructura propuesta por Barrios & Montilva (2010) la cual se basa en pasos (subprocesos), actividades y tareas o acciones (ver figura 5.3).

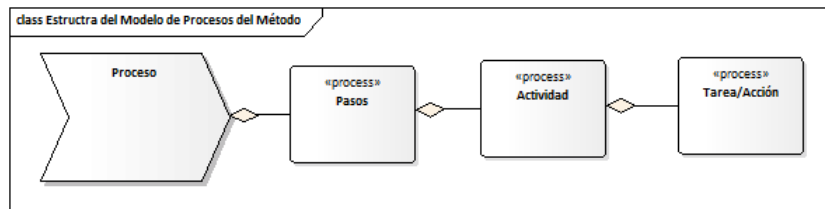


Figura 5.3. Estructura del modelo de procesos del método MASIWeb.

A continuación, se describen los procesos que conforman el método MASIWeb:

### 5.3.3.1. Procesos de Gestión

- Gestión del Proyecto:** comprende los pasos o subprocesos de iniciar el proyecto, organizar el equipo de desarrollo, planificar el proyecto, realizar el seguimiento de los avances del proyecto. La estructura es una adecuación de la propuesta en el Cuerpo de Conocimientos de Gestión de Proyectos del *Project Management Institute* (PMI, 2007) (ver figura 5.4).

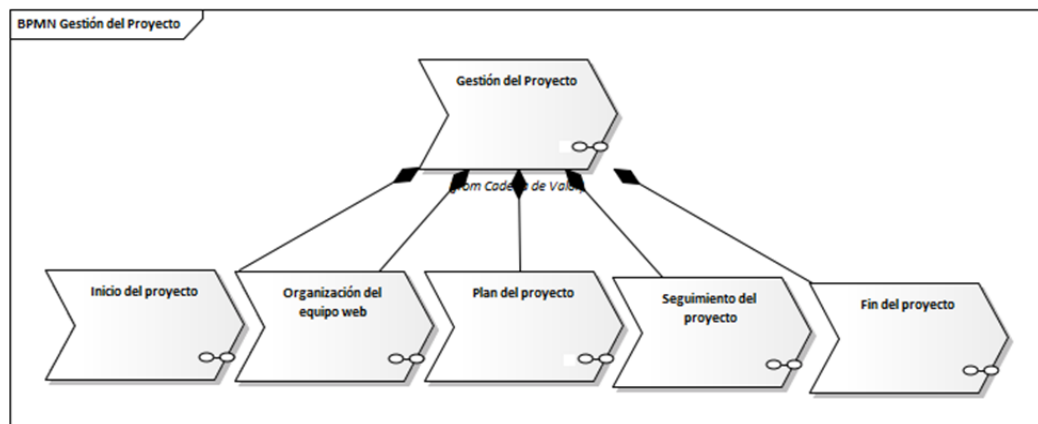


Figura 5.4. Diagrama de jerarquía del proceso de Gestión del Proyecto.



A continuación, se detallan los pasos o subprocesos, las actividades, técnicas y/o notaciones y, los productos generados en este proceso (ver tabla 5.2).

Tabla 5.2. Descripción del proceso de Gestión del Proyecto.

Pasos /Subprocesos	Actividades	Técnicas y/o notaciones	Productos
Inicio del proyecto	<ul style="list-style-type: none"> <li>Definir el problema: los afectados, el impacto y la solución.</li> <li>Identificar los objetivos, el propósito, los beneficios, las limitaciones y los involucrados de la aplicación.</li> <li>Establecer la lista inicial de requisitos del usuario.</li> </ul>	<ul style="list-style-type: none"> <li>Reuniones.</li> <li>Herramientas de apoyo especializadas para documentar.</li> </ul>	<ul style="list-style-type: none"> <li>Documento Visión del Proyecto (DVP) que contiene: <ul style="list-style-type: none"> <li>Tabla del Problema (TP).</li> <li>Tabla de la aplicación (TA).</li> <li>Lista Inicial de Requisitos del Usuario (LIRU).</li> </ul> </li> </ul>
Organización del equipo web	<ul style="list-style-type: none"> <li>Especificar el equipo de desarrollo de acuerdo al Modelo de Actores del método.</li> <li>Establecer las responsabilidades de cada integrante con rotación de roles.</li> </ul>	<ul style="list-style-type: none"> <li>UML.</li> <li>Herramientas de apoyo especializadas para documentar.</li> </ul>	<ul style="list-style-type: none"> <li>Documento Plan del Proyecto (DPP) que contiene: <ul style="list-style-type: none"> <li>Modelo de Actores de la Aplicación (MAA).</li> </ul> </li> </ul>
Plan del proyecto	<ul style="list-style-type: none"> <li>Definir el plan del proyecto indicando el alcance, el cronograma de las actividades, los responsables, los productos generados y porcentaje de avance.</li> <li>Establecer el cronograma del proceso de seguimiento y el de iteraciones.</li> <li>Validar el plan del proyecto con el profesor evaluador y el cliente.</li> </ul>	<ul style="list-style-type: none"> <li>Técnicas y herramientas de apoyo especializadas para planificación de proyectos.</li> <li>Diagrama Gantt.</li> <li>Reuniones.</li> <li>Técnicas de descomposición.</li> </ul>	<ul style="list-style-type: none"> <li>Plan del Proyecto (PP).</li> <li>Cronograma de Iteraciones y Avances del Proyecto (CIAP).</li> <li>Plan del Proyecto Validado (PPV).</li> </ul>
Seguimiento del proyecto	<ul style="list-style-type: none"> <li>Supervisar los avances del proyecto.</li> <li>Documentar los cambios en cada iteración.</li> </ul>	<ul style="list-style-type: none"> <li>Reuniones de Avances.</li> <li>Reuniones de los cambios en las iteraciones</li> </ul>	<ul style="list-style-type: none"> <li>Documento de Seguimiento del Proyecto (DSP) que contiene: <ul style="list-style-type: none"> <li>Minutas de los Avances (MA).</li> <li>Minutas de las iteraciones (MI).</li> </ul> </li> </ul>
	<ul style="list-style-type: none"> <li>Actualizar el plan del proyecto.</li> <li>Actualizar los documentos involucrados en el seguimiento.</li> </ul>	<ul style="list-style-type: none"> <li>Técnicas y herramientas de apoyo especializadas para planificación de proyectos.</li> <li>Diagrama Gantt</li> </ul>	<ul style="list-style-type: none"> <li>Plan del Proyecto cambiado.</li> <li>ERS cambiado.</li> <li>Modelado de Procesos de Negocios (MPN).</li> <li>Documento de Gestión de Riesgos (DGR).</li> <li>Documento de Diseño (DD).</li> <li>Base de Datos.</li> <li>Aplicación</li> </ul>
Fin del proyecto	<ul style="list-style-type: none"> <li>Realizar la ejecución de la aplicación.</li> <li>Entregar la aplicación en el formato establecido.</li> <li>Realizar la entrega del informe final junto con la documentación de los productos intermedios y finales.</li> </ul>	<ul style="list-style-type: none"> <li>Reuniones.</li> </ul>	<ul style="list-style-type: none"> <li>Productos Entregables intermedios y Finales (PEI y PEF) en el Formato de Entrega del Software (FES)</li> <li>Informe Final.</li> </ul>

Definiciones de las técnicas:

- Técnica de descomposición: divide y subdivide el trabajo en partes más pequeñas y más fáciles de manejar (PMI, 2017).
- Diagrama de Gantt: es una representación gráfica de barras con información al cronograma donde, las actividades se enumeran en el eje vertical, las fechas en el eje horizontal y, el período de tiempo, se representa con barras horizontales indicando fecha de inicio y finalización (PMI, 2017).

En la tabla 5.3 se describen las entradas y salidas de cada subproceso.

Tabla 5.3. Entradas y salidas del proceso de Gestión del Proyecto.

Entrada	Subproceso	Salida
<ul style="list-style-type: none"> <li>• Necesidades del cliente.</li> </ul>	Inicio del proyecto	<ul style="list-style-type: none"> <li>• Documento Visión del Proyecto (DVP).</li> </ul>
<ul style="list-style-type: none"> <li>• Modelo de Actores del método MASIWeb.</li> </ul>	Organización del equipo web	<ul style="list-style-type: none"> <li>• Modelo de Actores del Proyecto (MAP).</li> </ul>
<ul style="list-style-type: none"> <li>• Documento Visión del Proyecto (DVP).</li> <li>• Modelo de Productos de MASIWeb.</li> <li>• Modelo de Procesos de MASIWeb</li> </ul>	Plan del proyecto	<ul style="list-style-type: none"> <li>• Documento Plan del Proyecto (DPP).</li> </ul>
<ul style="list-style-type: none"> <li>• Documento Plan del Proyecto (DPP).</li> <li>• Documento de Especificación de Requisitos del <i>Software</i> (ERS).</li> <li>• Modelado de Procesos de Negocios (MPN).</li> <li>• Documento de Gestión de Riesgos (DGR).</li> <li>• Documento de Diseño (DD).</li> <li>• Base de Datos.</li> <li>• Aplicación.</li> </ul>	Seguimiento del proyecto	<ul style="list-style-type: none"> <li>• Documento de Seguimiento del Proyecto (DSP).</li> <li>• Plan del Proyecto cambiado.</li> <li>• ERS cambiado.</li> <li>• DD cambiado.</li> <li>• DGR cambiado.</li> <li>• Base de Datos revisada.</li> <li>• Aplicación revisada</li> </ul>
<ul style="list-style-type: none"> <li>• Productos Intermedios y Finales</li> </ul>	Cierre del proyecto	<ul style="list-style-type: none"> <li>• Informe Final</li> </ul>

### 5.3.3.2. Procesos de Soporte

•**Gestión de Riesgos:** comprende los pasos o subprocessos de identificación de riesgos, planificación de respuesta y monitoreo de los riesgos (ver figura 5.5).

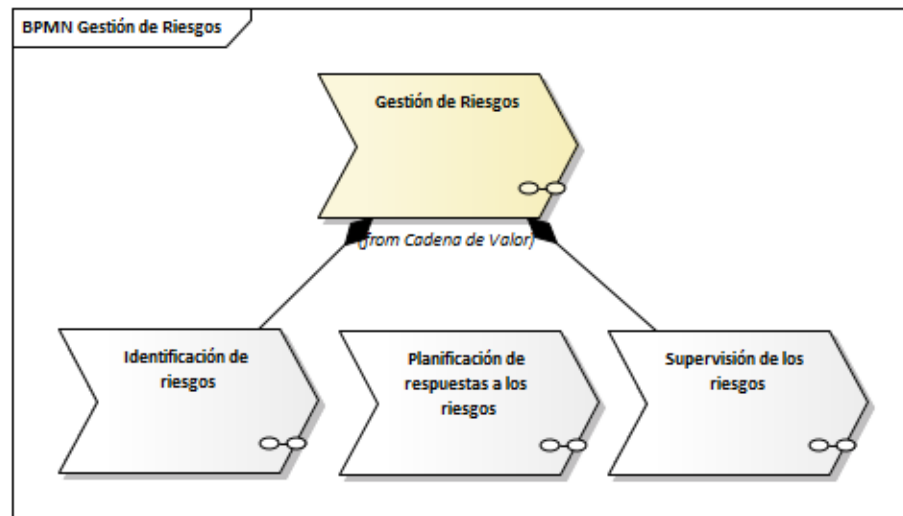


Figura 5.5. Diagrama de jerarquía del proceso de Gestión de Riesgos.

A continuación, se detallan los pasos o subprocessos, las actividades, técnicas y/o notaciones y los productos generados en este proceso (ver tabla 5.4).

Tabla 5.4. Descripción del proceso de soporte: Gestión de Riesgos.

Pasos /Subprocesos	Actividades	Técnicas y/o notaciones	Productos del Documento Gestión de Riesgos (DGR)
1. Identificación de riesgos	<ul style="list-style-type: none"> <li>• Identificar los riesgos.</li> <li>• Clasificar los riesgos.</li> <li>• Indicar el grado de magnitud o <i>ranking</i>(1..10) de los riesgos.</li> <li>• Describir el impacto ocasionado ((alto, bajo)/ (catastrófico, crítico, marginal, despreciable)) de cada riesgo.</li> </ul>	<ul style="list-style-type: none"> <li>• Reuniones.</li> <li>• Herramientas de apoyo especializadas para documentar.</li> <li>• Revisión documental</li> <li>• Diagrama de Ishikawa o de espina de pescado.</li> <li>• Tormenta de ideas.</li> <li>• Matriz FODA.</li> <li>• Herramientas de apoyo especializadas para documentar.</li> </ul>	<ul style="list-style-type: none"> <li>• Lista de Riesgos (LR).</li> </ul>
2. Planificación de respuestas a los riesgos	<ul style="list-style-type: none"> <li>• Identificar los riesgos a gestionar.</li> <li>• Determinar las estrategias para mitigar los riesgos identificados.</li> <li>• Determinar el plan de contingencia para aminorar el impacto de los riesgos identificados.</li> </ul>	<ul style="list-style-type: none"> <li>• Reuniones.</li> </ul>	<ul style="list-style-type: none"> <li>• Plan de Mitigación de Riesgos (PMR).</li> <li>• Plan de Contingencia de Riesgos (PCR).</li> </ul>
3. Supervisión de los Riesgos	<ul style="list-style-type: none"> <li>• Monitorear los planes de contingencia y de mitigación de los riesgos.</li> <li>• Realizar el seguimiento a los riesgos identificados y analizar nuevos riesgos.</li> </ul>	<ul style="list-style-type: none"> <li>• Reuniones.</li> </ul>	<ul style="list-style-type: none"> <li>• Documento Supervisión de Riesgos (DSR) que contiene:               <ul style="list-style-type: none"> <li>○ Plan de Mitigación de Riesgos Actualizado (PMRA).</li> <li>○ Plan de Contingencia de Riesgos Actualizado (PCRA).</li> </ul> </li> </ul>

Definiciones de las técnicas:

- Diagrama de Ishikawa o de espina de pescado: Este tipo de diagrama desglosa las causas del enunciado del problema identificado en ramas separadas, que ayudan a identificar la causa principal o raíz del problema (PMI, 2017).
- Tormenta de ideas: Esta técnica se utiliza para identificar una lista de ideas en un corto período de tiempo. Se lleva a cabo en un entorno de

grupo y es liderada por un facilitador. La tormenta de ideas comprende dos partes: generación de ideas y análisis (PMI, 2017).

- **Matriz FODA:** La técnica comienza con la identificación de las fortalezas y debilidades de la organización, centrándose ya sea en el proyecto, en la organización o en el negocio en general, luego identifica cualquier oportunidad con origen en las fortalezas y cualquier amenaza que resulte de las debilidades (PMI, 2017).

En la tabla 5.5 se describen las entradas y salidas de cada subproceso.

Tabla 5.5. *Entradas y salidas del proceso de Gestión de Riesgos.*

<b>Entrada</b>	<b>Subproceso</b>	<b>Salida</b>
<ul style="list-style-type: none"> <li>• Documento Visión del Proyecto (DVP).</li> <li>• Documento Plan del Proyecto (DPP).</li> </ul>	Identificación de los riesgos	<ul style="list-style-type: none"> <li>• Lista de Riesgos (LR).</li> </ul>
<ul style="list-style-type: none"> <li>• Lista de Riesgos (LR).</li> </ul>	Planificación de respuestas a los riesgos	<ul style="list-style-type: none"> <li>• Plan de Mitigación de Riesgos (PMR).</li> <li>• Plan de Contingencia de Riesgos (PCR).</li> </ul>
<ul style="list-style-type: none"> <li>• Plan de Mitigación de Riesgos (PMR).</li> <li>• Plan de Contingencia de Riesgos (PCR).</li> </ul>	Supervisión de los Riesgos.	<ul style="list-style-type: none"> <li>• Plan de Mitigación de Riesgos Actualizado (PMRA).</li> <li>• Plan de Contingencia de Riesgos Actualizado (PCRA).</li> </ul>

### **5.3.3.3. Proceso Técnico de Análisis y definición de requisitos**

Comprende los siguientes pasos o subprocesos: modelo de proceso de negocio, descubrimiento de requisitos, análisis de los requisitos, especificación de los requisitos y, gestión de requisitos; éstos se visualizan en la figura 5.6.

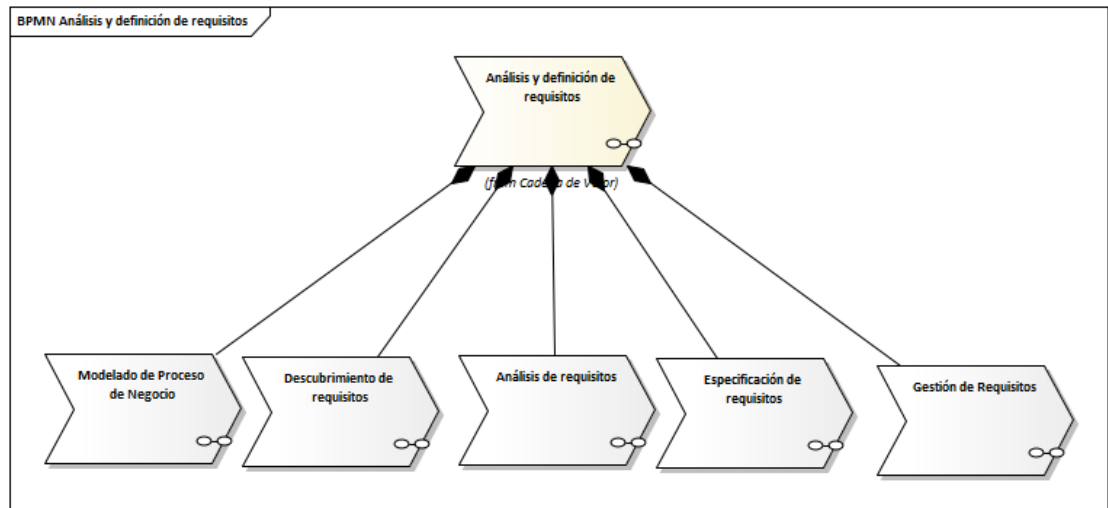


Figura 5.6. Diagrama de jerarquía del proceso de análisis y definición de requisitos.

A continuación, se detallan las actividades, técnicas y/o notaciones y productos generados en este proceso (ver tablas 5.6, 5.7 y 5.8).

Tabla 5.6. Descripción del proceso de análisis y definición de requisitos (a)

Pasos/ Subprocesos	Actividades	Técnicas y/o notaciones	Productos
1. Modelado de Proceso de Negocio	<ul style="list-style-type: none"> <li>Definir el (los) objetivo(s) del proceso, las entradas y salidas específicas, los recursos consumidos, la secuencia de las actividades, la información y los eventos que dirigen el proceso.</li> </ul>	<ul style="list-style-type: none"> <li>Entrevistas.</li> <li>Herramientas de apoyo especializadas para modelar.</li> <li>UML <i>Business</i>.</li> </ul>	<ul style="list-style-type: none"> <li>Modelo de Proceso de Negocio (MPN).</li> </ul>

Tabla 5.7. Descripción del proceso de Análisis y definición de requisitos (b).

Pasos/ Subproceso	Actividades	Técnicas y/o notaciones	Productos del documento Especificación de Requisitos de Software (ERS)
2. Descubrimiento de requisitos	<ul style="list-style-type: none"> <li>• Definir las técnicas de elicitación.</li> <li>• Describir los requisitos del usuario</li> <li>• Identificar las restricciones de la aplicación.</li> <li>• Definir los actores de la aplicación.</li> </ul>	<ul style="list-style-type: none"> <li>• Entrevistas.</li> <li>• Reuniones.</li> <li>• Cuestionarios</li> <li>• Observación.</li> <li>• Prototipos no funcionales (<i>papermock-up</i>).</li> <li>• Técnicas de comunicación formal e informal.</li> </ul>	<ul style="list-style-type: none"> <li>• Historias de Usuarios (HU).</li> <li>• Prototipos No Funcionales. (PNF)</li> <li>• Informe de Observaciones (IO).</li> <li>• Cuestionarios de Requisitos (CR).</li> <li>• Lista de restricciones de la Aplicación (LRA).</li> <li>• Lista de actores de la aplicación (LAA).</li> </ul>
3. Análisis de requisitos	<ul style="list-style-type: none"> <li>• Identificar requisitos funcionales y no funcionales.</li> <li>• Elaborar una tabla de los requisitos funcionales especificando: el identificador del requisito, su clasificación (Entrada, Proceso, Salida), los actores involucrados y la prioridad.</li> <li>• Elaborar una tabla por cada tipo de requisitos no funcionales de: <i>hardware</i>, <i>software</i>, interfaz y de calidad (ISO 9126). Especificando: el identificador del requisito, la descripción del requisito, los actores involucrados y la prioridad.</li> <li>• Elaborar una tabla con los datos de los usuarios de la aplicación: el identificador, la descripción, tipología (primario, secundario) y el respectivo rol.</li> <li>• Validar los requisitos con el usuario.</li> </ul>	<ul style="list-style-type: none"> <li>• Reuniones.</li> <li>• Técnicas de comunicación formal e informal</li> </ul>	<ul style="list-style-type: none"> <li>• Formato de Especificación de Requisitos Funcional (FERF).</li> <li>• Formato de Especificación de requisitos No Funcionales: <ul style="list-style-type: none"> <li>○ De <i>Hardware</i> (FERNF-H)</li> <li>○ De <i>Software</i> (FERNF-S).</li> <li>○ De Interfaz (FERNF-I).</li> <li>○ De Calidad (FERNF-C).</li> </ul> </li> <li>• Tabla de los Usuarios de la Aplicación (TUA).</li> <li>• Tabla de Validación de los Requisitos con el Usuario (TVU).</li> </ul>
4. Especificación de requisitos	<ul style="list-style-type: none"> <li>• Elaborar la Matriz de trazabilidad requisitos funcionales vs. Actores y la matriz requisitos vs. requisitos.</li> <li>• Elaborar los diagramas de casos de uso: se realiza la representación gráfica con la notación UML.</li> <li>• Realizar la descripción narrativa de los Casos de Uso.</li> <li>• Elaborar el documento Especificación de Requisitos de Software (ERS).</li> </ul>	<ul style="list-style-type: none"> <li>• Notación UML.</li> <li>• Herramientas de apoyo especializadas para modelar.</li> </ul>	<ul style="list-style-type: none"> <li>• Matriz de Trazabilidad requisitos vs Actores (MTRA).</li> <li>• Matriz de Trazabilidad requisitos vs requisitos (MTRR).</li> <li>• Diagramas de Casos de Uso (DCU).</li> <li>• Descripción narrativa de los CU (DNCU)</li> <li>• Documento de Especificación de Requisitos de Software (ERS).</li> </ul>

Tabla 5.8. Descripción del proceso de Análisis y definición de requisitos (c).

