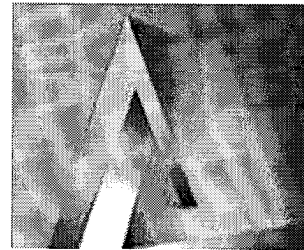


Universidad de Los Andes
Facultad de Ingeniería
Postgrado en Computación



Método Situacional para el Desarrollo de Aplicaciones *Web*

WWW.BDIGITAL.ULA.VE

SERBIULA
Tulio Febres Cordero

Realizado por: Ing. Marianella Mendoza B.
Tutor: Dra. Judith Barrios

DONACION

Proyecto de Grado presentado ante la Ilustre Universidad de Los Andes como
requisito parcial para optar al título de:
Magíster Scientiae en Computación

Licencia Creative Commons:
Mérida, Mayo 2004

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

WWW.BDIGITAL.ULA.VE

Le doy Gracias a **Dios** por guiar mi vida y por permitirme dedicar este nuevo logro a los tesoros mas preciados de mi vida: Mi esposo **Marcos**, y mis hijos **Nella Vanessa** y **Marcos Germán**.

Nella.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

AGRADECIMIENTO

De manera muy especial quiero agradecer a mi tutora de tesis, **Dra. Judith Barrios**, por brindarme sus conocimientos, y por su ayuda y estímulo a lo largo del desarrollo de esta investigación.

Asimismo, agradezco al **Dr. Wladimir Rodríguez** por sus consejos, los cuales fueron de gran ayuda para la culminación de este proyecto.

También quiero agradecer a **Luisa** por todo el apoyo, cariño y estímulo brindado a lo largo de todo el postgrado.

Además, quiero darle las gracias a mi hermana **Maria Alejandra**, mi cuñado **Hildebrando** y mis sobrinos **Luis Ernesto** y **José Alejandro** quienes me acompañaron y brindaron todo su cariño y apoyo.

Finalmente, le agradezco a mis amigos **Marlene, Samar, Omixa, Omar, Yudith...** por todo su apoyo y por sus palabras de aliento en momentos difíciles.

A todos **MUCHISIMAS GRACIAS...**

Nella.

RESUMEN

En los últimos años han surgido un conjunto de métodos para desarrollar aplicaciones *web*, los cuales no toman en cuenta los principios de la ingeniería de métodos, ni tampoco los que son proporcionados por la ingeniería *web*.

Un método para desarrollar aplicaciones *web* debe estar conformado por un modelo de producto, que describe las características genéricas de una aplicación *web* y un modelo de proceso, que indica las actividades necesarias para desarrollar este tipo de aplicaciones.

Ninguno de los modelos de procesos asociados a estos métodos relacionan las actividades a ejecutar en el ciclo de desarrollo con los modelos de producto asociados a cada una de las aplicaciones *web*

Asimismo, cabe destacar que ninguno de los métodos evaluados están orientados al contexto, es decir, que simplemente son un conjunto de fases y pasos predefinidos y lógicamente ordenados que no indican en qué situaciones particulares se deben ejecutar.

Para solventar estas deficiencias, se presenta en esta investigación una propuesta de un método situacional para el desarrollo de aplicaciones *web*, que cumpliendo con los principios de la ingeniería de métodos y de la ingeniería *web*, es orientado al contexto. Este método ofrece la posibilidad, al grupo de desarrollo, de elegir entre múltiples alternativas la más conveniente, en función del dominio de aplicación y de la situación particular enfrentada por tal grupo de desarrollo.

Nuestro método contiene una descripción clara y explícita del modelo de producto y del modelo de proceso mostrando la integración entre ambos, se adapta a diferentes contextos de modelado y garantiza una guía completa al equipo de desarrollo, tomando en cuenta las necesidades particulares del dominio de aplicación seleccionado (comercio electrónico, gobierno electrónico, etc).

El modelo de proceso se expresa como mapas intencionales. Al final se muestra el uso del método para un caso real o aplicación.