Pasos/ Subproceso	Actividades	Técnicas y/o notaciones	Productos del Documento Gestión de Requisitos (DGR)
5. Gestión de Requisitos	<ul style="list-style-type: none"> <li>• Identificar los requisitos a cambiar.</li> <li>• Analizar el impacto del cambio: Realizar una Matriz de Trazabilidad requisitos vs requisitos.</li> <li>• Documentar el cambio en el ERS generando otra versión del mismo.</li> </ul>	<ul style="list-style-type: none"> <li>• Herramientas de apoyo especializadas.</li> <li>• Reuniones de reflexión con el cliente.</li> </ul>	<ul style="list-style-type: none"> <li>• ERS cambiado.</li> </ul>

Definiciones de las técnicas:

- Entrevistas: es un intercambio verbal de información entre una persona (el entrevistador) y otra (el entrevistado) u otras (entrevistados), con el fin de recopilar información durante el encuentro (Hernández, Fernández & Baptista, 2006). Se llevan a cabo realizando preguntas, preparadas o espontáneas y registrando las respuestas (PMI, 2017).
- Técnicas de comunicación formal e informal: la comunicación consta de dos partes: La primera es la estrategia de comunicación y la segunda los mensajes en diversos formatos y medios, como se definen en la estrategia. Las comunicaciones en proyectos se apoyan en esfuerzos para evitar malos entendidos, los cuales se pueden reducir pero no eliminar, con el uso de las 5Cs en comunicaciones escritas: correcto, conciso, claro, coherente y controlado. La técnica formal son reuniones periódicas o *ad hoc* (para la ocasión (Real Academia Española [RAE], 2001)), en las cuales se utilizan agendas y actas de reunión, sesiones informativas para los interesados y presentaciones. Las informales son actividades de comunicación mediante correo electrónico, medios sociales, sitios Web y discusiones informales *ad hoc* (PMI, 2017).
- Cuestionarios: consiste en un documento formado por preguntas cuyas respuestas deben ser breves, específicas y concretas o incluso cerradas



por unas cuantas opciones en el propio cuestionario (*checklist*) (Escalona & Koch, 2002).

- Prototipos no funcionales (*papermock-up*): son bocetos a mano alzada, medio eficiente para elicitar requisitos de la interfaz de usuario, algunos requisitos funcionales y no funcionales (Hammar, 2001).
- Observaciones: Es una de las técnicas etnográficas más usadas, básicamente consiste como su nombre lo dice en observar cada uno de los procesos existentes; las personas interesadas deben poseer un alto grado de análisis que conlleve a interpretar y a comprender de manera eficaz cada uno de los procesos observados (Aybüke & Wohlin, 2005).

En la tabla 5.9 y 5.10 se describen las entradas y salidas del cada subproceso.

Tabla 5.9. *Entradas y salidas del proceso de Análisis y definición de requisitos (a).*

Entrada	Subproceso	Salida
<ul style="list-style-type: none"> <li>• Documento Visión del Proyecto (DVP).</li> </ul>	Modelado de proceso de negocio	<ul style="list-style-type: none"> <li>• Modelo de Proceso de Negocio (MPN).</li> </ul>
<ul style="list-style-type: none"> <li>• Documento Visión del Proyecto (DVP).</li> <li>• Modelo de Proceso de Negocio (MPN).</li> </ul>	Descubrimiento de requisitos	<ul style="list-style-type: none"> <li>• Historias de Usuarios (HU).</li> <li>• Prototipos No Funcionales. (PNF)</li> <li>• Informe de Observaciones (IO).</li> <li>• Cuestionarios de Requisitos (CR).</li> <li>• Lista de restricciones de la aplicación (LRA)</li> <li>• Lista de actores de la aplicación (LAA).</li> </ul>
<ul style="list-style-type: none"> <li>• Historias de Usuarios (HU).</li> <li>• Prototipos No Funcionales. (PNF)</li> <li>• Informe de Observaciones (IO).</li> <li>• Cuestionarios de Requisitos (CR).</li> <li>• Lista de las restricciones de la aplicación (LRA)</li> <li>• Lista de actores de la aplicación (LAA)</li> </ul>	Análisis de los requisitos	<ul style="list-style-type: none"> <li>• Formato de Especificación de Requisitos Funcionales (FERF).</li> <li>• Formato de Especificación de Requisitos No funcionales de: <ul style="list-style-type: none"> <li>• <i>Hardware</i> (FERNF-H)</li> <li>• <i>Software</i> (FERNF-S)</li> <li>• Interfaz (FERNF-I)</li> <li>• Calidad (FERNF-C)</li> </ul> </li> <li>• Tabla de los Usuarios de la Aplicación (TUZ).</li> <li>• Tabla de Validación de los requisitos con el Usuario (TVU).</li> </ul>

Tabla 5.10. Entradas y salidas del proceso de Análisis y definición de requisitos (b).

Entrada	Subproceso	Salidas
<ul style="list-style-type: none"> <li>• Formato de Especificación de Requisitos Funcionales (FERF).</li> <li>• Formato de Especificación de Requisitos No funcionales de:                             <ul style="list-style-type: none"> <li>○ Hardware (FERNF-H)</li> <li>○ Software (FERNF-S)</li> <li>○ Interfaz (FERNF-I)</li> <li>○ Calidad (FERNF-C)</li> </ul> </li> <li>• Tabla de los Usuarios de la Aplicación (TUZ).</li> <li>• Tabla de Validación de los requisitos con el Usuario (TVU).</li> </ul>	Especificación de los requisitos	<ul style="list-style-type: none"> <li>• Matriz de Trazabilidad requisitos vs Actores (MTRA).</li> <li>• Diagramas de Casos de Uso (DCU).</li> <li>• Descripción Narrativa de los CU (DNCU).</li> <li>• Documento de Especificación de Requisitos de Software (ERS).</li> </ul>
<ul style="list-style-type: none"> <li>• ERS</li> </ul>	Gestión de Requisitos	<ul style="list-style-type: none"> <li>• ERS cambiado</li> </ul>

### 5.3.3.4. Proceso Técnico de Diseño

Comprende los siguientes pasos o subprocesos: definición de la arquitectura de la aplicación, elaboración del modelo estático y dinámico, construcción del modelo de datos, diseño de la interfaz y diseño del modelo de navegación (ver figura 5.7).

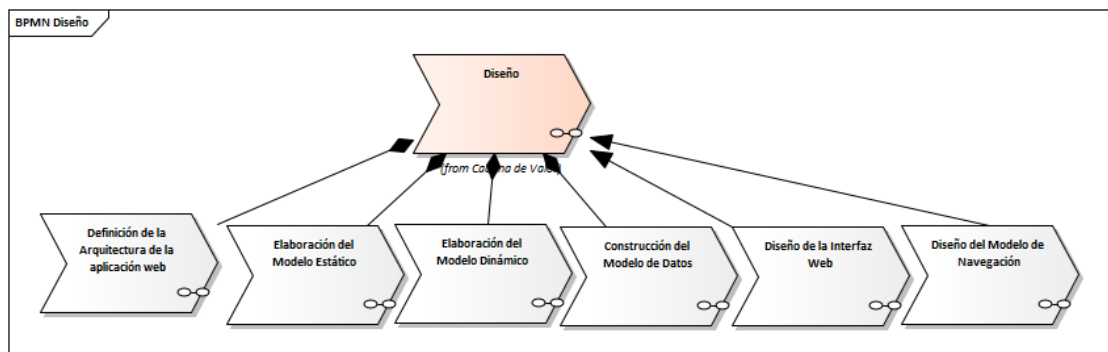


Figura 5.7. Diagrama de jerarquía del proceso de Diseño.

A continuación, se detallan las actividades, técnicas y/o notaciones y productos generados en este proceso (ver tablas 5.11 y 5.12).

Tabla 5.11. Descripción del proceso de Diseño (a)

Pasos /Subprocesos	Actividades	Técnicas y/o notaciones	Productos del Documento de Diseño (DD)
1. Definición de la Arquitectura de la aplicación Web	<ul style="list-style-type: none"> <li>• Diseñar la arquitectura del <i>software</i>.</li> <li>• Elaborar el Diagrama de Componentes.</li> <li>• Diagrama de los módulos.</li> <li>• Diseñar la arquitectura del contenido.</li> </ul>	<ul style="list-style-type: none"> <li>• Modelo Vista Controlador (MVC).</li> <li>• Modelado UML.</li> <li>• Modelado de Procesos en UML <i>Business</i>.</li> <li>• Notación semiformal (gráficos, imágenes).</li> <li>• Herramientas de apoyo especializadas para modelar procesos, elaborar diagramas UML y diseñar arquitectura de <i>software</i> y contenido.</li> </ul>	<ul style="list-style-type: none"> <li>• Documento Diseño de la Arquitectura Web (DAW) que contiene:               <ul style="list-style-type: none"> <li>○ Arquitectura de la aplicación Web. (AAW)</li> <li>○ Diagrama de los Módulos (DM).</li> <li>○ Diagrama de Componentes. (DDC)</li> <li>○ Arquitectura del Contenido Web (ACW).</li> </ul> </li> </ul>
2. Elaboración del Modelo Estático	<ul style="list-style-type: none"> <li>• Elaborar el Diagrama de Objetos.</li> <li>• Elaborar el Diagrama de Clases.</li> </ul>	<ul style="list-style-type: none"> <li>• Modelado de UML.</li> </ul>	<ul style="list-style-type: none"> <li>○ Modelo Estático (ME).</li> </ul>
3. Elaboración del Modelo Dinámico	<ul style="list-style-type: none"> <li>• Elaborar el Diagrama de Secuencia.</li> <li>• Elaborar el Diagrama de Colaboración.</li> </ul>	<ul style="list-style-type: none"> <li>• Modelado de UML.</li> </ul>	<ul style="list-style-type: none"> <li>○ Modelo Dinámico (MD).</li> </ul>
4. Construcción del Modelo de Datos.	<ul style="list-style-type: none"> <li>• Realizar el modelo entidad relación (E-R).</li> <li>• Realizar el diseño lógico.</li> <li>• Construir el diseño físico.</li> <li>• Validar el modelo de datos con el Profesor Evaluador (PE).</li> </ul>	<ul style="list-style-type: none"> <li>• Normalización de la Base de Datos.</li> <li>• Herramientas de apoyo especializadas</li> </ul>	<ul style="list-style-type: none"> <li>• Documento de Diseño de la Base de Datos (DBD) que contiene:               <ul style="list-style-type: none"> <li>○ Modelo Entidad Relación de la Base de Datos (ER-BD).</li> <li>○ Diseño Lógico de la Base de Datos (DL-BD).</li> <li>○ Diseño Físico de la Base de Datos (DF-BD).</li> <li>○ Documento Validado con el PE (DBD-VPE).</li> </ul> </li> </ul>

Tabla 5.12. Descripción del proceso de Diseño (b).

Pasos /Subprocesos	Actividades	Técnicas y/o notaciones	Productos del Documento de Diseño (DD)
5. Diseño de la interfaz Web (IW)	<ul style="list-style-type: none"> <li>• Listar sitios similares con sus bocetos.</li> <li>• Realizar bocetos de sitios similares.</li> <li>• Aplicar la técnica de ordenación de tarjetas para definir los prototipos.</li> <li>• Definir el estilo visual a través de prototipos de la interfaz Web:               <ul style="list-style-type: none"> <li>○ Plantillas con la estructura general de las paginas (cabecera, menús, pie de página, secciones)</li> <li>○ Plantillas de los siguientes elementos: formularios, listas, botones, tablas.</li> <li>○ Plantillas de páginas de insertar, consultar, modificar y/o eliminar.</li> </ul> </li> <li>• Cada prototipo debe contener un id, la técnica de interacción visual aplicada, descripción de la función de la plantilla.</li> <li>• Revisar los prototipos con el usuario.</li> <li>• Diseñar las interfaces a través de los prototipos validados por el usuario.</li> <li>• Evaluar la facilidad de uso (<i>usability</i>) de la interfaz Web utilizando la técnica de evaluación heurística.</li> <li>• Generar un informe indicando la técnica, cambios, observaciones de los cambios.</li> </ul>	<ul style="list-style-type: none"> <li>• Técnicas de comunicación formal o informal (correos, observaciones en los productos del documento de diseño).</li> <li>• Técnica de Análisis Competitivo.</li> <li>• Técnica ordenación de tarjetas (<i>cardsorting</i>).</li> <li>• Técnicas de prototipos grises.</li> <li>• Técnica de evaluación heurística.</li> </ul>	<ul style="list-style-type: none"> <li>• Documento de la Interfaz Web (DIW) que contiene:               <ul style="list-style-type: none"> <li>○ Lista de sitios similares (LSS).</li> <li>○ Tarjetas etiquetadas validadas por el usuario (TEVU).</li> <li>○ Prototipos de la Interfaz Web (PIW).</li> <li>○ Prototipos de la Interfaz Web Aprobados por el Usuario (PIWAU).</li> <li>○ Reporte de evaluación de la técnica de evaluación heurística de la facilidad de uso (<i>Usability</i>) (REHU).</li> <li>○ Informe de los cambios de la Interfaz Web (ICIW).</li> </ul> </li> </ul>
6. Diseño del Modelo de Navegación.	<ul style="list-style-type: none"> <li>• Diseñar el modelo de navegación (MN):               <ul style="list-style-type: none"> <li>○ De la Aplicación (A) y</li> <li>○ De Usuarios (U).</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Modelado UML-extensión WAE "Web Application Extensión"</li> </ul>	<ul style="list-style-type: none"> <li>○ Modelo de Navegación de la Aplicación (MNA).</li> <li>○ Modelo de Navegación de Usuarios (MNU).</li> </ul>

Descripción de las técnicas:

- **Modelo Vista Controlador (MVC):** es un patrón de diseño de *software* que separa los datos y la lógica de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador (Krasner & Pope, 1988).

- Técnica de Análisis Competitivo: implica estudiar productos existentes similares para descubrir sus cualidades y deficiencias; esto ayuda a desarrollar el concepto del producto y a generar ideas de diseño (Nielsen, 1993). Esta técnica se utilizará para definir las Listas de los Sitios Similares (LSS), con los aspectos que se seleccionan para el diseño y la dirección del sitio (*url*).
- Ordenación de tarjetas (*cardsorting*): ésta se basa en la observación de cómo los usuarios agrupan y asocian entre sí un número predeterminado de tarjetas etiquetadas con las diferentes categorías o secciones temáticas del diseño o un producto, servicio o cualquier *item*. De esta forma, partiendo del comportamiento de los propios usuarios, es posible organizar y clasificar la información Web conforme a su modelo mental. También puede contribuir con el diseño de prototipo, pues la estructura de los menús puede establecerse basándose en la información obtenida por medio de la aplicación de esta técnica (Hassan, 2004).
- Técnica de prototipos grises: el estándar ISO 13407 define el prototipo como una representación de todo o parte de un producto o sistema que, aunque limitado de algún modo, puede utilizarse con fines de evaluación (ISO, 1999). Los prototipos grises son páginas con enlaces funcionales y contenidos que se enfocan únicamente en la información que cada página o sección debe tener. Son llamados grises ya que no contempla el diseño sino la funcionalidad con el objetivo que los desarrolladores y clientes se enfoquen en el contenido sin distractores (Holter, 2006).
- Evaluación heurística: es un método de inspección que no requiere la participación de usuarios y se puede realizar en cualquier momento del

proyecto porque es sencilla y económica. Se evalúa principios reconocidos definidos por un experto, como Jakob Nielsen. Las revisiones pueden ser efectuados por evaluadores con o sin experiencia en la facilidad de uso (*usability*) (Pintos, 2014). Se presentarán los diez (10) principios de Nielsen (1994) y los ocho (8) definidos por Shneiderman (1986a) (ver tablas 5.13 y 5.14).

Tabla 5.13. *Los 10 principios heurísticos de Nielsen (1994).*

<b>Heurística</b>	<b>Descripción</b>
# 1: Visibilidad del estado del sistema.	El sistema siempre debe informar al usuario lo que está sucediendo por medio de comentarios apropiados dentro de un tiempo razonable. Ejemplo: cambiar el color de un botón o enlace cuando se presiona o selecciona respectivamente.
# 2: Coincidencia entre el sistema y el mundo real.	El sistema debe hablar el idioma de los usuarios con palabras, frases y conceptos familiares para el usuario en lugar de términos orientados al sistema; y debe seguir convenciones del mundo real. Ejemplo: las herramientas de marcar en amarillo párrafos en un libro digital.
# 3: Control del usuario y libertad.	Incluir acciones de deshacer y rehacer. Ejemplo los botones de retroceder y avanzar en un navegador Web.
# 4: Consistencia y estándares.	Aplicar normas y convenciones de los sistemas conocidos. Ejemplo: la imagen de lupa como metáfora para la acción buscar.
# 5: Prevención de errores.	Eliminar las condiciones propensas a errores. Ejemplo: enviar mensajes de confirmación con los datos del beneficiario y monto al realizar una transferencia bancaria.
# 6: Reconocimiento en lugar de recordar.	Minimice la carga de memoria del usuario haciendo visibles los objetos, acciones y las opciones. Ejemplo: utilizar una imagen para subrayar en un editor de texto en vez de ubicarlo en un menú de opciones de forma textual.
# 7: Flexibilidad y eficiencia de uso.	Permitir a los usuarios personalizar las acciones frecuentes. Son aceleradores no vistos por el usuario novato que a menudo aceleran la interacción para usuarios expertos. Ejemplo: utilizar las teclas: ctrl-c para copiar.
# 8: Diseño estético y minimalista.	Eliminar información irrelevante.
# 9: Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de errores.	Un mensaje de error debe expresarse en lenguaje sencillo sin códigos. Indicar qué falló y cómo solucionarlo. Ejemplo: páginas con mensaje de error de programación el cual se despliega en el lenguaje formal de código y no indica qué se debe hacer.
# 10: Ayuda y documentación.	Proporcionar ayuda y documentación. Información fácil de localizar, centrada en la tarea del usuario, enumerar los pasos concretos que se deben llevar a cabo de forma breve y concisa.

Tabla 5.14. *Aplicación de la evaluación heurística de Shneiderman (1986a).*

<b>Regla de Oro</b>	<b>Descripción</b>
# 1: Esforzarse por la consistencia.	Las acciones en situaciones similares deben ser consistentes. Se debe utilizar una terminología idéntica en las pantallas de entradas, los menús, las ventanas de ayuda; el color consistente, el diseño, las mayúsculas, las fuentes, entre otros.
# 2: Busque la facilidad de uso ( <i>usability</i> ) universal.	Reconocer las necesidades de diversos usuarios. Agregar funciones para principiantes como explicaciones y funciones para expertos, como atajos. Ejemplo: utilizar las teclas ctrl-z (deshacer último cambio), tab (movilizarse de un campo a otro), entre otros.
# 3: Ofrecer comentarios informativos	Para cada acción del usuario, debe haber una retroalimentación de la interfaz. Para acciones frecuentes y menores, la respuesta puede ser modesta, mientras que para acciones poco frecuentes e importantes, la respuesta debería ser más sustancial.
# 4: Diseñar cuadros de diálogo para producir el cierre.	La retroalimentación informativa al completar un grupo de acciones brinda a los usuarios la satisfacción del logro, una sensación de alivio, una señal para abandonar los planes de contingencia de sus mentes y un indicador para prepararse para el próximo grupo de acciones. Por ejemplo, los sitios Web de comercio electrónico hacen que los usuarios seleccionen productos para el pago y finalizan con una página de confirmación clara que completa la transacción.
# 5: Prevenir errores.	En la medida de lo posible, diseñe la interfaz para que los usuarios no puedan cometer errores graves. Ejemplo: ingresar valores numéricos en campos de valores alfabéticos.
# 6: Permitir la reversión fácil de acciones.	En la medida de lo posible, las acciones deben ser reversibles. Esta característica alivia la ansiedad, ya que los usuarios saben que los errores se pueden deshacer y alienta la exploración de opciones desconocidas. Ejemplo: descartar un correo al enviarlo por equivocación.,
# 7: Mantenga a los usuarios en control.	Los usuarios experimentados desean sentir que están a cargo de la interfaz y que la interfaz responde a sus acciones.
# 8: Reduzca la carga de memoria a corto plazo	La limitación del procesamiento de la información humana en la memoria a corto plazo requiere procedimientos simplificados de entrada de datos, pantallas comprensibles y comentarios informativos rápidos.

Se puede observar tres (3) heurísticas similares. La 5,6 y 7 de Nielsen (1994) con 5,7 y 8 de Shneiderman (1986a), respectivamente.

La técnica es utilizada agregando una columna que represente el nivel de cumplimiento con una medida de evaluación, que puede ser rangos (0-9) o porcentajes (0-100) (ver tabla 5.15).

Tabla 5.15. *Aplicación de los principios heurísticos de autor (año).*

<b>Heurística</b>	<b>Nivel de cumplimiento(0-100)%</b>

En la tabla 5.16 se describen las entradas y salidas de cada subproceso.

Tabla 5.16. *Entradas y salidas del proceso de Diseño.*

<b>Entrada</b>	<b>Subproceso</b>	<b>Salida</b>
<ul style="list-style-type: none"> <li>• ERS</li> </ul>	Definición de la arquitectura de la aplicación Web.	<ul style="list-style-type: none"> <li>• Arquitectura de la aplicación Web. (AAW)</li> <li>• Diagrama de los Módulos (DM).</li> <li>• Diagrama de Componentes. (DDC)</li> <li>• Arquitectura del Contenido Web (ACW).</li> </ul>
<ul style="list-style-type: none"> <li>• Formatos de Especificación de Requisitos Funcionales (FERF).</li> </ul>	Elaboración del Modelo Estático	<ul style="list-style-type: none"> <li>• Modelo Estático (ME)</li> </ul>
<ul style="list-style-type: none"> <li>• Descripción Narrativa de los CU (DNCU).</li> </ul>	Elaboración del Modelo Dinámico	<ul style="list-style-type: none"> <li>• Modelo Dinámico (MD).</li> </ul>
<ul style="list-style-type: none"> <li>• Modelo Estático (ME).</li> </ul>	Construcción del Modelo de Datos.	<ul style="list-style-type: none"> <li>• Documento del Diseño de la Base de Datos (DDBD).</li> <li>• Documento del Diseño de la Base de Datos (DDBD-VPE).</li> </ul>
<ul style="list-style-type: none"> <li>• ERS</li> <li>• Arquitectura del Contenido Web (ACW).</li> <li>• Documento del Diseño de la Base de Datos (DDBD).</li> </ul>	Diseño de la Interfaz Web.	<ul style="list-style-type: none"> <li>• Documento de la Interfaz Web (DIW)</li> </ul>
<ul style="list-style-type: none"> <li>• Diagrama de los Módulos (DM).</li> <li>• Prototipos de la Interfaz Web Aprobados por el Usuario (PIWAU).</li> <li>• Lista de actores de la aplicación (LAA).</li> </ul>	Diseño del Modelo de Navegación.	<ul style="list-style-type: none"> <li>• Modelo de Navegación de la Aplicación (MNA).</li> <li>• Modelo de Navegación de usuarios (MNU).</li> </ul>



### 5.3.3.5. Proceso Técnico de Construcción & Integración

Comprende los siguientes pasos o subprocessos: construcción de la base de datos, construcción de la interfaz, codificación de los componentes, ejecución de las pruebas unitarias e integración de los componentes (ver figura 5.8).

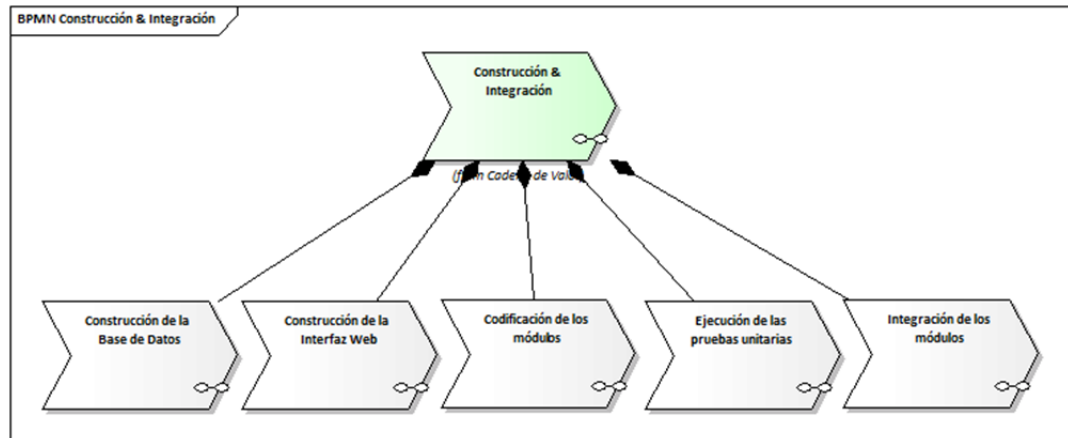


Figura 5.8. Diagrama de jerarquía del proceso de Construcción & Integración.

A continuación, se detallan las actividades, técnicas y/o notaciones y productos generados en este proceso (ver tablas 5.17 y 5.18).

Tabla 5.17. Descripción del proceso de Construcción & Integración (a).

Pasos /Subprocesos	Actividades	Técnicas y/o notaciones	Productos
1. Construcción de la Base de Datos	<ul style="list-style-type: none"> <li>• Construir la Base de Datos.</li> <li>• Comprobar las propiedades: atomicidad, consistencia, aislamiento y durabilidad (ACID) ejecutando consultas.</li> <li>• Validar la Base de Datos con el Profesor Evaluador (PE).</li> </ul>	<ul style="list-style-type: none"> <li>• Técnica de migración de base de datos: sentencias de lenguaje de manipulación de datos (DML).</li> </ul>	<ul style="list-style-type: none"> <li>• Base de Datos.</li> <li>• Base de Datos probada y validada por el (PE).</li> </ul>
2. Construcción de la Interfaz Web (IW)	<ul style="list-style-type: none"> <li>• Diseñar la interfaz.</li> <li>• Realizar las pruebas de navegación.</li> <li>• Ejecutar las técnicas de facilidad de uso (<i>usability</i>).</li> </ul>	<ul style="list-style-type: none"> <li>• Técnica de pensar en alto (<i>ThinkingAloud</i>).</li> <li>• Técnica de inspecciones de consistencia.</li> <li>• Técnica: Cuestionario de</li> </ul>	<ul style="list-style-type: none"> <li>• Casos de Pruebas de Navegación y facilidad de uso (<i>usability</i>) ejecutadas (CPNU)</li> </ul>

Tabla 5.18. Descripción del proceso de Construcción & Integración(b).

		<p>facilidad de uso.</p> <ul style="list-style-type: none"> <li>Herramientas de apoyo especializadas para la evaluación de la accesibilidad y facilidad de uso.</li> </ul>	<p>en el documento Informe de Ejecución de Pruebas (IEP). Interfaz probada</p>
3.Codificación de los módulos	<ul style="list-style-type: none"> <li>Codificar los módulos.</li> </ul>	<ul style="list-style-type: none"> <li>Técnicas para crear código fuente comprensible.</li> </ul>	<ul style="list-style-type: none"> <li>Programas.</li> <li>Módulos.</li> </ul>
4.Ejecución de pruebas unitarias	<ul style="list-style-type: none"> <li>Ejecutar las pruebas unitarias.</li> <li>Depurar los errores encontrados.</li> </ul>	<ul style="list-style-type: none"> <li>Técnicas de Caja Blanca:                             <ul style="list-style-type: none"> <li>Pruebas de la ruta básica.</li> <li>Estructuras de control.</li> </ul> </li> <li>Técnicas de caja negra:</li> </ul>	<ul style="list-style-type: none"> <li>Casos de Pruebas Unitarias Ejecutadas (CPUE) en el documento Informe de Ejecución de Pruebas (IEP).</li> </ul>
5.Integración de los módulos	<ul style="list-style-type: none"> <li>Ejecutar las pruebas unitarias.</li> <li>Depurar los errores encontrados.</li> </ul>	<ul style="list-style-type: none"> <li>Partición equivalente.</li> <li>Análisis de valores límites</li> <li>Herramientas de apoyo especializadas de pruebas.</li> <li>Herramientas especializadas de control y seguimiento de errores (<i>Bug Tracking System</i>, BTS).</li> </ul>	<ul style="list-style-type: none"> <li>Especificación de Pruebas de Integración (EPI) en el documento Informe de Ejecución de Pruebas (IEP).</li> </ul>

Descripción de las técnicas:

- Técnica de migración de base de datos- sentencias de lenguaje de manipulación de datos: por medio de este lenguaje el usuario puede realizar consultas o manipular datos. El lenguaje más utilizado es el SQL (*StructuredQueryLanguage*). Los comandos son: *select*, *insert*, *delete* y *update*(Leguizamón, A (2017; Cardona, 2017).
- Técnicas para crear código fuente comprensible: según SWEBOK (Bourque & Fairley, 2014), se sugieren: (1) convenciones de nomenclatura y diseño del código fuente; (2) uso de clases, variables, constantes con nombres similares; (3) manejo de condiciones de error; (4) documentación del código.

- Técnicas de pensar en alto (*ThinkingAloud*): en esta técnica se les solicita a los usuarios que expresen en voz alta sus pensamientos, sentimientos y opiniones sobre cualquier aspecto (diseño, funcionalidad, entre otros) mientras interactúa con el sistema. El usuario puede indicar que la secuencia que le establece la interfaz para el objetivo de su tarea no es lo que el usuario esperaba (Nielsen, 1993).
- La técnica de inspecciones de consistencia: se basa en verificar la consistencia en color, terminología, disposición de elementos en la interfaz, formatos o pantallas de entrada/salida, entre otros, de las interfaces de usuario (O'Mahony, 2003). El evaluador Profesor puede apoyarse en la técnica de lista de verificación (*checklist*).
- El cuestionario a utilizar debe ser definido por el profesor evaluador en conjunto con el equipo de desarrolladores. Se presentan tres (3) cuestionarios:
  - Cuestionario desarrollado por el Dr. David Travis (@userfocus) basado en los principios de Nielsen (1994). La documentación está en formato excel y disponible en: <https://www.userfocus.co.uk/resources/guidelines.html>. Los criterios de este cuestionario se visualizan en la tabla 5.19.

Tabla 5.19. *Criterios del cuestionario de facilidad de uso de Travis (2012).*

<b>Criterio a evaluar</b>	<b># de preguntas</b>
1. Página de inicio	20
2. Orientación a tareas y funcionalidad del sitio	44
3. Navegabilidad y arquitectura de la información	29
4. Formularios y entrada de datos	23
5. Confianza y credibilidad	13
6. Calidad del contenido y escritura	23
7. Diagramación y diseño gráfico	38
8. Ayuda, retroalimentación & recuperación de errores	20
9. Ayuda	37

- SIRIUS (Suárez, 2011): Es un sistema de evaluación de la facilidad de uso basado en heurísticas, se presenta en formato excel y da como resultado un valor porcentual del nivel de facilidad de uso (*usability*). Disponible en: <https://olgacarreras.blogspot.com/2011/07/sirius-nueva-sistema-para-la-evaluacion.html>. Los criterios de esta lista se visualizan en la tabla 5.20.

Tabla 5.20. *Criterios del cuestionario de facilidad de uso SIRIUS (2011).*

<b>Criterio a evaluar</b>	<b># de preguntas</b>
1. Aspectos generales	10
2. Identidad e información	7
3. Estructura de navegación	14
4. Rotulado	6
5. <i>Layout</i> de la página	10
6. Entendibilidad y factibilidad	7
7. Control y retroalimentación	10
8. Elementos multimedia	6
9. Búsqueda	8
10. Ayuda	5

- El cuestionario para satisfacción del usuario de la interacción (*The Questionnaire for User Interaction Satisfaction (QUIS)*) (Shneiderman, 1986b; Chin, Diehl & Norman, 1988) es un producto diseñado por el laboratorio de interacción humano-computador de la Universidad de Maryland. Los criterios a evaluar son cinco (5) (ver tabla 5.21). y los *item's* en cada criterio se visualiza en la tabla 5.22. Actualmente está disponible la versión 7.0. La fiabilidad global (coeficiente  $\alpha$ ) es de 0.94, pero no existe una fiabilidad de sub-escala. (Chin, Diehl & Norman, 1988). El anexo B contiene la media en sus dos valores límites de cada uno de los 27 *ítem's*. del cuestionario QUIS.

Tabla 5.21. *Criterios del cuestionario de facilidad de uso QUIS (Shneiderman, 1986b,Chin, Diehl & Norman, 1988).*

Criterio a evaluar	# de preguntas
1. Reacciones globales del software	6
2. Pantalla	4
3. Terminología	6
4. Aprendizaje	6
5. Capacidades del sistema	5

Tabla 5.22. *Cuestionario para la satisfacción de la interfaz de usuario (QUIS; Shneiderman, 1986b,Chin, Diehl & Norman, 1988).*

Reacciones globales del software		0	1	2	3	4	5	6	7	8	9	NA
1.	terrible											maravilloso
2.	difícil											fácil
3.	frustrante											satisfactorio
4.	débil											muy potente
5.	aburrido											estimulante
6.	rígido											flexible
Pantalla		0	1	2	3	4	5	6	7	8	9	NA
7.	El texto en la pantalla son difícil de leer											fácil de leer
8.	Resaltado simplifica la tarea no siempre											muchas veces
9.	Organización de la información confusa											muy clara
10.	Secuencia de pantallas confusa											muy clara
Terminología		0	1	2	3	4	5	6	7	8	9	NA
11.	Uso de términos en todo el sistema inconsistente											consistente
12.	Está relacionada con la tarea que realiza nunca											siempre
13.	Posición de los mensajes inconsistente											consistente
14.	Descripción de las entradas de datos confuso											claro
15.	Informa sobre el progreso nunca											siempre
16.	Mensajes de error No entendibles											entendibles
Aprendizaje		0	1	2	3	4	5	6	7	8	9	NA
17.	El aprendizaje de usar el sistema es difícil											fácil
18.	Explorar nuevas funciones por ensayo y error difícil											fácil
19.	Recordar nombres y usar comandos difícil											fácil
20.	Ejecutas las tareas es sencillo nunca											siempre
21.	Los mensajes de ayuda en la pantalla son Difícil de entender											Fácil de entender
22.	Material de referencia complementarios confuso											claro
Capacidades del sistema		0	1	2	3	4	5	6	7	8	9	NA
23.	Velocidad del sistema Muy lento											rápido
24.	El sistema es fiable nunca											siempre
25.	Los sonidos generados son ruidoso											Silencioso
26.	Corregir los errores difícil											fácil
27.	Diseñado para todos los niveles de usuarios nunca											siempre

- Técnica de caja blanca - ruta básica: consiste en definir un conjunto básico de rutas de ejecución del programa. Los casos de prueba deben garantizar que se ejecute cada instrucción del programa por lo menos una vez durante la prueba (Pressman, 2005).
- Técnica de caja blanca- estructuras de control: se evalúan las pruebas de condición y pruebas de bucles. El método de pruebas de condición se concentra en la prueba de cada condición del programa. Se evalúan los operadores booleanos (faltantes, incorrectos y/o adicionales), los paréntesis booleanos, operadores relacionales y expresión aritmética. El método de pruebas de bucles se concentra exclusivamente en la validez de los diferentes tipos de bucles: simples, anidados y concatenados.
- Técnica de caja negra - partición de equivalencia: implica la partición de dominio de entrada en una colección de subconjuntos(o clases equivalentes) de entradas válidas que son aceptadas y procesadas por sistemas y, de entradas no válidas que no son aceptadas y que deberían mostrar un mensaje de error; en ambos casos para las condiciones de entrada. Una condición de entrada es un valor numérico, un rango de valores, un conjunto de valores relacionados o una condición booleana (Pressman, 2005).
- Técnica de caja negra - análisis de valores límites: los casos de prueba se eligen cerca de los límites del dominio de entrada de variables (Pressman, 2005).

En la tabla 5.23 se describen las entradas y salidas de cada subproceso.

Tabla 5.23. Entradas y salidas del proceso de Construcción & Integración.

Entrada	Subproceso	Salida
<ul style="list-style-type: none"> <li>Documento del Diseño de la Base de Datos (DDBD)</li> </ul>	Construcción de la Base de Datos	<ul style="list-style-type: none"> <li>Base de Datos probada</li> </ul>
<ul style="list-style-type: none"> <li>Documento de Diseño (DD)</li> </ul>	Construcción de la interfaz Web	<ul style="list-style-type: none"> <li>Interfaz probada</li> <li>Informe de Ejecución de Pruebas (IEP).</li> </ul>
<ul style="list-style-type: none"> <li>ERS</li> <li>Interfaz probada</li> </ul>	Codificación de los módulos	<ul style="list-style-type: none"> <li>Programas</li> <li>Módulos</li> </ul>
<ul style="list-style-type: none"> <li>Aplicación</li> </ul>	Ejecución de las pruebas unitarias	<ul style="list-style-type: none"> <li>Informe de Ejecución de Pruebas (IEP).</li> </ul>
<ul style="list-style-type: none"> <li>Plan de Pruebas</li> </ul>	Integración de los módulos	<ul style="list-style-type: none"> <li>Programas integrados</li> <li>Informe de Ejecución de Pruebas (IEP).</li> <li>Documento de Diseño Actualizado (DDA).</li> </ul>

### 5.3.3.6. Proceso Técnico de Pruebas

La entrada o insumo para este proceso es el *software* y los casos de pruebas. En la figura 5.9 se visualizan los pasos descritos en el modelo de procesos y, en la tabla 5.24, se detallan las actividades con las técnicas y/o notaciones y, productos generados en este proceso.

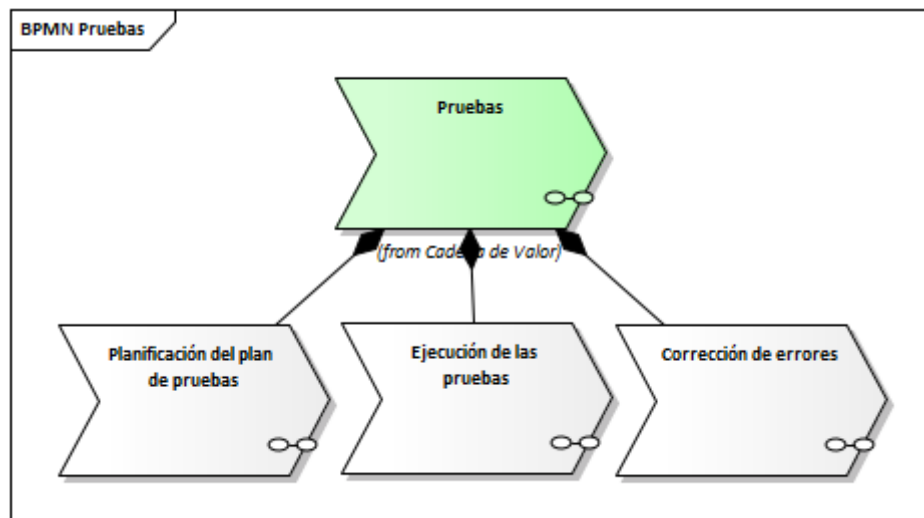


Figura 5.9. Diagrama de jerarquía del proceso de Pruebas.

Tabla 5.24. Descripción del proceso de Pruebas.

Pasos /Subprocesos	Actividades	Técnicas y/o notaciones	Productos
1. Planificación del plan de pruebas	<ul style="list-style-type: none"> <li>• Definir las técnicas de pruebas: caja blanca, negra y las técnicas de facilidad de uso (<i>usability</i>).</li> <li>• Definir los tipos de pruebas.</li> <li>• Elaborar el cronograma de pruebas.</li> <li>• Diseñar los casos de pruebas.</li> <li>• Diseñar la Lista de los Requisitos de la aplicación a probar especificando: id del requisito, descripción, el estatus ejecutado(SI/NO), observaciones</li> </ul>	<ul style="list-style-type: none"> <li>• Técnicas de Caja Blanca:                             <ul style="list-style-type: none"> <li>○ Pruebas de la ruta básica.</li> <li>○ Estructuras de control.</li> </ul> </li> <li>• Técnicas de caja negra:                             <ul style="list-style-type: none"> <li>○ Partición equivalente.</li> <li>○ Análisis de valores límites</li> </ul> </li> <li>• Diagrama Gantt.</li> </ul>	<ul style="list-style-type: none"> <li>• Documento Plan de Pruebas (DPP) que contiene:                             <ul style="list-style-type: none"> <li>○ Técnicas.</li> <li>○ Tipos de Pruebas.</li> <li>○ Cronograma de Pruebas.</li> <li>○ Casos de Pruebas (CP).</li> <li>○ Lista de Requisitos a Probar (LRP).</li> </ul> </li> </ul>
2. Ejecución de las pruebas	<ul style="list-style-type: none"> <li>• Ejecutar pruebas funcionales.</li> <li>• Ejecutar pruebas no funcionales.</li> <li>• Ejecutar pruebas de aceptación.</li> <li>• Ejecutar pruebas unitarias, de integración y de sistema.</li> <li>• Reportar los errores encontrados.</li> <li>• Documentar las pruebas.</li> <li>• Realizar las técnicas de facilidad de uso:                             <ul style="list-style-type: none"> <li>○ Técnica de pensar en alto (<i>ThinkingAloud</i>).</li> <li>○ Técnica de inspecciones de consistencia.</li> <li>○ Cuestionario de facilidad de uso (<i>usability</i>)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Herramientas de apoyo especializadas en ejecución de pruebas.</li> <li>• Técnica de depuración estática.</li> <li>• Técnica de depuración dinámica: <i>PrintDebugging</i>.</li> <li>• Técnica de depuración dinámica: depuración por trazas (<i>Trace Debugging</i>).</li> <li>• Técnicas de Caja Blanca.</li> <li>• Técnicas de caja negra.</li> <li>• Técnicas de prueba de la facilidad de uso.</li> <li>• Herramientas de seguimiento de errores (<i>Bug Tracking Tools</i>).</li> </ul>	<ul style="list-style-type: none"> <li>• Informe de Ejecución de Pruebas. (IEP).                             <ul style="list-style-type: none"> <li>○ Casos de Pruebas ejecutados (CPE).</li> <li>○ Informe de Errores (IE).</li> </ul> </li> <li>• Documento Plan de Prueba Actualizado (DPPA).</li> </ul>
3. Corrección de errores	<ul style="list-style-type: none"> <li>• Corregir los errores</li> </ul>	<ul style="list-style-type: none"> <li>• Técnica de depuración estática.</li> <li>• Técnica de depuración dinámica: <i>PrintDebugging</i>.</li> <li>• Técnica de depuración dinámica: depuración por trazas (<i>Trace Debugging</i>)</li> <li>• Herramientas de depuración automatizadas.</li> </ul>	<ul style="list-style-type: none"> <li>• Aplicación probada y ejecutada.</li> <li>• Manual de Usuario del Software (MUS).</li> </ul>



Descripción de las técnicas:

- Técnica de depuración estática: es la revisión del código. Se le dice estático ya que se realiza sin ejecutar el programa (Bourque & Fairley, 2014).
- Técnica de depuración dinámica *PrintDebugging*: es una de las técnicas de depuración más usadas. También conocida como *Echo Debugging*. Consiste en insertar la sentencia *print* en el código para evaluar variables, sentencias en el código, flujo de sentencias. (O'Donnell & Hall, 1988).
- Técnica de depuración estática depuración por trazas (*Trace Debugging*): permite realizar trazas del programa colocando puntos de ruptura en una sentencia para evaluar el contenido de variables o registros en cada paso (Bourque & Fairley, 2014).