Palabras claves: Aplicación *Web*, Método de Desarrollo, método situacional, Ingeniería de Métodos.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

INDICE GENERAL

INTRODUCCION	1
CAPITULO I. EL PROBLEMA	
1. PLANTEAMIENTO DEL PROBLEMA	3
2. OBJETIVOS DE LA INVESTIGACION	4
3. JUSTIFICACION DE LA INVESTIGACION	5
4. METODOLOGIA	6
5. SELECCIÓN DE METODOS PARA EL DESARROLLO DE APLICACIONES WEB	7
6. RESULTADOS ESPERADOS	14
CAPITULO 2. MARCO TEORICO	
1. APLICACIONES WEB	15
1.1. Caracterización de las Aplicaciones Web	15
1.2. Tipos de Aplicaciones Web	18
1.3. Métodos para el Desarrollo de Aplicaciones Web	21
1.3.1. Métodos Ágiles para el Desarrollo de Aplicaciones	21
1.3.2. Métodos para el Desarrollo de Sistemas de Información Web (SIW)	22
1.3.3. Métodos para el Desarrollo de Aplicaciones Hipermedia	23
1.3.4. Métodos para el Desarrollo de Sitios Web Instruccionales	24
1.3.5. Métodos para el Desarrollo de Aplicaciones de Comercio Electrónico (e-Commerce)	25
1.3.6. Metodologías para el Desarrollo de Aplicaciones Web	25
2. INGENIERIA DE METODOS	27
2.1. Enfoques de la Ingeniería de Métodos	28
2.1.1. Meta-modelización	29
2.1.2. Orientación Situacional	32
2.2. Técnicas para la Construcción de Métodos Situacionales	33
2.2.1. Ingeniería de Métodos Basada en el Ensamblaje	34
2.2.2. Ingeniería de Métodos Basada en la Extensión	35
2.2.3. Ingeniería de Métodos Basada en el Paradigma	35
3. INGENIERIA DE DOMINIO	37
3.1. DOMINIO	37
3.2. FASES DE LA INGENIERIA DE DOMINIO	39
3.2.1. Análisis del Dominio	39
3.2.2. Diseño del Dominio	40
3.2.3. Implementación del Dominio	41
3.3. METODOS DE LA INGENIERIA DE DOMINIO	41
3.3.1. FODA (Feature-Oriented Domain Analysis)	41
3.3.2. ODM (Organization Domain Modeling)	42
3.3.3. RSEB (Reuse-Driven Software Engineering Business)	42
3.3.4. PuSE (Product Line Software Engineering)	43
3.3.5. FAST (Family-Oriented Abstraction, Specification, and Translation)	43
4. REUTILIZACION DE SOFTWARE	44
4.1. PRODUCTOS REUTILIZABLES	44
4.2. PROCESO DE DESARROLLO DE SOFTWARE BASADO EN LA REUTILIZACION DE COMPONENTES	45
4.3. TECNOLOGIAS DE COMPONENTES	46
4.3.1. Common Object Request Broker Architecture (CORBA)	46
4.3.2. Java 2 Platform Enterprise Edition (J2EE)	47
4.3.3. .NET	48
5. MODELO DE MADUREZ DE CAPACIDAD INTEGRADO (CMMI)	49