En la tabla 5.25 se describen las entradas y salidas de cada subproceso.

Tabla 5.25. Entradas y salidas del proceso de Pruebas.

Entrada	Subproceso	Salida
ERS Documento de Diseño (DD).	Planificación del plan de pruebas	• Documento Plan de Pruebas (DPP).
Aplicación. Casos de Pruebas (CP).	Ejecución de las pruebas	• Documento Plan de Prueba Actualizado (DPPA). • Informe de Ejecución de Pruebas. (IEP).
Aplicación • Informe de Ejecución de Pruebas. (IEP)	Corrección de errores	• Aplicación Probada y Ejecutada (APE). Manual de Usuario del Software (MUS).

### 5.3.3.7. Proceso Técnico de Lanzamiento

Los pasos o subprocesos de este proceso son: instalación de la aplicación, diseño y ejecución de pruebas de instalación, como se muestra la figura 5.10.

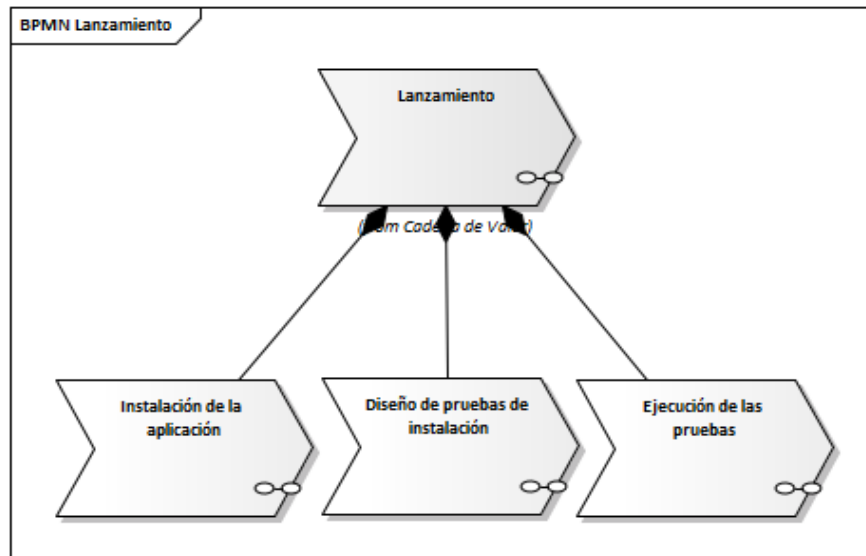


Figura 5.10. Diagrama de jerarquía del proceso de Lanzamiento.

En las tablas 5.26 y 5.27 se detallan los pasos, actividades, técnicas y/o notaciones y, productos generados en este proceso.

Tabla 5.26. Descripción del proceso de Lanzamiento (a).

Pasos /subprocesos	Actividades	Técnicas y/o notaciones	Productos
1. Instalación de la aplicación	<ul style="list-style-type: none"> <li>• Instalar la Base de Datos.</li> <li>• Instalar la aplicación.</li> <li>• Actualizar la base de datos.</li> </ul>	Técnica de migración de base de datos: importación/exportación de archivos: <i>CommaSeparatedValues</i> (CSV).	<ul style="list-style-type: none"> <li>• Base de Datos instalada y actualizada.</li> <li>• Aplicación instalada.</li> <li>• Documento de Instalación del <i>Software</i> (DIS)</li> </ul>
2. Diseño de pruebas de instalación	<ul style="list-style-type: none"> <li>• Definir los requisitos funcionales/ No funcionales a ser probados.</li> <li>• Definir los casos de pruebas de instalación.</li> </ul>	<ul style="list-style-type: none"> <li>• Técnicas de caja blanca.</li> <li>• Técnicas de caja negra.</li> </ul>	<ul style="list-style-type: none"> <li>• Especificación de Casos de Pruebas de Instalación (ECPI).</li> </ul>
3. Ejecución de las pruebas	<ul style="list-style-type: none"> <li>• Ejecutar las pruebas.</li> <li>• Corregir los errores.</li> <li>• Realizar El Formato de Entrega del <i>Software</i> (FES)</li> <li>• Entregar al Cliente</li> </ul>	<ul style="list-style-type: none"> <li>• Herramientas de apoyo especializadas para ejecutar pruebas.</li> <li>• Técnica de depuración dinámica: <i>PrintDebugging</i>.</li> </ul>	<ul style="list-style-type: none"> <li>• Informe de resultados de la Ejecución de Pruebas de Instalación (IREPI).</li> </ul>

Tabla 5.27. Descripción del proceso de Lanzamiento (b).

		<ul style="list-style-type: none"> <li>• Técnica de depuración dinámica: depuración por trazas (<i>Trace Debugging</i>)</li> <li>• Herramientas de seguimiento de errores (<i>Bug Tracking Tools</i>).</li> <li>• Técnica de pensar en alto (<i>ThinkingAloud</i>).</li> </ul>	<ul style="list-style-type: none"> <li>• Formato de Entrega del <i>Software</i> (FES).</li> <li>• Aplicación instalada y probada (AIP).</li> </ul>
--	--	--	--

Descripción de las técnicas:

- Técnica de migración de base de datos - importación/exportación de archivos *Comma Separated Values* (CSV): son archivos planos que separan los campos por un delimitador, generalmente punto y coma (;) (Leguizamón, 2017).

En la tabla 5.28 se describen las entradas y salidas de cada subprocesos.

Tabla 5.28. Entradas y salidas del proceso de Lanzamiento.

Entrada	Subproceso	Salida
<ul style="list-style-type: none"> <li>• Aplicación Probada y Ejecutada (APE).</li> <li>• Manual de Usuario del <i>software</i> (MUS).</li> </ul>	Instalación de la aplicación	<ul style="list-style-type: none"> <li>• Base de Datos instalada y actualizada.</li> <li>• Aplicación instalada.</li> <li>• Documento de Instalación del <i>Software</i> (DIS)</li> </ul>
<ul style="list-style-type: none"> <li>• Aplicación instalada.</li> </ul>	Diseño de pruebas de instalación	<ul style="list-style-type: none"> <li>• Especificación de Casos de Pruebas de Instalación (ECPI).</li> </ul>
<ul style="list-style-type: none"> <li>• Especificación de Casos de Pruebas de Instalación (ECPI)</li> </ul>	Ejecución de pruebas	<ul style="list-style-type: none"> <li>• Informe de resultados de la Ejecución de Pruebas de Instalación (IREPI).</li> <li>• Aplicación instalada y probada (AIP).</li> </ul>

### 5.3.4. Modelo de Productos

Este modelo describe, identifica y clasifica los diferentes tipos de productos como documentos, modelos, diagramas, formatos textuales, minutas, listas de productos, prototipos, informes o código que se generan durante la ejecución de los procesos para el desarrollo de la aplicación.

Para el equipo de desarrollo representa el contenido de los productos que cada actor debería producir según sus funciones en la ejecución de los procesos técnicos, de gestión y de soporte. La figura 5.11 muestra los productos del método MASIWeb, los cuales se clasifican de acuerdo a los grupos de procesos donde ellos se crean.

Los productos intermedios son todos aquellos documentos, librerías, matrices, entre otros, que son usados como insumo en la producción de otros productos terminados; se elaboran durante la ejecución de los procesos técnicos, de soporte y de gestión y son necesarios para desarrollar la aplicación (Montilva, Barrios & Rivero, 2008).

Los Productos finales son todos aquellos que constituyen parte integrante de la aplicación y se entregan al usuario una vez que el proceso de desarrollo de la aplicación finaliza (Montilva, Barrios & Rivero, 2008).

El método MASIWeb los define como Productos Finales Entregables (PFE) y Productos Intermedios Entregables (PIE).

Los productos pueden clasificarse por importancia en esenciales, opcionales y recomendables. A continuación, se describen cuáles productos están en cada clasificación:

- Esenciales: son los productos que se desarrollan en semestres o trayectos donde se imparte la materia Ingeniería de *Software* y desarrollan una aplicación de mediana complejidad. Los productos a generar son: documento visión del proyecto, plan del proyecto, documento de especificación de requisitos de *software*, documento de diseño, documento plan de pruebas y, manual de usuario del sistema.

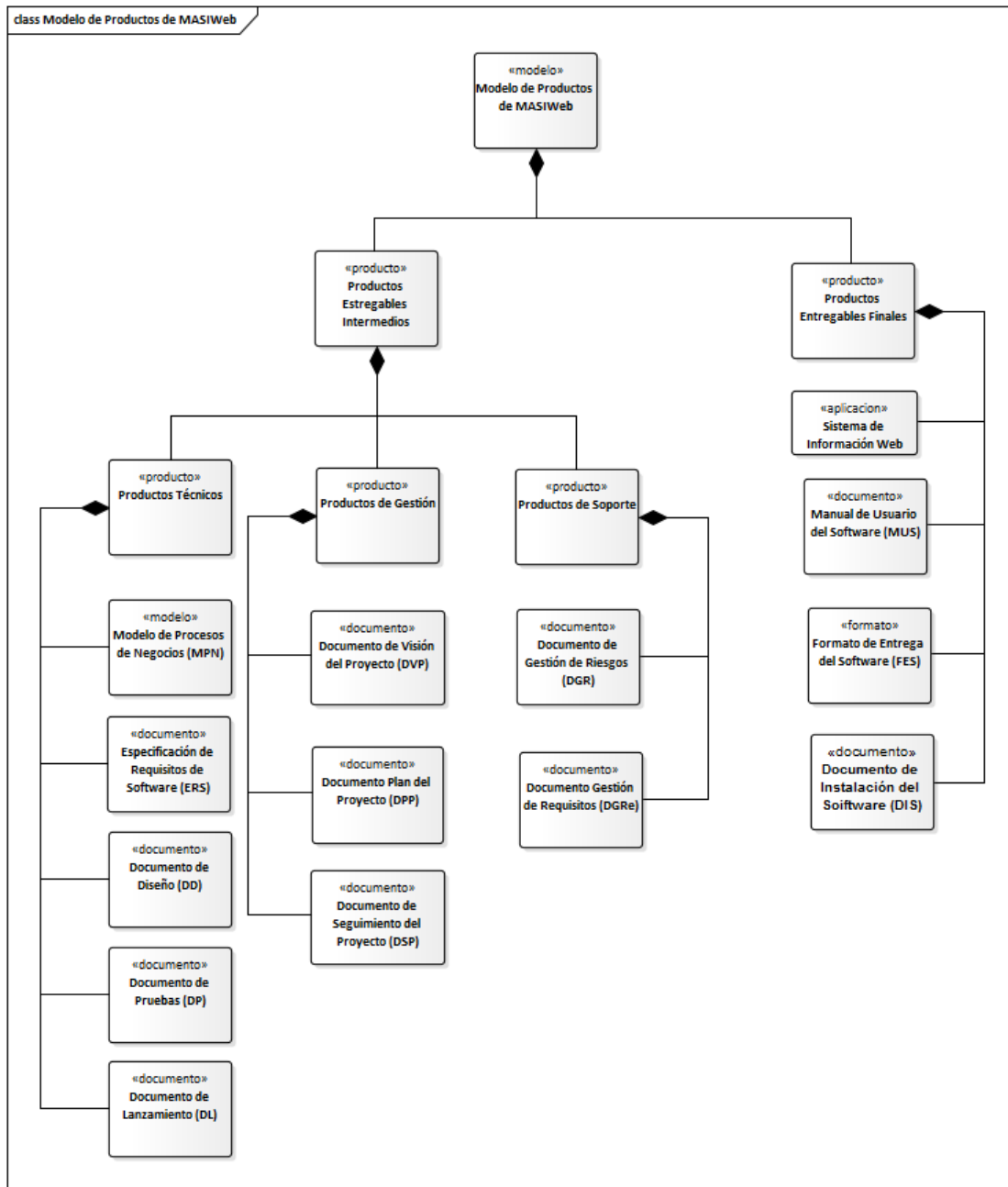


Figura 5.11. Modelo de Productos del método MASIWeb.

- Opcionales: a diferencia de los anteriores, estos productos se desarrollan en semestres o trayectos más avanzados, en los cuales los estudiantes ya han cursado y tienen conocimiento de Ingeniería de *Software* y,

participan en la administración de gestión de proyectos informáticos. Los productos a desarrollar son: Modelo de Proceso de Negocio, Documento de Gestión de Riesgo y Gestión de Requisitos.

- **Recomendables:** todos los productos. Escenario donde los estudiantes tienen conocimiento en las dos materias antes mencionada: Ingeniería de *Software* y Administración de Gestión de Proyectos.

A continuación, se detallan los productos del método representados mediante diagrama de clases de la figura 5.11.

#### 5.3.4.1. Modelo de Procesos de Negocios (MPN)

Es una representación abstracta (gráfica) que describe las actividades de una organización y, cómo se relacionan e interactúan con los recursos del negocio para lograr la meta establecida para el proceso (*Object Management Group [OMG], 2008*). Este modelo se realiza a través de notación UML *Business* (Eriksson & Penker, 2000) y el método *Business Modeling* (Montilva & Barrios, 2004) (ver figura 5.12).

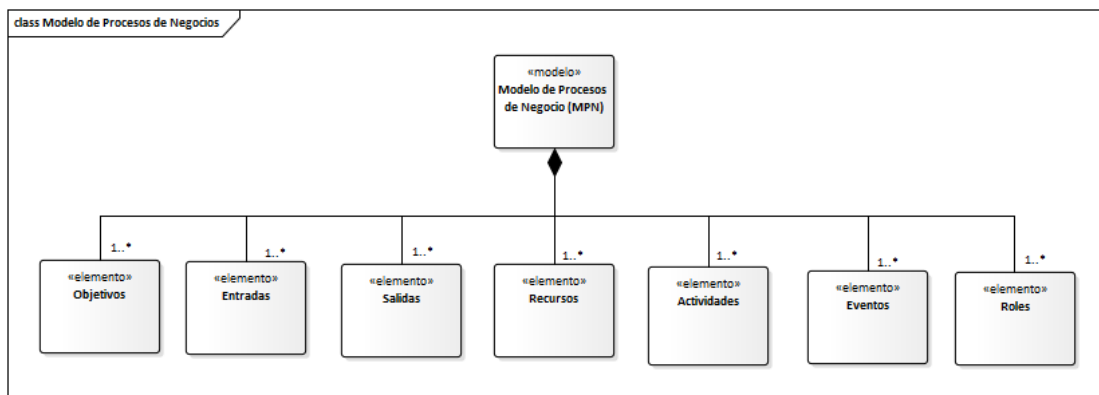


Figura 5.12. Modelo de Procesos de Negocios (MPN).

### 5.3.4.2. Documento Especificación de Requisitos de Software (ERS)

El objetivo es identificar, especificar y documentar cada requisito funcional y no funcional de la aplicación (ver figura 5.13).

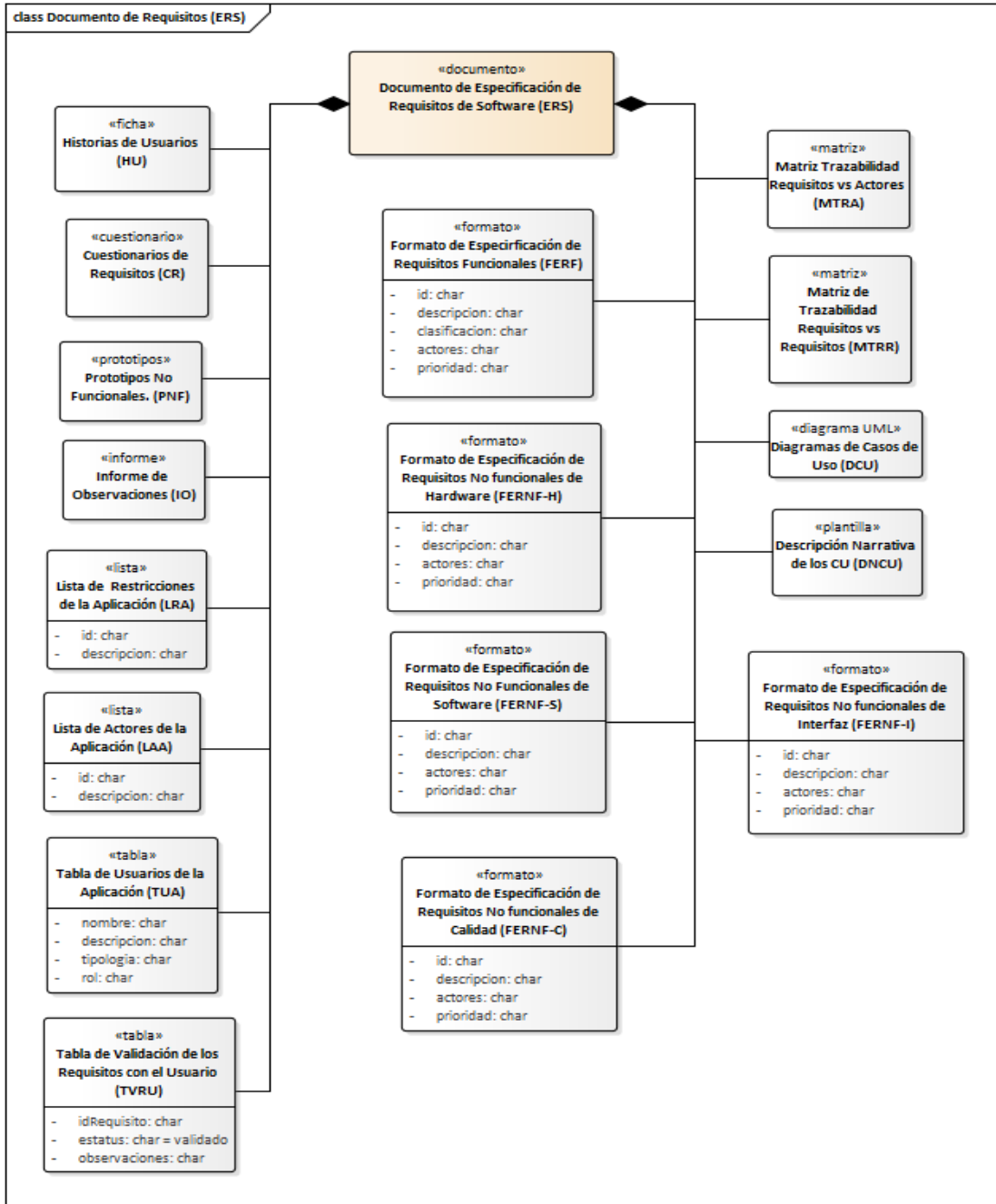


Figura 5.13. Documento de Especificación de Requisitos de Software (ERS).

Los productos se elaboran en cada uno de los subprocesos o pasos del proceso Análisis y definición de requisitos (ver tabla 5.29).