5.1. NIVELES DE MADUREZ	49
6. CONCLUSION	51

CAPITULO 3. MODELO DE PRODUCTO DE UNA APLICACIÓN WEB

1. CARACTERISTICAS DE LAS APLICACIONES WEB SELECCIONADAS	53
1.1. PORTALES	55
1.1.1. Infraestructura de Componentes de un Portal	56
1.1.2. Arquitectura de un Portal	57
1.2. APLICACIONES DE COMERCIO ELECTRONICO (<i>e-Commerce</i>)	59
1.2.1. Etapas de las Aplicaciones de Comercio Electrónico	60
1.2.2. Arquitectura de una Aplicación de Comercio Electrónico	62
1.3. APLICACIONES DE NEGOCIOS INTELIGENTES (<i>e-Business</i>)	63
1.3.1. Etapas de las Aplicaciones <i>e-Business</i>	66
1.3.2. Arquitectura de las Aplicaciones <i>e-Business</i>	67
1.4. SISTEMAS DE INFORMACION WEB (SIW)	68
1.4.1. Componentes básicos de un SIW	69
1.4.2. Arquitectura de un SIW	71
1.5. GOBIERNOS ELECTRONICOS (<i>e-Government</i>)	72
1.5.1. Evolución del Gobierno Electrónico	74
1.5.2. Arquitectura de una Aplicación de Gobierno Electrónico	76
2. ASPECTOS COMUNES ENTRE LAS APLICACIONES WEB SELECCIONADAS	77
3. MODELO DE PRODUCTO DE UNA APLICACIÓN WEB	78
4. MODELO DE PRODUCTO DE UN SISTEMA DE NEGOCIOS	80
5. MODELO DE PRODUCTO ASOCIADO AL CICLO DE VIDA DE UNA APLICACIÓN WEB	82
5.1. SUB-MODELO DE GESTION DE PROYECTO	83
5.2. SUB-MODELO DE LA APLICACIÓN WEB	85
5.3. SUB-MODELO DE IMPLANTACION Y MANTENIMIENTO DE LA APLICACIÓN WEB	87
6. CONCLUSION	87

CAPITULO 4. GENERALIZACION DE LOS MODELO DE PROCESOS PARA DESARROLLAR APLICACIONES WEB

1. METODOS PARA DESARROLLAR LAS APLICACIONES WEB SELECCIONADAS	89
1.1. METODO PARA EL DESARROLLO DE PORTALES	90
1.2. METODO PARA EL DESARROLLO DE APLICACIONES DE COMERCIO ELECTRONICO (<i>e-commerce</i>)	92
1.3. METODO PARA EL DESARROLLO DE APLICACIONES <i>e-Business</i>	93
1.4. METODO PARA EL DESARROLLO DE SISTEMAS DE INFORMACION WEB (SIW)	95
1.5. METODO PARA LA IMPLANTACION DEL GOBIERNO ELECTRONICO (<i>e-Government</i>)	98
2. EVALUACION DE LOS METODOS SELECCIONADOS	101
2.1. LA VISTA DE DOMINIO	102
2.2. LA VISTA DE USO	103
2.3. LA VISTA DEL PRODUCTO	104
2.4. LA VISTA DEL PROCESO	105
3. ASPECTOS COMUNES Y DIFERENTES ENTRE LOS METODOS PARA DESARROLLAR LAS APLICACIONES WEB SELECCIONADAS	107
4. FASES COMUNES DE DESARROLLO DE UNA APLICACIÓN WEB	108
5. CONCLUSION	111

Licencia Creative Commons:

CAPITULO 5. METODO SITUACIONAL PARA EL DESARROLLO DE APLICACIONES WEB: MODELO DE PROCESO	
1. PROCESO DE CONSTRUCCION DEL METODO SITUACIONAL	112
2. MODELADO DE NEGOCIO	113
3. MODELO DE PROCESO PARA EL DESARROLLO DE APLICACIONES <i>WEB</i>	117
3.1. NIVELES DE ABSTRACION DE LOS MAPAS INTENCIONALES	117
3.2. MAPA GLOBAL	118
3.3. MAPAS LOCALES DE ALTO NIVEL	124
3.3.1. GESTIONAR EL PROYECTO	124
3.3.2. DESARROLLAR LA APLICACIÓN <i>WEB</i>	128
3.3.3. IMPLANTAR Y MANTENER LA APLICACIÓN <i>WEB</i>	135
3.4. MAPAS LOCALES DE BAJO NIVEL: INTENCION DESARROLLAR LA APLICACIÓN <i>WEB</i>	139
3.4.1. DEFINIR Y ESPECIFICAR REQUERIMIENTOS	139
3.4.2. DISEÑAR LA APLICACIÓN <i>WEB</i>	144
3.4.3. IMPLEMENTAR LA APLICACIÓN <i>WEB</i>	153
3.4.4. PROBAR LA APLICACIÓN <i>WEB</i>	163
4. RECORRIDO DE LOS MAPAS DE PROCESOS: UNA APLICACIÓN REAL	169
4.1. RECORRIDO DEL MAPA GLOBAL	169
4.2. RECORRIDO DEL MAPA LOCAL DE ALTO NIVEL: GESTIONAR EL DESARROLLO DE LA APLICACIÓN <i>WEB</i>	172
4.3. RECORRIDO DEL MAPA LOCAL DE ALTO NIVEL: DESARROLLAR LA APLICACIÓN <i>WEB</i>	173
4.4. RECORRIDO DEL MAPA LOCAL DE BAJO NIVEL DE LA INTENCION "DESARROLLAR LA APLICACIÓN <i>WEB</i> ": DEFINIR Y ESPECIFICAR REQUERIMIENTOS	175
4.5. RECORRIDO DEL MAPA LOCAL DE BAJO NIVEL DE LA INTENCION "DESARROLLAR LA APLICACIÓN <i>WEB</i> ": DISEÑAR LA APLICACIÓN <i>WEB</i>	177
4.6. RECORRIDO DEL MAPA LOCAL DE BAJO NIVEL DE LA INTENCION "DESARROLLAR LA APLICACIÓN <i>WEB</i> ": IMPLEMENTAR LA APLICACIÓN <i>WEB</i>	179
4.7. RECORRIDO DEL MAPA LOCAL DE BAJO NIVEL DE LA INTENCION "DESARROLLAR LA APLICACIÓN <i>WEB</i> ": PROBAR LA APLICACIÓN <i>WEB</i>	181
5. CONCLUSION	182
CONCLUSIONES	184
RECOMENDACIONES	188
REFERENCIAS Y BIBLIOGRAFIA	189

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

INDICE DE FIGURAS

CAPITULO 2. MARCO TEORICO

Figura 1. Árbol de Requisitos de Calidad	17
Figura 2. Niveles de Abstracción de un Fragmento de Método	28
Figura 3. Modelo y Meta-Modelo de un Diagrama de Clases	29
Figura 4. Meta-Modelo para Construir Mapas de Procesos.	30
Figura 5. Ejemplo de un Mapa (Modelo de Procesos).	32
Figura 6. El proceso de la Ingeniería de Métodos Situacional	33
Figura 7. Modelo de Procesos Basado en el Ensamblaje para Ingeniería de Métodos Situacional.	34
Figura 8. Modelo de Procesos Basado en Extensión para Ingeniería de Métodos Situacional.	35
Figura 9. Modelo de Procesos Basado en Paradigma para Ingeniería de Métodos Situacional.	36

CAPITULO 3. MODELO DE PRODUCTO DE UNA APLICACIÓN WEB

Figura 10. Diagrama de caso de uso de un portal	56
Figura 11. Modelo de Producto de un Portal	58
Figura 12. Diagrama de Casos de Uso de Una Aplicación de Comercio Electrónico	61
Figura 13. Etapas de las Aplicaciones de Comercio Electrónico	62
Figura 14. Modelo de Producto de una Aplicación de Comercio Electrónico	63
Figura 15. Diagrama de Casos de Uso de las Aplicaciones <i>e-Business</i>	65
Figura 16. Etapas de las Aplicaciones de Negocios Inteligentes (<i>E-Business</i>).	66
Figura 17. Modelo de Producto de una Aplicación de Negocios Inteligentes (<i>e-Business</i>)	68
Figura 18. Diagrama de Casos de Uso de un Sistema de Información Web (WIS)	70
Figura 19. Modelo de Producto de un Sistema de Información Web (SIW)	72
Figura 20. Diagrama de Casos de Uso de una Aplicación de Gobierno Electrónico (<i>e-Government</i>)	73
Figura 21. Evolución de las Aplicaciones de Gobierno Electrónico	75
Figura 22. Modelo de Producto de una Aplicación de Gobierno Electrónico (<i>e-Government</i>)	76
Figura 23. Diagrama de Capas que agrupa a las Aplicaciones Web Seleccionadas	78
Figura 24. Modelo de Producto de una Aplicación Web en general	79
Figura 25. Modelo de Producto de un Sistema de Negocio	82
Figura 26. Modelo de Producto Asociado al Ciclo de Vida de la Aplicación Web	83
Figura 27. Sub-Modelo de Gestión de Proyecto	84
Figura 28. Sub-Modelo de la Aplicación Web	86
Figura 29. Sub-Modelo de Implantación y Mantenimiento de la Aplicación Web.	87