Tabla 5.29. *Productos Intermedios Entregables del documento ERS.*

Subproceso	Producto
Descubrimiento de los requisitos	<ol style="list-style-type: none"> <li>1. Historias de Usuarios (HU).</li> <li>2. Prototipos No Funcionales (PNF).</li> <li>3. Informe de Observación (IO)</li> <li>4. Cuestionarios de Requisitos (CR).</li> <li>5. Lista de las restricciones de la Aplicación (LRA).</li> <li>6. Lista de Actores de la Aplicación (LAA).</li> </ol>
Análisis de los requisitos	<ol style="list-style-type: none"> <li>7. Formato de Especificación de Requisitos Funcional (FCRF)</li> <li>8. Formato de Especificación de Requisitos No Funcional de <i>Hardware</i> (FERNF-H)</li> <li>9. Formato de Especificación de Requisitos No Funcional de <i>Software</i> (FERNF-S).</li> <li>10. Formato de Especificación de Requisitos No Funcional de Interfaz (FERNF-I).</li> <li>11. Formato de Especificación de Requisitos No Funcional de Calidad (FERNF-C).</li> <li>12. Tabla de los Usuarios de la Aplicación (TUA).</li> <li>13. Tabla de Validación de los Requisitos con el Usuario (TVU).</li> </ol>
Especificación de los requisitos	<ol style="list-style-type: none"> <li>14. Matriz de Trazabilidad requisitos vs Actores (MTRA).</li> <li>15. Matriz de Trazabilidad de requisitos vs requisitos (MTRR)</li> <li>16. Diagramas de Casos de Uso (DCU).</li> <li>17. Descripción narrativa de los CU (DNCU).</li> </ol>

#### 5.3.4.3. Documento de Diseño (DD)

El objetivo es detallar la arquitectura de la aplicación, sus componentes de procedimientos, datos e interfaz, los cuales están relacionados. Los productos contenidos en este documento son: diseño arquitectónico, diseño de la arquitectura de contenido, diseño de componentes, diseño de interfaz, diseño de la base de datos, diseño de navegación de la aplicación y, diseño de navegación de los usuarios (ver tabla 5.30).



Tabla 5.30. *Productos Intermedios Entregables del Documento de Diseño.*

Paso /Subproceso	Producto
Definición de la Arquitectura de la aplicación Web	<ul style="list-style-type: none"> <li>• Documento Diseño de la Arquitectura Web (DAW) que contiene:               <ol style="list-style-type: none"> <li>1. Arquitectura de la aplicación web. (AAW)</li> <li>2. Diagrama de los Módulos (DM).</li> <li>3. Diagrama de Componentes. (DDC)</li> <li>4. Arquitectura del Contenido Web (ACW).</li> </ol> </li> </ul>
Elaboración del Modelo Estático	<ol style="list-style-type: none"> <li>5. Modelo Estático (ME).</li> </ol>
Elaboración del Modelo Dinámico	<ol style="list-style-type: none"> <li>6. Modelo Dinámico (MD).</li> </ol>
Construcción del Modelo de Datos	<ul style="list-style-type: none"> <li>• Documento de Diseño de la Base de Datos (DBD) que contiene:               <ol style="list-style-type: none"> <li>7. Modelo Entidad Relación de la Base de Datos (ER-BD).</li> <li>8. Diseño Lógico de la Base de Datos (DL-BD).</li> <li>9. Diseño Físico de la Base de Datos (DF-BD).</li> </ol> </li> <li>• Documento de Diseño de la Base de Datos (DBD) validado por el PE (DBD-VPE)</li> </ul>
Diseño de la interfaz Web (IW)	<ul style="list-style-type: none"> <li>• Documento de la Interfaz Web (DIW) que contiene:               <ol style="list-style-type: none"> <li>10. Lista de Sitios Similares (LSS).</li> <li>11. Tarjetas Etiquetadas Validadas por el Usuario (TEVU).</li> <li>12. Prototipos de la Interfaz Web (PIW).</li> <li>13. Prototipos de la Interfaz Web Aprobados por el Usuario (PIWAU).</li> <li>14. Reporte de la Evaluación Heurística de la facilidad de uso (<i>Usability</i>). (REHU).</li> <li>15. Informe de los cambios de la Interfaz Web (ICIW).</li> </ol> </li> </ul>
Diseño del Modelo de Navegación	<ol style="list-style-type: none"> <li>16. Modelo de Navegación de la Aplicación (MNA).</li> <li>17. Modelo de Navegación de usuarios (MNU).</li> </ol>

Los productos están agrupados en tres documentos: el Documento de la Arquitectura Web (DAW), el Documento del Diseño de la Base de Datos (DDBD) y, el Documento del Diseño de la Interfaz Web (DDIW) (ver figura 5.14).

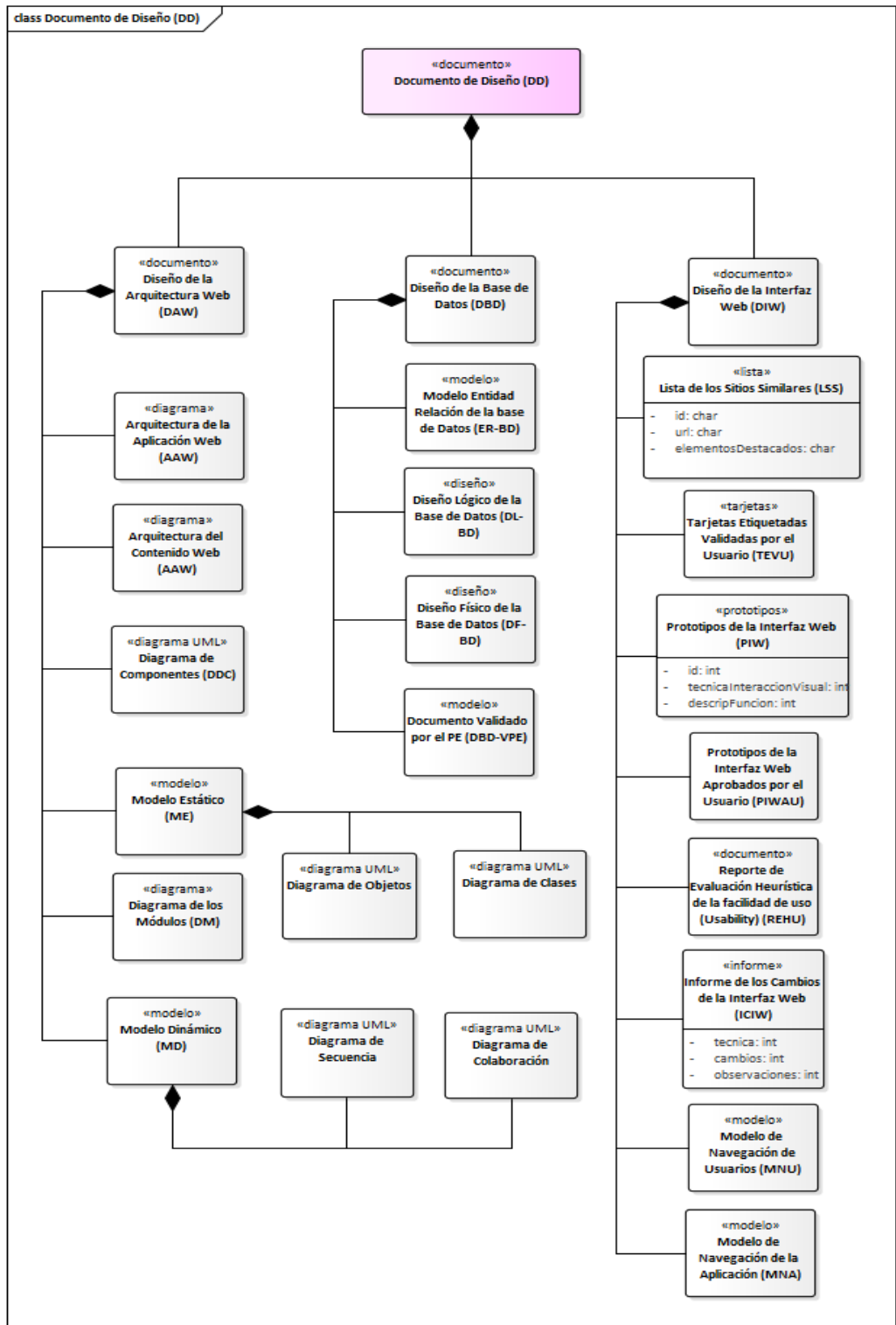


Figura 5.14. Documento de Diseño (DD).

El diseño de la Arquitectura del Contenido según Powell (2000) citado por Pressman (2005, p. 586), es una estructura donde se especifican las páginas y éstas, se enlazan a través de los hipervínculos, es la hipermedia global de la Aplicación Web. La clasificación es lineal, reticular, jerárquica y en red.

#### 5.3.4.4. Documento de Pruebas (DP)

El propósito de este documento es definir las técnicas de pruebas, los lineamientos y la documentación de la ejecución las pruebas, los requisitos a ser probados y, el informe de errores (ver figura 5.15).

Los productos se elaboran en cada uno de los subprocesos o pasos del proceso de Pruebas (ver tablas 5.31 y 5.32).

Tabla 5.31. *Productos Intermedios Entregables del Documento de Pruebas.*

<b>Paso/Subproceso</b>	<b>Producto</b>
Planificación de las pruebas	<ul style="list-style-type: none"> <li>• Documento Plan de Pruebas (DPP) que contiene:               <ol style="list-style-type: none"> <li>1. Técnicas de Pruebas.</li> <li>2. Tipos de Pruebas.</li> <li>3. Cronograma de Pruebas.</li> <li>4. Casos de Pruebas (CP)</li> <li>5. Lista de Requisitos a Probar (LRP).</li> </ol> </li> </ul>
Ejecución de las pruebas	<ul style="list-style-type: none"> <li>• Informe de Ejecución de Pruebas. (IEP).</li> <li>• Casos de Pruebas ejecutados (CPE).</li> <li>• Informe de Errores (IE).</li> <li>• Documento Plan de Prueba Actualizado (DPPA).</li> </ul>

Tabla 5.32. *Productos Finales Entregables del proceso de Pruebas.*

<b>Paso/Subproceso</b>	<b>Producto</b>
Corrección de errores	<ul style="list-style-type: none"> <li>• Aplicación probada y ejecutada.</li> <li>• Manual de Usuario del <i>Software</i> (MUS).</li> </ul>

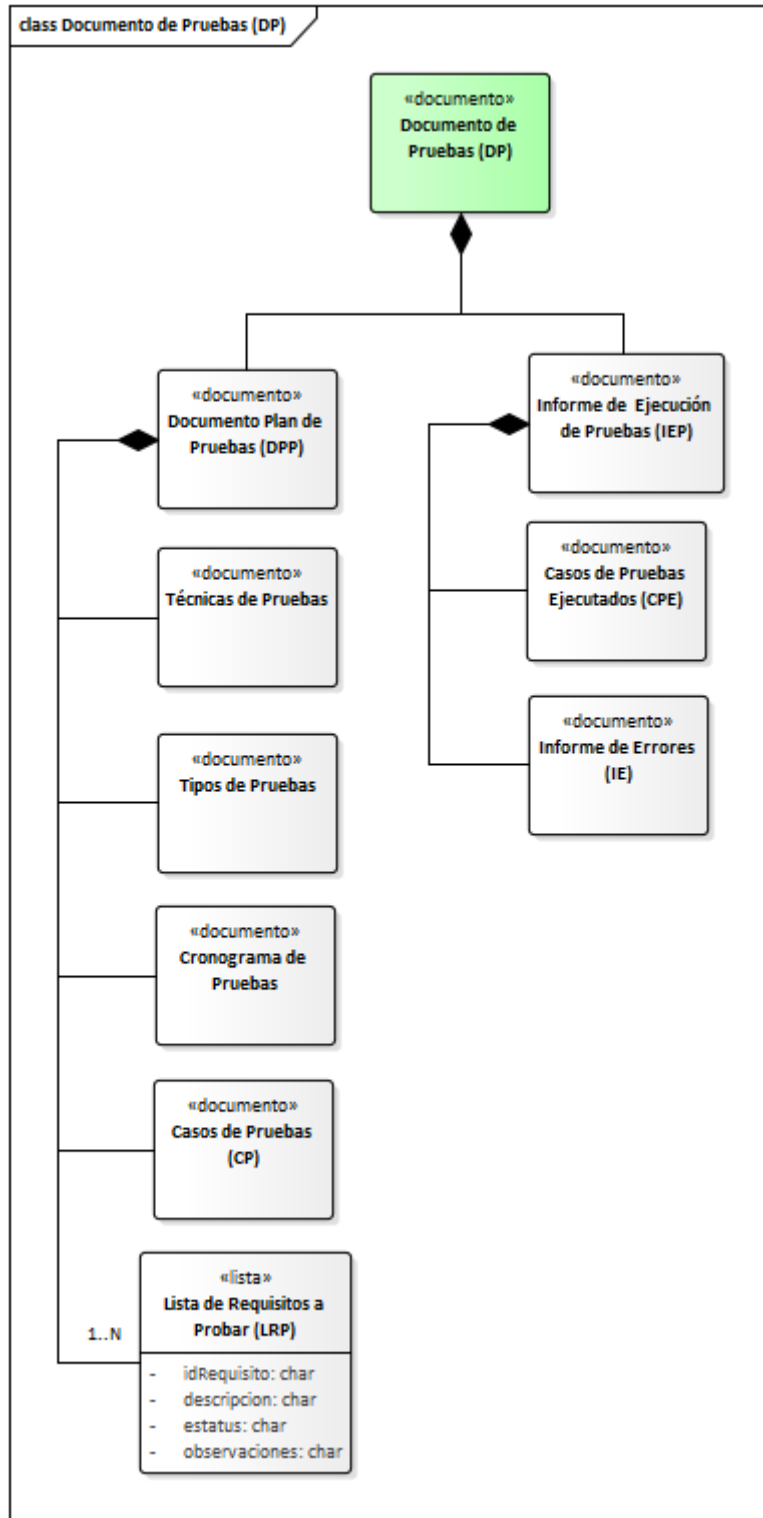


Figura 5.15. Documento de Pruebas (DP).

#### 5.3.4.5. Documento Manual de Usuario del Software (MUS)

Producto final entregable el cual contiene la ayuda en línea de los procedimientos a ejecutarse de cada una de las funciones de la aplicación.

#### 5.3.4.6. Documento de Lanzamiento (DL)

Este documento especifica las pruebas con sus respectivos casos de pruebas e informes de ejecución (ver figura 5.16).

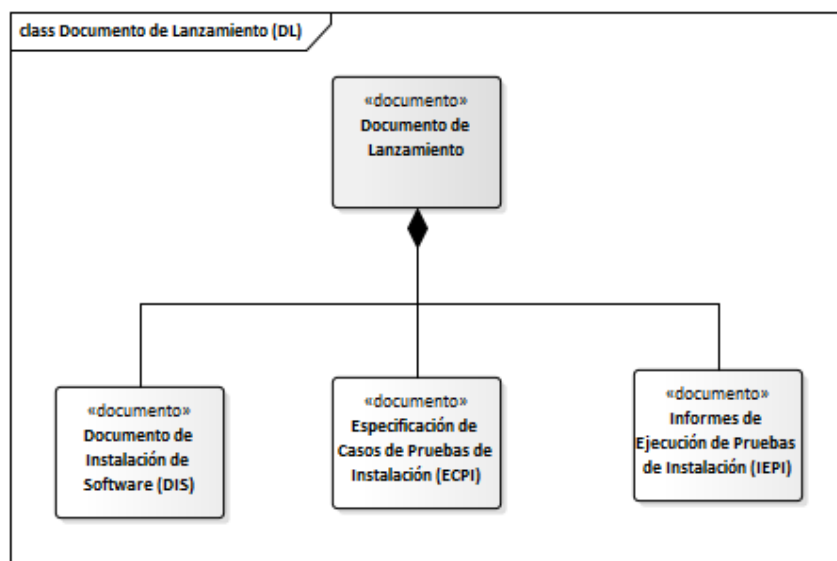


Figura 5.16. Documento de Lanzamiento (DL).

A continuación, se visualizan los productos intermedios y finales en cada uno de los subprocesos del proceso de Lanzamiento (ver tablas 5.33 y 5.34).

Tabla 5.33. *Productos Intermedios Entregables del documento de Lanzamiento.*

<b>Paso/Subproceso</b>	<b>Producto</b>
Instalación de la aplicación.	<ul style="list-style-type: none"> <li>• Documento de Instalación del <i>Software</i> (DIS)</li> </ul>
Diseño de pruebas de instalación.	<ul style="list-style-type: none"> <li>• Especificación de Casos de Pruebas de Instalación (ECPI).</li> </ul>
Ejecución de pruebas de instalación	<ul style="list-style-type: none"> <li>• Informe de resultados de la Ejecución de Pruebas de Instalación (IREPI).</li> </ul>

Tabla 5.34. *Productos Finales Entregables del documento de Lanzamiento.*

<b>Paso/Subproceso</b>	<b>Producto</b>
Instalación de la aplicación	<ul style="list-style-type: none"> <li>• Base de Datos instalada y actualizada.</li> <li>• Aplicación instalada.</li> </ul>
Ejecución de pruebas	<ul style="list-style-type: none"> <li>• Aplicación instalada y probada (AIP).</li> </ul>

#### **5.3.4.7. Documento de Instalación del *Software* (DIS)**

Contiene el procedimiento necesario para instalar el SIW en el servidor Web.

#### **5.3.4.8. Formato de Entrega del *Software* (FES)**

El SIW en conjunto con el Documento de Instalación del *Software* (DIS), y, el escrito de la documentación del Proyecto se entregará, en el medio físico de Disco Compacto (CD).

#### **5.3.4.9. Documento Visión del Proyecto (DVP)**

Es el documento inicial de todo proyecto, en el cual se especifica el problema, la solución y la lista inicial de los requisitos. A continuación, se visualizan los productos generados en este documento, a través de un diagrama de clases (ver figura 5.17).

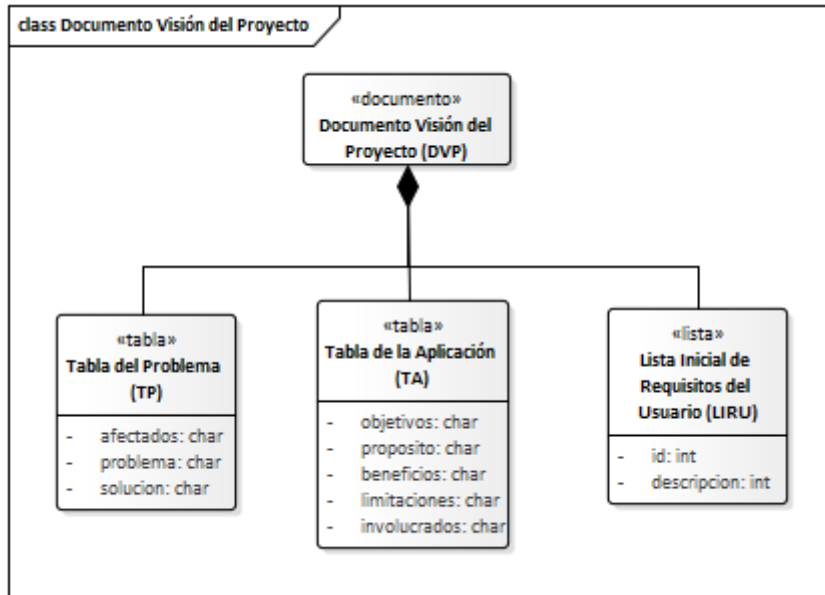


Figura 5.17. Documento Visión del Proyecto (DVP).

#### 5.3.4.10. Documento Plan del Proyecto (DPP)

El propósito de éste documento es describir el cronograma de las actividades de desarrollo y el de iteraciones, el modelo de actores y de productos (ver figura 5.18).

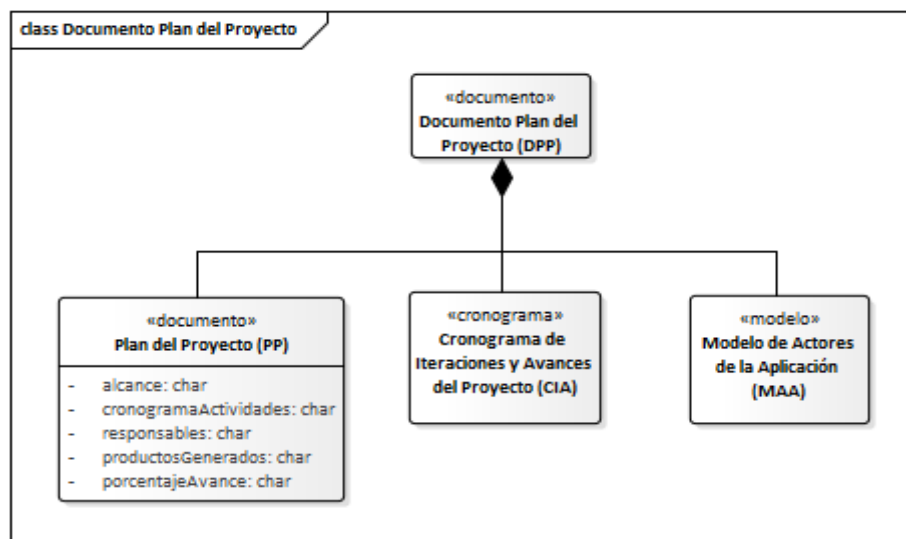


Figura 5.18. Documento Plan del Proyecto (DPP).

#### 5.3.4.11. Documento de Seguimiento del Proyecto (DSP)

El propósito de éste documento es almacenar las minutas de los avances e iteraciones programadas en el Documento Plan del proyecto (DPP) (ver figura 5.19).

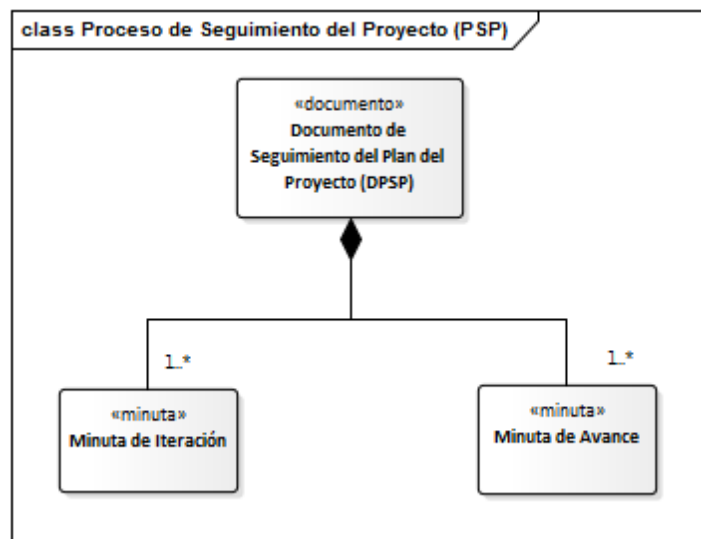


Figura 5.19.Documento Seguimiento del Proyecto (DSP).

#### 5.3.4.12. Documento de Gestión de Riesgos (DGR)

Es el documento que identifica los posibles riesgos del proyecto. Por cada riesgo se debe proporcionar la siguiente información: una descripción del riesgo, el grado de magnitud o *ranking* (1..10), el impacto que pudiera ocasionar(alto, bajo), los indicadores asociados al riesgo, la(s) estrategia(s) de mitigación para aminorar el impacto en el proyecto, el plan de contingencia en caso que el riesgo sea una realidad y el seguimiento de la ejecución de los planes acordados (ver figura 5.20).

Este documento se elabora en paralelo al Plan del Proyecto (PDD).



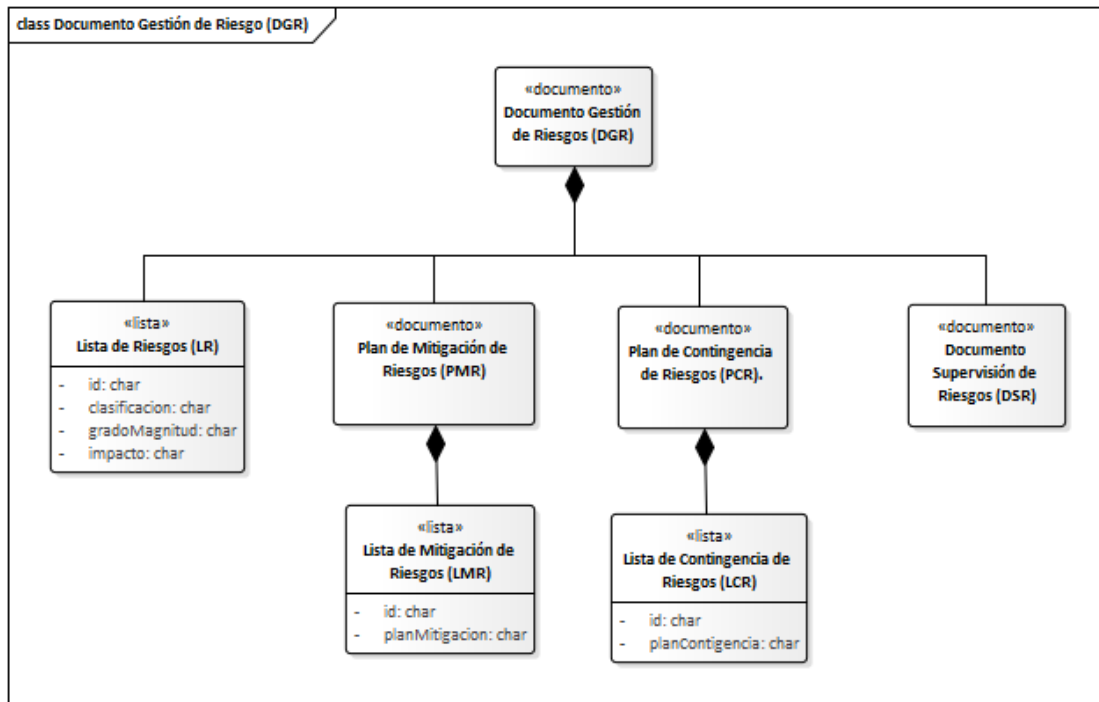


Figura 5.20. Documento Gestión de Riesgos (DGR).

### 5.3.5. Modelo de Actores

Este modelo define los actores interesados (*stakeholders*) involucrados en el desarrollo de la aplicación, sus roles y sus respectivas responsabilidades.

El grupo de trabajo es pequeño (máximo 3 participantes). Los roles del desarrollador son intercambiables; ejemplo, el líder el analista puede ejercer los roles de programador y diseñador. Los roles programador y probador no se deberían ejecutarse por un mismo actor. Se recomienda que el rol diseñador de base de datos tenga conocimientos sólidos o ser especialista en base de datos.

A continuación, se visualiza el modelo de Actores, la cantidad de profesores evaluadores del proyecto pueden ser 1 o más (\*) (ver figura 5.21).

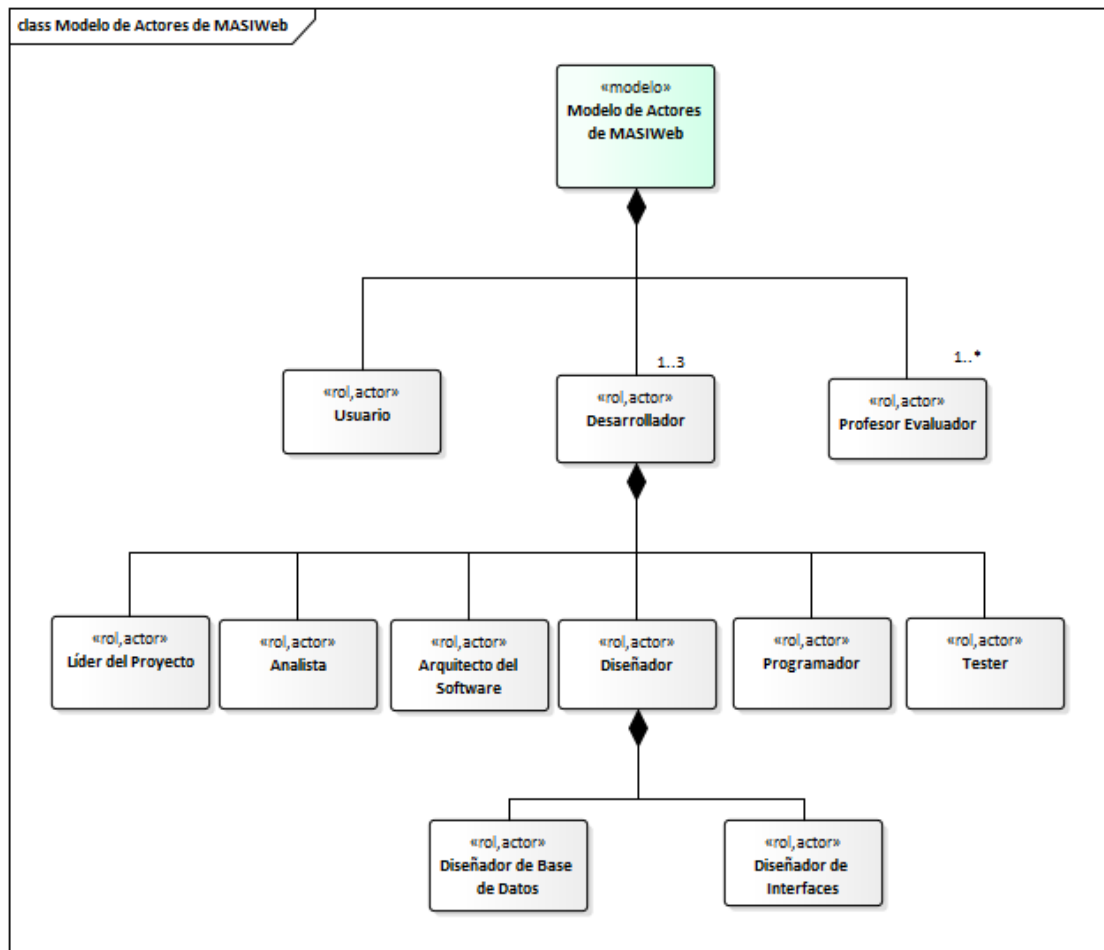


Figura 5.21. Modelo de Actores del método MASIWeb.

Los roles y responsabilidades de los actores se describen en la tabla 5.35.

Tabla 5.35. Roles y responsabilidades de los actores.

Rol	Responsabilidades
Profesor Evaluador	<ul style="list-style-type: none"> <li>• Verificar y Validar los procesos, los productos y la ejecución exitosa de la aplicación.</li> <li>• Coordinar en conjunto con el grupo de desarrolladores las sesiones de revisión e iteraciones.</li> <li>• Especificar mejoras de los procesos, productos y la aplicación en las iteraciones programadas, documentando los cambios en las Minutas de Iteración (MI).</li> <li>• Hacer seguimiento de las mejoras indicadas en las minutas.</li> <li>• Velar que los participantes empleen apropiadamente los modelos de procesos, productos y actores instanciados a partir del método MASIWeb.</li> <li>• Guiar al equipo de desarrollo en el uso del método MASIWeb y las diferentes técnicas, herramientas, notaciones, prácticas y estándares especificados en el modelo de procesos.</li> </ul>
Líder del Proyecto	<ul style="list-style-type: none"> <li>• Desarrollar el objetivo, los beneficios, las limitaciones y los involucrados del proyecto en el documento Visión.</li> <li>• Describir el Plan del Proyecto, el tiempo estimado, el equipo de desarrollo y las actividades en cada proceso.</li> <li>• Definir los riesgos y la gestión de los mismos.</li> </ul>
Analista	<ul style="list-style-type: none"> <li>• Entrevistarse con el cliente.</li> <li>• Definir los requisitos del proyecto.</li> <li>• Discutir con el cliente las funcionalidades del proyecto (requisitos funcionales y las restricciones de los mismos (requisitos no funcionales)).</li> </ul>
Arquitecto del Software	<ul style="list-style-type: none"> <li>• Definir el diseño de la arquitectura del proyecto.</li> <li>• Especificar las distintas vistas de la arquitectura.</li> </ul>
Diseñador de Interfaz	<ul style="list-style-type: none"> <li>• Diseñar la interfaz de Usuario.</li> <li>• Diseñar la arquitectura del contenido.</li> <li>• Modelar los diagramas UML: diagrama de clase, de secuencia, de colaboración y de despliegue.</li> </ul>
Diseñador de Bases de Datos	<ul style="list-style-type: none"> <li>• Diseñar el Modelo de Datos.</li> <li>• Modelar los diagramas UML: diagrama de Clases</li> </ul>
Programador	<ul style="list-style-type: none"> <li>• Construir la base de Datos.</li> <li>• Codificar la aplicación.</li> <li>• Realizar las diferentes pruebas.</li> <li>• Corregir los errores.</li> <li>• Integrar los módulos.</li> </ul>
Probador	<ul style="list-style-type: none"> <li>• Elaboración del Plan de Pruebas.</li> <li>• Ejecución de las pruebas funcionales, no funcionales., unitarias, de integración, de aceptación.</li> <li>• Validaciones con el Usuario.</li> </ul>

### 5.3.6. Glosario de términos

A continuación, se describe un conjunto de definiciones esenciales para familiarizarse y, aplicar, el método MASIWeb de desarrollo de aplicaciones Web basado en Ingeniería de Métodos (ver tablas 5.36 y 5.37).

Tabla 5.36. *Glosario de Términos (a).*

<b>Concepto</b>	<b>Definición</b>
Arquitectura del <i>Software</i>	Es la estructura o las estructuras del sistema, que contiene componentes de <i>software</i> , las propiedades y sus relaciones.
Casos de Pruebas	Plantilla en la que se definen los parámetros o argumentos de entrada y/o salida para probar un componente, parte de un código o el <i>software</i> en su totalidad.
Depuración	Proceso de ejecutar pruebas paso a paso.
Diagramas de Caso de Uso	Modelan la funcionalidad del sistema usando actores, casos de usos y relaciones.
Diagramas de Colaboración	Muestra la organización estructural de objetos que interactúan unos con otros.
Diagramas de Clases	Describe los objetos, sus propiedades y comportamientos.
Diagramas de Componentes	Define la organización y dependencia de los componentes de un sistema.
Diagramas de Objetos	Contiene un conjunto de instancias (objetos) de los elementos encontrados en los diagramas de clase.
Diagramas de Secuencia	Representa la secuencia de los mensajes intercambiados por los objetos que participan en un escenario enfatizando el orden.
Flujo de Eventos de los Casos de Usos	Proporciona detalles textuales de un caso de uso. Los flujos básicos y alternativos.
Ingeniería de Métodos	Es una disciplina de diseño, construcción y adopción de métodos y herramientas para el desarrollo de Sistemas de <i>Software</i> .
ISO 9126	Es un estándar internacional para la evaluación de la calidad del <i>software</i> .
Matriz de Trazabilidad	Es una tabla que vincula dos entidades definidas en el plano vertical y horizontal.
Modelo de Actores	Identifica los roles requeridos para desarrollar una aplicación y recomienda una estructura organizacional para el equipo de desarrollo.
Modelo	Un modelo permite visualizar, especificar, construir y documentar un proceso a través de una notación definida.
Modelo Dinámico	Se denomina dinámico porque muestra el comportamiento de los objetos en un cierto momento.
Modelo Estático	Se denomina estático porque muestra todas las relaciones a lo largo del tiempo, no a las que corresponde en un cierto momento.
Modelo de Navegación	Representa los nodos o enlaces de navegación unidos por los vínculos de navegación. Es la estructura de navegación entre las páginas, menús, formularios o cualquier elemento Web a través de los enlaces.

Tabla 5.37. Glosario de Términos (b).

Concepto	Definición
Modelo de Procesos de Negocios	Es una representación gráfica que describe los procesos de negocios de una empresa u organización con el objeto de que puedan ser analizados y mejorados.
Modelo de Productos	Describe los productos que se elaboran mediante la aplicación del método.
Modelo de Procesos	Especifica el conjunto de acciones requeridas que los actores deben ejecutar para desarrollar un producto.
Procesos Técnicos	Se relacionan con las actividades de análisis, diseño, implementación y pruebas de las aplicaciones.
Procesos de Gestión	Se encargan de gestionar el desarrollo de cada aplicación como un proyecto de ingeniería; involucran, por lo tanto, actividades de planificación, organización, administración, dirección y control del proyecto.
Procesos de Gestión	Se encargan de gestionar el desarrollo de cada aplicación como un proyecto de ingeniería; involucran, por lo tanto, actividades de planificación, organización, administración, dirección y control del proyecto.
Procesos de Soporte	Complementan los procesos técnicos y gerenciales con actividades, tales como: el aseguramiento de la calidad, la gestión de la configuración, la capacitación de los actores y la gestión de riesgos del proyecto.
Proceso iterativo	Proceso de ciclo corto de construcción repetitivo. Al final de cada iteración se le entrega al cliente una versión mejorada o con mayores funcionalidades.
Prototipo	Construcción de una maqueta o modelo de un sistema de <i>software</i> o de una parte del mismo.
Requisito	Un requisito es algo que el sistema debe ser capaz de hacer (o una restricción que debe cumplir) para que pueda cumplir su propósito y satisfacer a sus usuarios.
Requisito funcional	Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares.
Requisito no funcional	Son las restricciones de los servicios ofrecidas por el sistema; se refiere a las propiedades emergentes y no a las propiedades del sistema. Definiendo propiedades emergentes aquellas que se deben cumplir como un todo y que es difícil atribuírselo a un componente en particular; ejemplo la seguridad.
Sistema de Información Web (SIW)	Sistemas de Información tradicionales que emplean las tecnologías de internet y las capacidades de un navegador para integrarse con diversos sistemas heterogéneos, tales como <i>e-Business</i> , <i>Groupware</i> , Sistemas ERP entre otros.
Técnica de prueba Caja Blanca	Es una técnica para realizar pruebas a nivel de componentes, se basa en el conocimiento de la estructura y de las sentencias del programa y requiere de un conocimiento exhaustivo del código.
Técnica de prueba Caja Negra	Pruebas de Caja Negra llamada también pruebas de comportamiento o enfoque funcional, se centra en los requisitos y no en el código fuente.
UML	<i>Unified Modeling Language</i> : Lenguaje de Modelado de sistemas y <i>software</i> , estandarizado por el consorcio <i>Object Management Group</i> (OMG).

## 5.4. Directivas para la instanciación

La instanciación del método MASIWeb es una actividad metodológica donde el equipo de desarrollo usa el método para determinar el equipo de trabajo, los procesos y los productos necesarios para el desarrollo del SIW (ver figura 5.22).

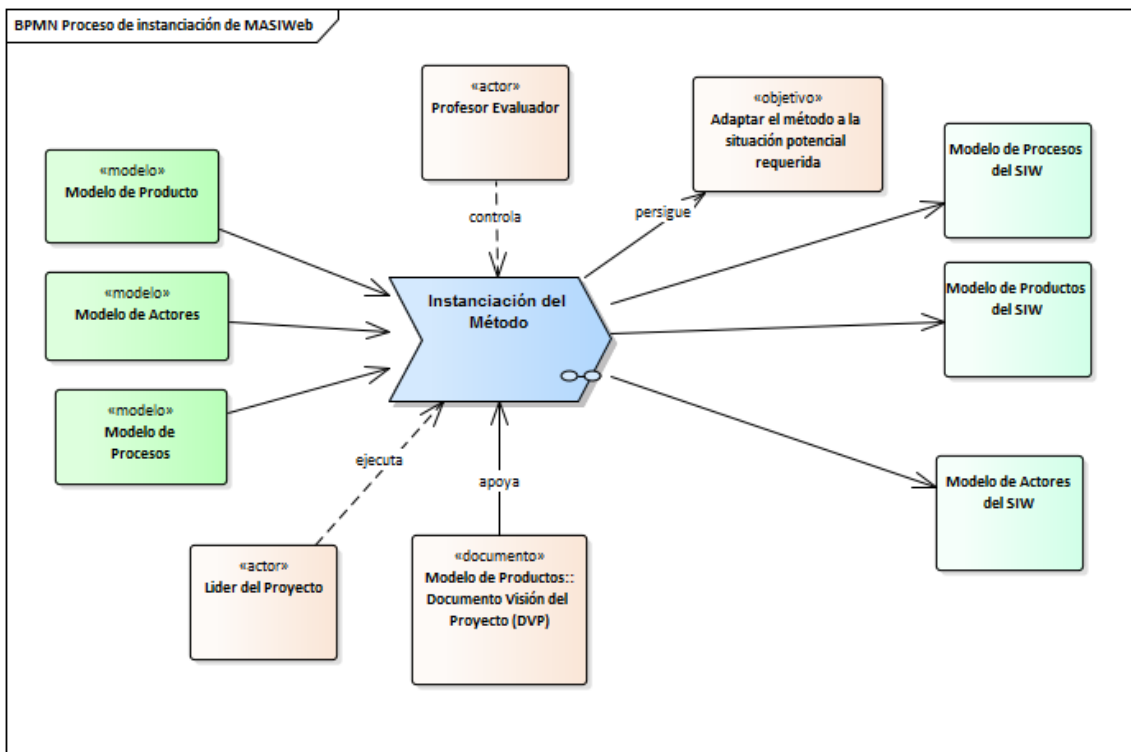


Figura 5.22. Proceso de instanciación del método MASIWeb.

### 5.4.1. Situaciones potenciales del método.

En este contexto se describen dos (2) situaciones particulares o potenciales de instanciación:

1. En desarrollos de *software* donde se ejecuten procesos iniciales o esenciales de planificación de proyecto en conjunto con los procesos técnicos; escenario que se denominará Situación Potencial de Procesos Técnicos (SPPT) (ver figura 5.23).

Los procesos del diseño de la interfaz Web y el desarrollo de los módulos son cíclicos e iterativo ya que deben ser constantemente evaluados por el cliente para corregir problemas de:

- Facilidad de uso de la interfaz.
- Errores en el proceso de pruebas de unidad y de integración.

2. En proyectos que amerite la implementación de procesos de administración de proyectos, proceso en el cual se generan los productos esenciales y productos recomendables, escenario que se denominará Situación Potencial de Administración de Proyectos (SPAP) (ver figura 5.24). A diferencia de la situación anterior, se diseña el Modelo de Procesos de Negocios (MPN) y, se ejecutan los procesos de gestión, de requisitos y de riesgos en toda la ejecución del ciclo de desarrollo del *software*.

#### **5.4.2. Flujos de trabajo de las situaciones potenciales**

A continuación, se visualiza el flujo de trabajo de las dos situaciones potenciales.

Para una mayor comprensión se muestran dos figuras correspondientes a las actividades de cada situación potencial (ver figuras 5.25 y 5.26). Y dos figuras que describen las actividades comunes que se ejecutan para continuar con el proceso del ciclo de desarrollo de *software* (ver figuras 5.27 y 5.28).