CAPITULO 4. GENERALIZACION DE LOS MODELO DE PROCESOS PARA DESARROLLAR APLICACIONES WEB

Figura 30. Modelo de Procesos asociado al Método para el Desarrollo de un Portal	91
Figura 31. Modelo de Procesos asociado al Método para el Desarrollo de <i>e-Commerce</i> .	92
Figura 32. Modelo de Procesos asociado a Método para el Desarrollo de <i>e-Business</i>	94

Figura 33. Modelo de Procesos asociado al Método para el Desarrollo de Sitios Web Corporativos	97
Figura 34. Modelo de Procesos asociado al Método para la Implantación de Gobierno Electrónico (<i>e-Government</i>).	98
Figura 35. Fases comunes encontradas en los métodos para el desarrollo de las aplicaciones <i>web</i> seleccionadas	109

CAPITULO 5. METODO SITUACIONAL PARA EL DESARROLLO DE APLICACIONES WEB: MODELO DE PROCESO

Figura 36. Mapa de la Intención de <i>Modelar el Negocio</i>	114
Figura 37. Mapa Global del Método Situacional para el Desarrollo de Aplicaciones <i>Web</i>	118
Figura 38. Mapa Local de Alto Nivel de la Intención de <i>Gestionar el Proyecto</i> del Método Situacional para el Desarrollo de Aplicaciones <i>Web</i>	124
Figura 39. Mapa Local de Alto Nivel de la Intención de <i>Desarrollar la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones <i>Web</i>	128
Figura 40. Mapa Local de Alto Nivel de la Intención de <i>Implantar y Mantener la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones <i>Web</i>	135
Figura 41. Mapa Local de Bajo Nivel de la Intención de <i>Definir y Especificar Requerimientos</i> del Método Situacional para el Desarrollo de Aplicaciones <i>Web</i>	140
Figura 42. Mapa Local de Bajo Nivel de la Intención de <i>Diseñar la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones <i>Web</i>	145
Figura 43. Mapa Local de Bajo Nivel de la Intención de <i>Implementar la Aplicación Web (.NET)</i> del Método Situacional para el Desarrollo de Aplicaciones <i>Web</i>	154
Figura 44. Mapa Local de Bajo Nivel de la Intención de <i>Implementar la Aplicación Web (J2EE)</i> del Método Situacional para el Desarrollo de Aplicaciones <i>Web</i>	155
Figura 45. Mapa Local de Bajo Nivel de la Intención de <i>Probar la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones <i>Web</i>	164
Figura 46. Recorrido del Mapa Global del Método Situacional para el Desarrollo de Aplicaciones <i>Web</i>	170
Figura 47. Recorrido del Mapa Local de Alto Nivel de la Intención de <i>Gestionar el Proyecto</i>	172
Figura 48. Recorrido del Mapa Local de Alto Nivel de la Intención de <i>Desarrollar la Aplicación Web</i>	174
Figura 49. Recorrido del Mapa Local de Bajo Nivel de la Intención de <i>Definir y Especificar Requerimientos</i>	176
Figura 50. Recorrido del Mapa Local de Bajo Nivel de la Intención de <i>Diseñar la Aplicación Web</i>	178
Figura 51. Recorrido del Mapa Local de Bajo Nivel de la Intención de <i>Implementar la Aplicación Web</i>	180
Figura 52. Recorrido del Mapa Local de Bajo Nivel de la Intención de <i>Probar la Aplicación Web</i>	182

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

INDICE DE TABLAS

CAPITULO 1. EL PROBLEMA

Tabla No. 1. Vista del dominio (¿Para qué sirve? ¿para qué es concebido?)	8
Tabla No. 2. Vista de Uso (Objetivos del que los usa)	9
Tabla No. 3. Vista del Producto (Resultado de aplicar el método)	10
Tabla No. 4. Vista del Proceso (Desarrollo del producto)	12