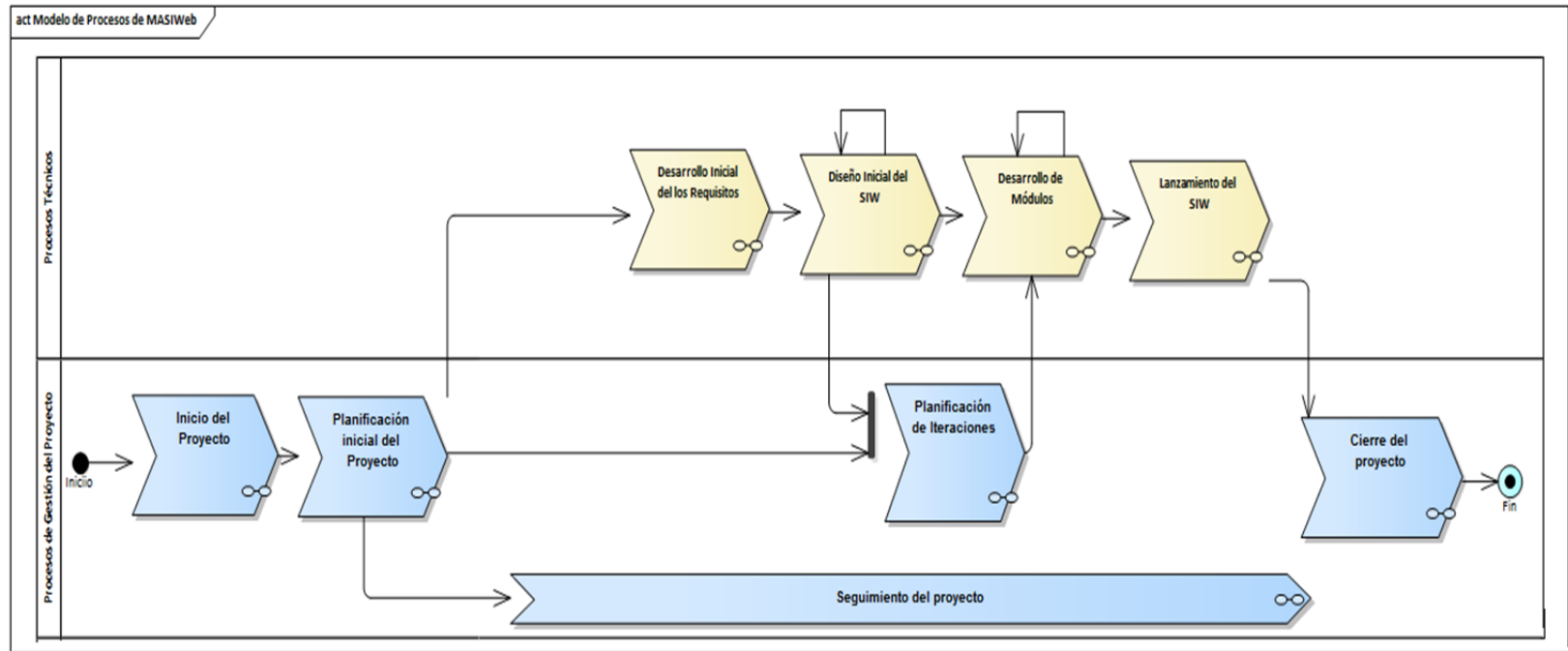


Figura 5.23. Modelo principal de situación potencial de procesos técnicos.



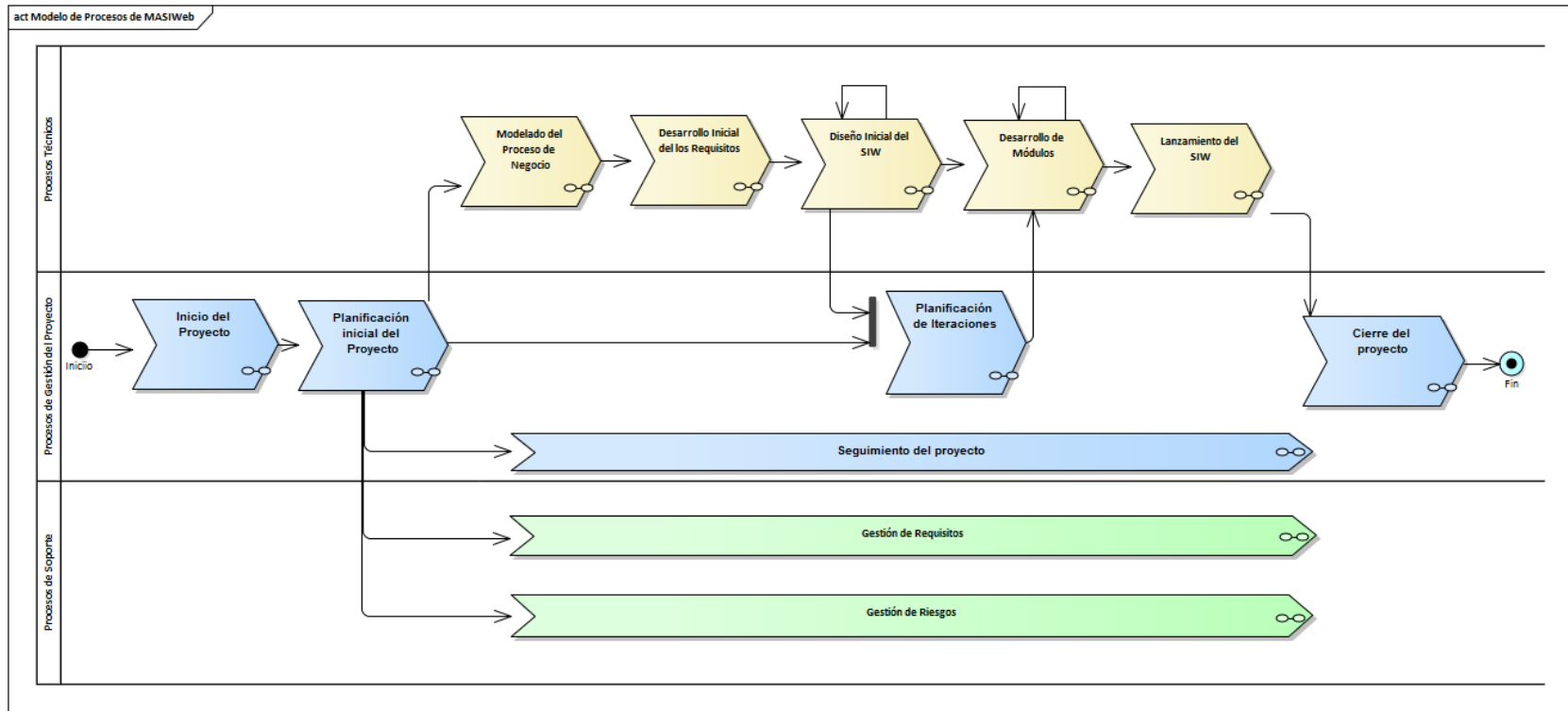


Figura 5.24. Modelo principal de situación potencial de administración de proyectos.

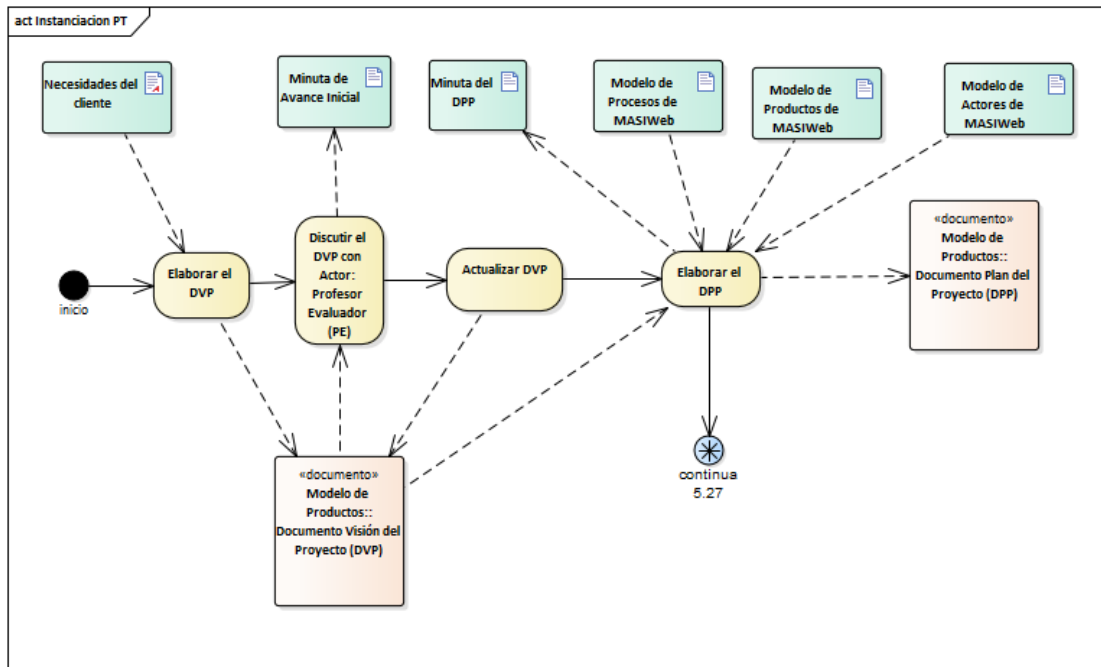


Figura 5.25. Diagrama de flujo de Situaci3n Potencial de Procesos T3cnicos.

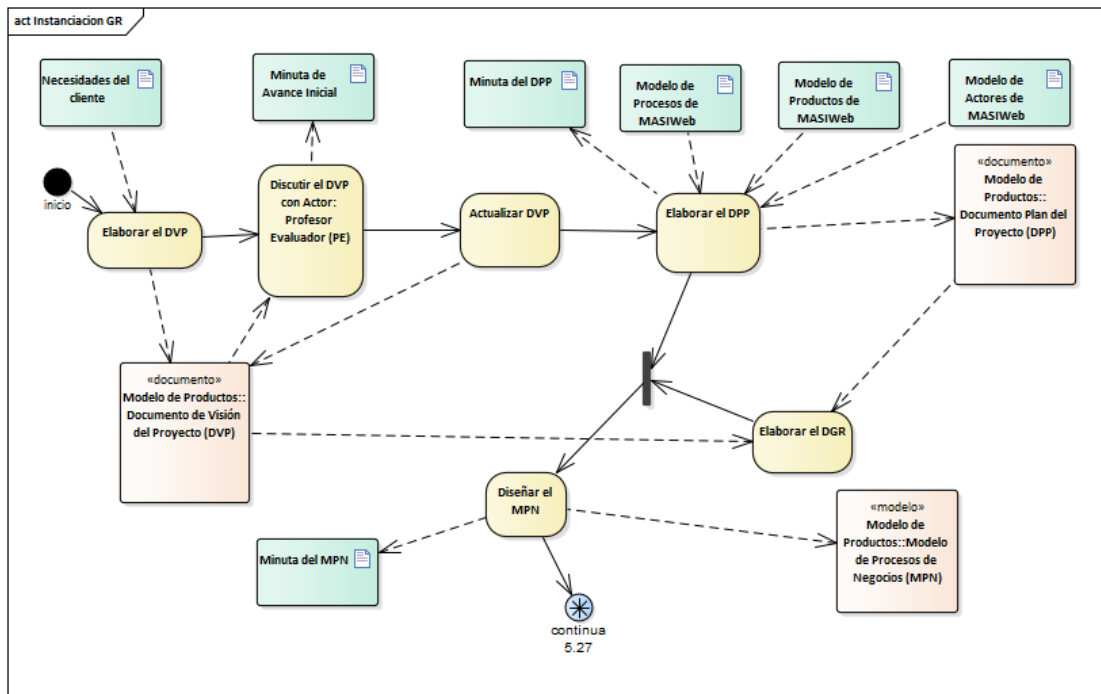


Figura 5.26. Diagrama de flujo de Situaci3n Potencial de Administraci3n de Proyectos.

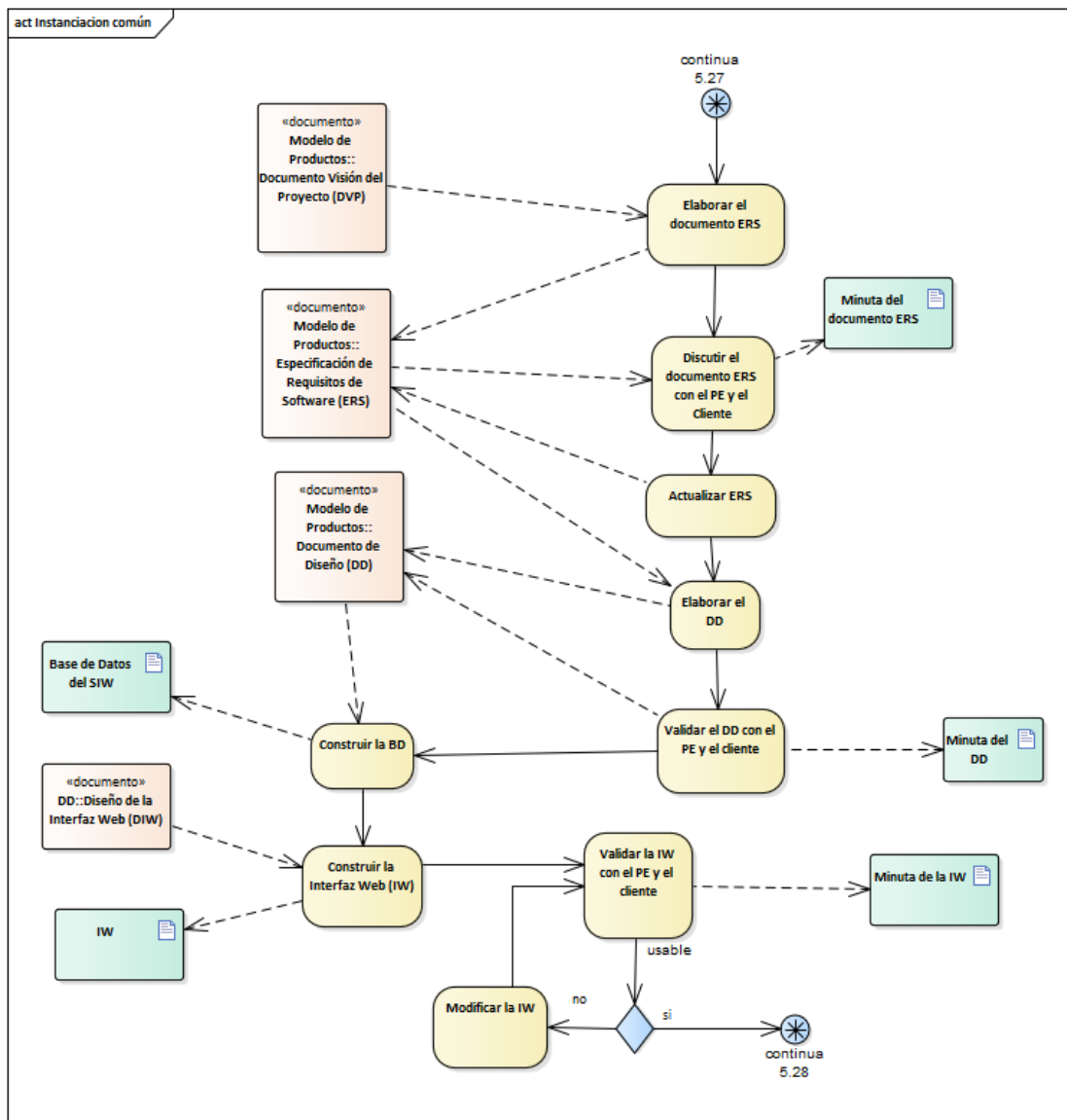


Figura 5.27. Diagrama de flujo de continuidad de los procesos potenciales (a).

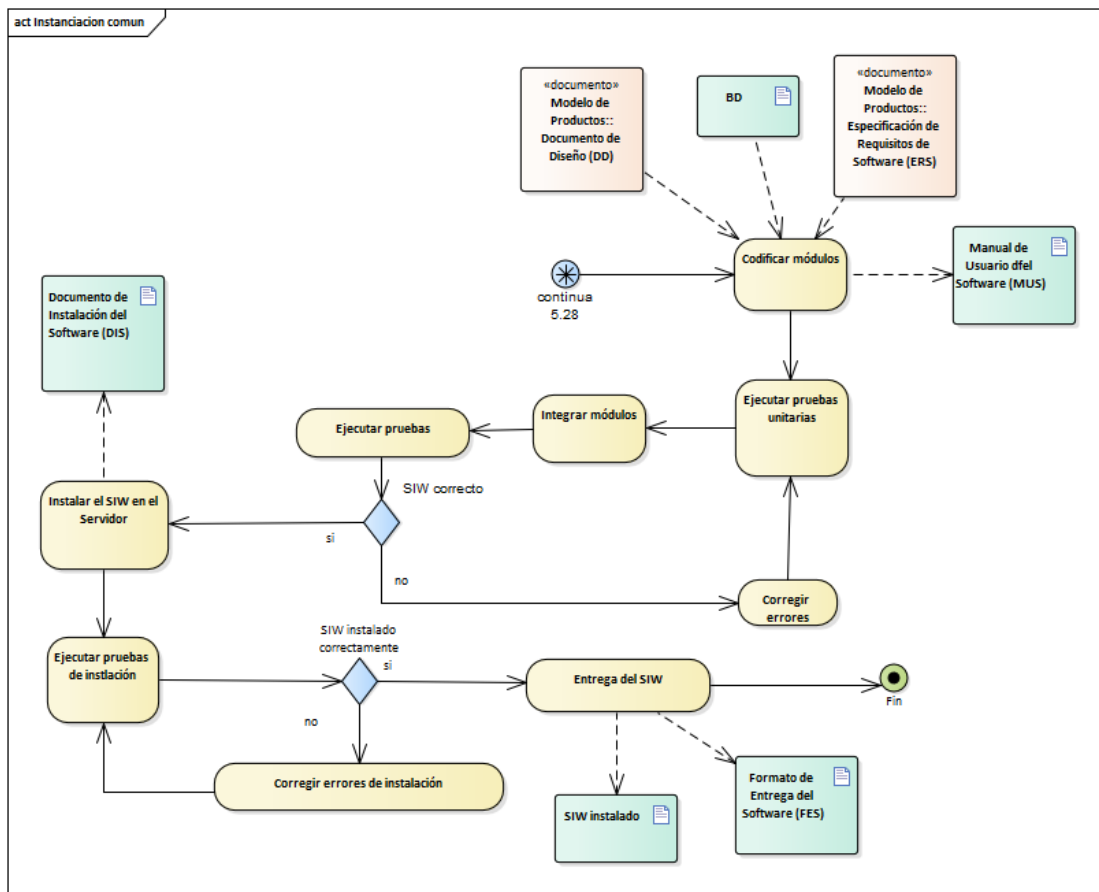


Figura 5.28. Diagrama de flujo de continuidad de los procesos potenciales (b).

## CONCLUSIONES

En el presente trabajo de investigación se logró el rediseño de la Metodología de Gil (2013), como previsto en el objetivo principal de este proyecto, produciendo el Método Académico para el desarrollo de Sistemas de Información Web basado en la Ingeniería de Métodos, MASIWeb.

Un método de desarrollo es más que una metodología de desarrollo ya que indica el cómo (procesos) ejecutar las actividades, el qué (productos) se genera en cada actividad y quién (actores) dirige las actividades para resolver un problema en específico.

La evaluación de la metodología de Gil (2013) en términos del marco de referencia propuesto por Mendoza & Barrios (2004), el cual considera cuatro (4) aspectos fundamentales: la vista del dominio, la vista de uso, la vista del producto y la vista del proceso, permitió determinar las siguientes debilidades:

- En el área de aplicación la metodología puede ser utilizada en cualquier tipo de aplicación Web.
- Considera los procesos técnicos, obviando los gerenciales lo que dificulta el desarrollo del *software* de un modo organizado y sistemático.
- No define una orientación de uso comercial, académico, publicitario o de servicios.
- No indica qué hacer ni cómo hacerlo. Tampoco indica el uso de herramientas de apoyo ni estándares.
- El modelo de proceso responde al enfoque tipo cascada.
- Las fases se dividen en aspectos y no en actividades cuya descripción es muy superficial; por consiguiente no tiene una estructura de

proceso dividido en pasos (subprocesos), actividades y tareas, lo que dificulta la comprensión del ciclo del proceso de desarrollo.

- En relación a los procesos de gestión de proyecto especifica aspectos de estimación de costos, definición del equipo de desarrollo y beneficios a obtener pero no indica técnicas ni formatos para definir los mismos.
- No existe un modelo de productos explícito, sin embargo contempla la descripción de productos en cada una de las fases.
- No especifica un Modelo de Actores, solo existe una descripción del equipo, pero no indica el formato para definir el mismo.

En la evaluación de la metodología de Gil (2013), se detectaron algunas debilidades, que no se abordaron por no corresponder al alcance determinado para este trabajo.

El rediseño de la metodología de Gil (2013) se fundamenta en la Ingeniería de Métodos y, se basa en uno de los principios de esta disciplina, el cual establece que un método debe estar definido por tres modelos: modelo de procesos, modelo de productos y modelo de actores. Estos modelos se crean en función del resultado del proceso de evaluación.

Se destaca entre los puntos relevantes de cada uno de los modelos propuestos lo siguiente:

- El Modelo de Procesos tiene las siguientes características: (1) una clasificación de los procesos en: procesos técnicos, de gestión y de soporte; (2) cuenta con procesos de gestión de proyectos, tales como organización del equipo, y el plan del desarrollo y seguimiento del proyecto; (3) se estructura como una cadena de valor, la cual identifica y relaciona los procesos; (4) se define una jerarquía de diagramas para cada uno de los procesos para manejar mejor su complejidad; (5) los

subprocesos o pasos se descomponen en actividades permitiendo describir los procesos detalladamente; (6) cada proceso es representado mediante una tabla que asocia los pasos, actividades, herramientas, técnicas, notaciones y, productos generados de forma explícita. Las actividades se especifican a través de un conjunto de acciones a ejecutar en un orden establecido; (7) los productos o insumos de entrada y salida se especifican para cada subproceso; (8) se definen los procesos de soporte: gestión de riesgos y de requisitos; (9) se describe claramente el propósito específico y orientación de uso del método; (10) se describe con detalle los tipos de pruebas a ejecutar; (11) y los requisitos de calidad requeridos; (12) las técnicas para evaluar la facilidad de uso; (13) el grado de participación y el rol del usuario y del Profesor Evaluador (PE) en la validación de los procesos y subprocesos propuestos como parte del modelo.

En cuanto a las mejoras de algunos elementos o propiedades de la metodología de Gil (2013) se describen las siguientes: (1) se reestructuran los procesos técnicos en análisis y requisitos, diseño, programación & integración y pruebas; (2) el proceso de gestión de proyecto se ejecuta a lo largo de los procesos técnicos; (3) las actividades de identificar los requisitos, diseñar la base de datos y la estructura de navegación se definen en los procesos correspondientes.

- Para el Modelo de Productos se definen completamente, con las siguientes propiedades: (1) los productos generados en el proceso de desarrollo son clasificados en Productos Entregables Intermedios (PEI) y Productos Entregables Finales (PEF); (2) son representados en un diagrama de clase indicando los productos en cada grupo correspondiente; (3) la documentación de los productos a generar en

cada uno de los procesos se realiza utilizando tablas y diagramas de clases, detallando hasta el nivel de atributos de los mismos; (4) se definen productos para la entrega final de la aplicación y para la estructura del equipo de desarrollo; (5) se elimina el término *bechmarking* y, se indica la técnica de análisis competitivo generando el producto Listas de Sitios Similares (LSS).

- En el Modelo de Actores definido: (1) se especifican los tipos de usuarios y su grado de participación; (2) se describen los roles y responsabilidades de los miembros del equipo de trabajo (3) se define un rol de Profesor Evaluador (PE), el cual tiene como función validar y verificar los procesos, productos y ejecución exitosa del proyecto (4) se define el modelo a través de la instanciación.

El método se ha validado a través del proceso de instanciación. Para tal fin se definen dos (2) situaciones potenciales de desarrollo de proyectos: (1) donde se ejecuten procesos técnicos y (2) donde se ejecuten los procesos técnicos y los procesos de administración de proyectos.

Los aportes de este método MASIWeb, se resumen en:

1. Se basa en una estructura de procesos definida por IM a través de pasos y actividades,
2. Especificación de los productos en conjunto con las herramientas y técnicas para su elaboración, característica de un método.
3. Posee una estructura fundamentada en uno de los principios de IM, el cual establece que un método debe estar definido por tres elementos: modelo de procesos, de productos y de actores.
4. El modelo de actores se caracteriza por intercambiar los roles, esto permite que los integrantes del equipo de trabajo adquieran conocimientos y habilidades en las diferentes actividades y



responsabilidades de los diversos roles que participan en el proceso del desarrollo del proyecto.

- 5.El método es de propósito específico, uno de los principios de IM. Está dirigido a Sistemas de Información Web (WIS) y orientado al área académica. Esta orientación concreta y específica proporciona conocimientos teóricos y prácticos en la ejecución del proceso de desarrollo de *software* y habilidades en el uso de las diversas técnicas y herramientas especializadas para generar los productos.
6. Es un proceso ágilmente balanceado ya que combina prácticas ágiles con procesos y prácticas disciplinadas, algunas establecidas en SWEBOK y PMBOK.
- 7.Fomenta disciplina para la elaboración de la documentación de entregables de una manera organizada; proceso necesario para facilitar el mantenimiento en los procesos de desarrollo de *software*.
- 8.La instanciación del método indica al equipo de trabajo el orden en la cual se deberían ejecutar los procesos de cualquiera de las dos (2) situaciones potenciales definidas.

En conclusión, se dispone ahora de un método de desarrollo bien definido y documentado, el cual establece pasos, actividades, prácticas, técnicas, estándares y herramientas que los equipos de trabajo deben emplear para desarrollar sistemas de información web en el ámbito académico venezolano.

Paran consolidar los modelos creados es necesario ponerlo a prueba en proyectos reales, evaluarlo y mejorarlo si es necesario; proceso que no se ejecutó debido a la deserción de la muestra elegida (materia Ingeniería del Software II del tercer trayecto de la carrera Ingeniería de Informática del UPTVal), la cual se inició con 44 estudiantes y finalizó con 11; número que no significa un valor representativo para evaluar el método.

## RECOMENDACIONES

Para la utilización efectiva del método propuesto, se sugiere:

- El equipo de desarrolladores del *software* debe poseer conocimientos sólidos en los procesos técnicos de desarrollo de *software*.
- Aplicar este método en los proyectos de desarrollo de *software* de la institución académica, documentando las diferentes observaciones que resulten de su implementación, aplicando marcos de referencia para evaluación de métodos, con el fin de realizar mejoras. Lo señalado permitirá contar con una herramienta ajustada a los requerimientos reales del estudiante, de los avances tecnológicos y del entorno, haciendo más efectivo el proceso enseñanza aprendizaje en esta área de proyectos y por ende alcanzar los objetivos propuestos en el desarrollo de aplicaciones de sistemas de información bajo tecnología Web.
- La aplicación del método en desarrollos de sistemas de información web será parte de trabajos futuros para solventar problemas que no se resolvieron.
- Se recomienda agregar a la propuesta, actividades de Gestión de Proyecto, Gestión de Calidad y Gestión de Configuración y Cambio; lo que conllevará a un diseño del método más robusto que responda a proyectos de mayor exigencia.

## REFERENCIAS BIBLIOGRÁFICAS

- Aybüke, A. & Wohlin C. (2005). *Engineering and Managing Software Requirements*. Berlin: Springer.
- Balestrini, M. (2006). *Cómo se elabora un proyecto de investigación*. 7ma ed. Caracas: BL Consultores Asociados, Servicios.
- Barrios, J. (2002). *Material del seminario de Ingeniería de Métodos*, Universidad de Los Andes, Postgrado en Computación. Mérida, Venezuela.
- Barrios, J. & Nurcan S. (2002). *A Method Engineering proposal for the analysis and representation of the organizational impact of change*, SERP'2002. Las Vegas.
- Barrios, J. & Montilva, J. (2010). *W\_WATCH: Método White\_WATCH para el desarrollo de Proyectos Pequeños de Software*. Informe Técnico Versión 1.2. Proyecto METHODIUS. Universidad de Los Andes, Mérida, Venezuela.
- Booch (1991). *Object Oriented Design with Applications*. Benjamin/Cummings Publishing.
- Bourque, P. & Fairley, R. E. (2014). *Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0*. IEEE Computer Society Press.

- Brinkkemper, S. (1995). Method Engineering: Engineering of Information Systems Development Methods and Tools. *Information and Software Technology*, 37(11).
- Brito A., K. (2009). *Selección de metodologías de desarrollo para aplicaciones Web en la Facultad de Informática de la Universidad de Cienfuegos*, Cienfuegos, Cuba.
- Cardona, C. (2017). *Base de Datos I*. Colombia, Bogotá: Fondo editorial Areandino.
- Cheesman, J. & Daniels, J. (2001). *UML Component. A simple process for specifying Component Based Software*. EEUU: Addison Wesley.
- Chin, J. P., Diehl, V. A., & Norman, K. L. (1988, Mayo). *Development of an instrument measuring user satisfaction of the human-computer interface*. Trabajo presentado en In Proceedings of the SIGCHI conference on Human factors in computing systems. Recuperado de [https://www.researchgate.net/publication/248594191\\_Development\\_of\\_a\\_n\\_Instrument\\_Measuring\\_User\\_Satisfaction\\_of\\_the\\_Human-Computer\\_Interface/related](https://www.researchgate.net/publication/248594191_Development_of_a_n_Instrument_Measuring_User_Satisfaction_of_the_Human-Computer_Interface/related)
- Conallen, J. (1999). Modelling Web Applications Architectures with UML. *Communications of the ACM*, 42(10), 63-70.
- Eriksson, H. & Penker, M. (2000). *M. Business Modeling with UML*. Wiley
- Escalona, M. J. & Koch, N. (2002). *Ingeniería de Requisitos en Aplicaciones para la Web—Un estudio comparativo*. Universidad de Sevilla.

García C., F. (2007). *La investigación tecnológica*. 2da ed. México: Limusa.

Gil, D. M. (2013). *Metodología "Creación de Sitios Web"*. Trabajo de Ascenso presentado ante el Consejo Directivo del Instituto Universitario de Tecnología de Valencia, para optar a la categoría de Profesor Asistente no publicada. Instituto Universitario de Tecnología de Valencia, Carabobo, Venezuela.

Ginige, A. & Murugesan, S. (2001). Web engineering: a introduction. *IEEE Multimedia*, 8(1): 14-18.

Hammar, C. (2001). Designing User-Centered Web Applications in Web Time. *IEEE Software*, 18(1), 62-29.

Harmsen, F. & Saeki, M. (1996). *Comparison of four method engineering language*. En Brinkkemper, S., Lyytinen, K., Welker, R., Method engineering: principles of method construction and tool support. (pp.209-231). Atlanta: Georgia, United States. Recuperado de [http://link.springer.com/chapter/10.1007/978-0-387-35080-6\\_14](http://link.springer.com/chapter/10.1007/978-0-387-35080-6_14)

Hassan M., Y. (2004). Arquitectura de la Información en los entornos virtuales de aprendizaje. Aplicación de la técnica de *CardSorting* y análisis cuantitativo de los resultados. *El Profesional de la Información*, 13(2): 93-99

Hernández, R., Fernández, C. & Baptista, L. (2006). *Metodología de la Investigación*. 4ta ed. México: MacGrawHill.

IEEE (1993). *IEEE Standard 610.12-1990*. IEEE Software Engineering Standards Collection. New York: Institute of Electrical and Electronics Engineers.

Jacobson, I. (1992). *Object Oriented Software Engineering: A use Case Driven Approach*. Addison Wesley.

Jarke, M. (1992). *Proposal for a Scenario Classification Framework*; RWTH, Lehrstuhl Informatik Germany. Recuperado de <https://www.dagstuhl.de/Reports/98/98061.doc>

Kappel, G., Pröll, B., Reich, S. & Retschitzegger, W. (2003). *Web engineering: the discipline of systematic development of Web applications*. John Wiley & Sons. Recuperado de [http://dliia.ir/Scientific/e\\_book/Technology/Electrical\\_Nuclear\\_Engine\\_Electronics/TK\\_5101\\_6720\\_Telecommunication\\_/021828.pdf](http://dliia.ir/Scientific/e_book/Technology/Electrical_Nuclear_Engine_Electronics/TK_5101_6720_Telecommunication_/021828.pdf)

Krasner, G. E., Pope, S. T. (1988). A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *Journal of object oriented programming*, 1(3), 26-49.

Leguizamón, A. (2017). *Pautas para una correcta migración de base de datos*. Universidad Libre, Facultad de Ingeniería. Bogotá, Colombia. Recuperado de <https://repository.unilibre.edu.co/handle/10901/11161>

Lizardo, G. (2013). *Reutilización de Componentes de Métodos de la suite WATCH a través del Eclipse Process Framework (EPF) Composer*. (Tesis de pregrado. Escuela de Ingeniería de Sistemas. Universidad de Los Andes, Mérida, Venezuela).

- Mendoza, M. (2004). *Método Situacional para el Desarrollo de Aplicaciones Web*. (Tesis de Maestría. Universidad de Los Andes, Postgrado en Computación, Mérida, Venezuela).
- Mendoza, M. & Barrios, J. (2004). Propuestas metodológicas para el desarrollo de aplicaciones Web: una evaluación según la ingeniería de métodos. *Ciencia e Ingeniería*, 25(2), 89-96. Recuperado de <http://www.pgcomp.ula.ve/publicaciones.php>
- Montilva, J. & Barrios, J. A. (2003). Component Based Method for Developing Web Applications. *Revista Colombiana de Computación (Colombian Journal of Computation)*. 4(1), 21- 34.
- Montilva, J. & Barrios, J. (2004). *A Business Modeling Method for Information Systems Development*. En Montilva, J. Besembel, I., Pérez, M. y Losavio, F. *Sistemas de Información e Ingeniería de Software: Temas Selectos*. Centro de Estudios en Informática. Mérida, Venezuela, 2004a. pp. 147-164.
- Montilva, J., Barrios, J. & Rivero, M. (2008). *GRAY WATCH Método de Desarrollo de Aplicaciones Empresariales*. Mérida, Venezuela.
- Montilva, J., Hamzan, K. & Gharawi, M. (2000). *The WATCH Model for Developing Business Software*. Published at the Proceedings of the IV International Multi conference on Systemic Cybernetics and Informatics – SCI'2000, Orlando, Florida, USA.
- Montilva, J., Montilva, W. & Barrios, J. (2011, Agosto). *A balanced and adaptable method for software development in very small enterprises: the blue WATCH variant*. Proceedings of LASES 2011, Medellín, Colombia.

- METHODIUS (2009). Estado de la industria venezolana de *software* - informe técnico abril 2008, Portal del Proyecto “Métodos y Modelos de Desarrollo de *Software* para Empresas Venezolanas – METHODIUS.
- Nielsen, J. (1993). *Usability Engineering*. New York, NY: Academic Press Professional.
- Nielsen, J. (1994). *Ten Usability Heuristics*. Recuperado de <https://www.nngroup.com/articles/ten-usability-heuristics/>
- OMG, Inc. (2008). “*Business Process Model and Notation, V1.1*”. Identificación: formal/2008-01-17
- Odell, J.J. (1996). A. Primer to Method Engineering. *INFOSYS: The electronic newsletter for information system*. 3(19).
- O'Donnell, J. T. & Hall, C. V. (1988). Debugging in applicative languages. *Lisp and symbolic computation*, 1(2), 113-145.
- O'Mahony, S. (2003). Guarding the Commons: How Community Managed *Software* Projects Protect their Work. *Research Policy*, 32(7): pp. 1179-1198.
- Pintos, J. (2014). *UF 1843: Aplicación de técnicas de usabilidad y accesibilidad en el entorno cliente*. Antequera, España: IC editorial
- Rumbaugh, J. (1991). *Modelado Orientado a Objetos*. Prentice Hall.



PMI, (2017). *Guide to the Project Management Body of Knowledge (PMBOK Guide)*.6ta. ed. Harrisburg, Pennsylvania: Project Management Institute, Newtown Square.

Pressman, R. (2005). *Ingeniería del Software: Un Enfoque Práctico*. 6ta ed. México: MacGrawHill.

Sabino, C. (1995). *El proceso de investigación*. Caracas: Panapo de Venezuela.

Shneiderman, B. (1986a). *Eight Golden Rules of Interface Design*. Recuperado de <https://www.cs.umd.edu/users/ben/goldenrules.html>

Shneiderman, B. (1986b). *Designing the user interface: Strategies for effective human-computer interaction*. Addison-Wesley, Reading MA.

Suárez T., M. (2011). *SIRIUS: Sistema de Evaluación de la Usabilidad Web Orientado al Usuario y basado en la Determinación de Tareas Críticas*.(Tesis doctoral). Universidad de Oviedo, España. Recuperado de <https://olgacarreras.blogspot.com/2011/07/sirius-nueva-sistema-para-la-evaluacion.html>

Ralyté, J. (2001). *Ingénierie des Méthodes à base de composants*. (Tesis Doctoral, Universidad de París 1 –S). Recuperado de <https://www.theses.fr/2001pa010006>

RAE, (2001). *Diccionario*. Recuperado de <http://www.rae.es>.

Schewe, M. H., Thalheim, K. D. & Zhao, J. (2005). View Integration and cooperation in databases, data warehouse and web information

systems. *In Journal on Data Semantics IV* (pp. 213-249). Springer, Berlin, Heidelberg.

Sommerville, I. (2005). *Ingeniería del software*. 7ma ed. Madrid, España: Pearson Addison Wesley.

Weitzenfeld, A. (2004). *Ingeniería de Software orientado a Objetos con UML, Java e Internet*. México: Thomson. Recuperado de <https://ingenieriasoftware2011.files.wordpress.com/2011/07/ingenieria-de-software-orientada-a-objetos-con-uml-java-e-internet.pdf>

## **ANEXOS**

## **ANEXO A**

**Proyectos Socio Tecnológico (PST) disponibles en la Biblioteca “Centro de Información y Documentación Biblioteca Ing. Nelson Rivero” de la UPTVal, Trayecto III, Período 1-2017**

Nombre de Proyecto Socio Tecnológico	Tipo de Aplicación	Tamaño del Grupo	Metodología usada
Gestión de Proyecto Socio Tecnológico en Los Programas Nacionales de Formación como apoyo de la función Investigativa del Instituto Universitario de Valencia	AW	2	UWE
Aplicación Informática para el proceso de gestión de estudiantes y notas de la Escuela Básica de Educación Domingo Navas Spinola	AW	2	Marco de trabajo para la Ingeniería Web
Aplicación Informática para expedición de documentos del Consejo Comunal San José II del Municipio Miranda Estado Carabobo	AW	2	RUP
Sistema Informático Web de Control de Inventario	AW	2	XP
Sistema Informático de Gestión Académica y Administrativa	AW	2	RUP
Sistema de Facturación de Compra y Venta	AW	1	XP
Sistema Informático Web para el control de reservas y proveedoras de servicios al campamento Pueblo vaquero C.A.	AW	2	XP
Aplicación Web para el control de reservaciones de la sala de reuniones de la empresa Mann+Hummer Filtration Technology Venezuela	AW	2	XP
Sistema Informático para el Registro de información del concejo comunal "Luz y Verdad" de la Urbanización la Isabelica	AW	3	XP
Sistema Informático Web para el control y registro de visitantes del MUVA (Museo de Arte Valencia) Edo. Carabobo	AW	3	XP
Sistema Informático para la gestión de servicio Comunitario	AW y E	3	XP
Sistema Informático para la gestión de Misión Sucre	AW	4	XP
Sistema Administrativo empresa Punto Olas	AW	3	XP
Sistema para la gestión de mantenimiento de camiones y herramientas de Sermatec	AW	3	XP
Sistema MIKA-RELOJ	E	1	XP
Sistema Ambulatorio Ciudad Alianza	E	3	XP
Sistema Informático para Gestión y Control de Inventario	E	2	RUP
Sistema Informático para la Gestión y Control de Inventario PROCARVEN	E	3	XP
Herramienta Informática para el Centro Internacional de Estudios Superiores	E	2	XP
Sistema Informático de Gestión de Datos e Información de Estudiantes Casa Hogar	E	3	RUP
Sistema Informático para el control de inventario de la empresa Fachacar C.A., ubicada en los Guayos	E	2	XP
Sistema Informático para el registro de notas de la Unidad Curricular Acreditables del Programa de Formación de Informática del Instituto Universitario de Tecnología de Valencia.	E	4	XP
Sistema informático para la organización de los estudios radiológicos en el Hospital Dr. Miguel Malpica	E	3	XP

Leyenda: AW (Aplicación Web); E(Escritorio)

## **ANEXO B**

**Medias y desviación estándar de los grupos de criterios del QUIS  
(Shneiderman, 1986b, Chin, Diehl & Norman, 1988).**

	Criterio positivo		Criterio negativo		
	Media	$\sigma$	Media	$\sigma$	
<b>Reacciones globales del software</b>					
1.	7.21	1.21	4.44	2.41	***
2.	7.12	1.43	3.29	2.14	***
3.	6.68	1.35	3.75	2.14	***
4.	5.59	2.05	4.33	2.61	
5.	7.00	1.52	5.06	2.73	**
6.	6.28	1.76	3.52	2.53	***
<b>Pantalla</b>					
7. El texto en la pantalla son	6.94	1.80	7.19	1.80	
8. Resaltado simplifica la tarea	6.20	1.81	6.12	2.50	
9. Organización de la información	6.29	1.62	5.76	1.89	
10. Secuencia de pantallas	6.45	1.46	5.69	1.48	
<b>Terminología</b>					
11. Uso de términos en todo el sistema	7.09	1.42	6.68	1.62	
12. Está relacionada con la tarea que realiza	6.39	1.90	5.79	1.93	
13. Posición de los mensajes	7.24	1.44	6.25	2.08	
14. Descripción de las entradas de datos	6.03	1.98	5.00	2.52	
15. Informa sobre el progreso	6.24	1.89	5.24	2.46	
16. Mensajes de error	5.97	2.15	4.71	3.24	
<b>Aprendizaje</b>					
17. El aprendizaje de usar el sistema es	5.67	2.10	3.53	2.67	
18. Explorar nuevas funciones por ensayo y error	5.62	2.00	3.76	2.46	
19. Recordar nombres y usar comandos	5.52	2.40	4.56	2.68	
20. Ejecutas las tareas es sencillo	5.94	1.72	4.65	2.50	
21. ella son	5.94	2.11	4.94	3.01	
22. Material de referencia complementarios	5.29	2.00	3.93	2.74	
<b>Capacidades del sistema</b>					
23. Velocidad del sistema	6.09	2.28	4.29	2.61	*
24. El sistema es fiable	7.45	1.50	6.35	1.66	*
25. Los sonidos generados son	6.50	2.05	7.14	1.66	
26. Corregir los errores	6.64	1.92	4.71	2.44	**
27. Diseñado para todos los niveles de usuarios	5.63	1.81	3.88	2.09	**

Leyenda:

\*denota  $\alpha < .05$

\*\* denota  $\alpha < .01$

\*\*\*denota  $\alpha < .001$