CAPITULO 2. MARCO TEORICO

Tabla No. 5. Categorías de Aplicaciones Web	18
---	----

CAPITULO 4. GENERALIZACION DE LOS MODELOS DE PROCESOS PARA DESARROLLAR APLICACIONES WEB

Tabla No. 6. Vista del dominio (¿Para qué sirve? ¿para qué es concebido?)	102
Tabla No. 7. Vista de Uso (Objetivos del que los usa)	103
Tabla No. 8. Vista del Producto (Resultado de aplicar el método)	104
Tabla No. 9. Vista del Proceso (Desarrollo del producto) (a)	106
Tabla No. 9. Vista del Proceso (Desarrollo del producto) (b)	107

CAPITULO 5. METODO SITUACIONAL PARA EL DESARROLLO DE APLICACIONES WEB: MODELO DE PROCESO

Tabla No. 10. Directivas de Ejecución de Intención (DEI) de la Intención de <i>Modelar el Negocio</i> (a)	115
Tabla No. 10. Directivas de Ejecución de Intención (DEI) de la Intención de <i>Modelar el Negocio</i> (b)	116
Tabla No. 11. Directivas de Ejecución de Intención (DEI) del Mapa Global del Método Situacional para el Desarrollo de Aplicaciones Web (a)	120
Tabla No. 11. Directivas de Ejecución de Intención (DEI) del Mapa Global del Método Situacional para el Desarrollo de Aplicaciones Web (b)	121
Tabla No. 11. Directivas de Ejecución de Intención (DEI) del Mapa Global del Método Situacional para el Desarrollo de Aplicaciones Web (c)	122
Tabla No. 11. Directivas de Ejecución de Intención (DEI) del Mapa Global del Método Situacional para el Desarrollo de Aplicaciones Web (d)	123
Tabla No. 12. Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto Nivel de la Intención de <i>Gestionar el Proyecto</i> del Método Situacional para el Desarrollo de Aplicaciones Web	127
Tabla No. 13. Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto Nivel de la Intención de <i>Desarrollar la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones Web (a)	132
Tabla No. 13. Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto Nivel de la Intención de <i>Desarrollar la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones Web (b)	133
Tabla No. 13. Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto Nivel de la Intención de <i>Desarrollar la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones Web (c)	134

Tabla No. 14. Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto Nivel de la Intención de <i>Implantar y Mantener la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones Web (a)	137
Tabla No. 14. Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto Nivel de la Intención de <i>Implantar y Mantener la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones Web (b)	138
Tabla No. 14. Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto Nivel de la Intención de <i>Implantar y Mantener la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones Web (c)	139
Tabla No. 15. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de <i>Definir y Especificar Requerimientos</i> del Método Situacional para el Desarrollo de Aplicaciones Web (a)	142
Tabla No. 15. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de <i>Definir y Especificar Requerimientos</i> del Método Situacional para el Desarrollo de Aplicaciones Web (b)	143
Tabla No. 16. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de <i>Diseñar la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones Web (a)	149
Tabla No. 16. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de <i>Diseñar la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones Web (b)	150
Tabla No. 16. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de <i>Diseñar la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones Web (c)	151
Tabla No. 16. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de <i>Diseñar la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones Web (d)	152
Tabla No. 16. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de <i>Diseñar la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones Web (e)	153
Tabla No. 17. Directivas de Ejecución de Intención (DEI) Comunes de los Mapas Locales de Bajo Nivel de la Intención de <i>Implementar la Aplicación Web con .NET y J2EE</i> del Método Situacional para el Desarrollo de Aplicaciones Web (a)	159
Tabla No. 17. Directivas de Ejecución de Intención (DEI) Comunes de los Mapas Locales de Bajo Nivel de la Intención de <i>Implementar la Aplicación Web con .NET y J2EE</i> del Método Situacional para el Desarrollo de Aplicaciones Web (b)	160
Tabla No. 18. Directivas de Ejecución de Intención (DEI) propias del Mapa Local de Bajo Nivel de la Intención de <i>Implementar la Aplicación Web con .NET</i> del Método Situacional para el Desarrollo de Aplicaciones Web (a)	160
Tabla No. 18. Directivas de Ejecución de Intención (DEI) propias del Mapa Local de Bajo Nivel de la Intención de <i>Implementar la Aplicación Web con .NET</i> del Método Situacional para el Desarrollo de Aplicaciones Web (b)	161
Tabla No. 19. Directivas de Ejecución de Intención (DEI) propias del Mapa Local de Bajo Nivel de la Intención de <i>Implementar la Aplicación Web con J2EE</i> del Método Situacional para el Desarrollo de Aplicaciones Web (a)	161

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Tabla No. 19. Directivas de Ejecución de Intención (DEI) propias del Mapa Local de Bajo Nivel de la Intención de <i>Implementar la Aplicación Web con J2EE</i> del Método Situacional para el Desarrollo de Aplicaciones <i>Web</i> (b)	162
Tabla No. 19. Directivas de Ejecución de Intención (DEI) propias del Mapa Local de Bajo Nivel de la Intención de <i>Implementar la Aplicación Web con J2EE</i> del Método Situacional para el Desarrollo de Aplicaciones <i>Web</i> (c)	163
Tabla No. 20. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de <i>Probar la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones <i>Web</i> (a)	166
Tabla No. 20. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de <i>Probar la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones <i>Web</i> (b)	167
Tabla No. 20. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de <i>Probar la Aplicación Web</i> del Método Situacional para el Desarrollo de Aplicaciones <i>Web</i> (c)	168
Tabla No. 21. Directivas de Ejecución de Intención (DEI) del Mapa Global del Método Situacional para el Desarrollo de Aplicaciones <i>Web</i> ejecutadas en el caso de estudio	171
Tabla No. 22. Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto Nivel de la Intención de <i>Gestionar el Proyecto</i> ejecutadas en el Estudio de Caso	173
Tabla No. 23. Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto Nivel de la Intención de <i>Desarrollar la Aplicación web</i> del estudio de caso	175
Tabla No. 24. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de <i>Implementar la Aplicación web</i> ejecutadas en el Estudio de caso.	181

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

INTRODUCCION

Una aplicación *web* es un sistema de *software* cuya funcionalidad es mostrada a través de la plataforma *World Wide Web (WWW)*. Desarrollar este tipo de aplicaciones es un proceso complejo que demanda amplios conocimientos técnicos y experiencia por parte del personal involucrado.

Un método para desarrollar aplicaciones *web* debe estar conformado por un modelo de producto, que describe las características genéricas de una aplicación *web* y un modelo de proceso, que indica las actividades necesarias para desarrollar este tipo de aplicaciones.

En la actualidad, han surgido una gran cantidad de métodos bien conocidos para desarrollar aplicaciones *web*, entre los cuales se pueden mencionar: los métodos ágiles, los métodos para aplicaciones hipermedia, los métodos para desarrollar aplicaciones de *e-commerce*, y los métodos para el desarrollo de aplicaciones *web* en general.

En general, estos métodos están estructurados en un conjunto de fases y pasos predefinidos y lógicamente ordenados, que guían al grupo de desarrollo en el desarrollo de un tipo de aplicación predeterminado, sin tomar en cuenta situaciones particulares de proyecto lo cual dificulta su adaptación a múltiples contextos, donde las decisiones asociadas se toman en base a la experiencia del grupo de desarrollo.

El propósito de esta investigación es proponer un método “situacional” que permita guiar el desarrollo de aplicaciones *web* para múltiples contextos. Este método ofrece la posibilidad, al grupo de desarrollo, de elegir entre múltiples alternativas, la más conveniente, en función del dominio de aplicación y de la situación particular de proyecto, independientemente de la experiencia en desarrollo de cada uno de sus miembros.

Este trabajo se estructura en cinco (5) capítulos. El primer capítulo describe la problemática existente, el planteamiento de los objetivos, la justificación, la metodología utilizada para el desarrollo de la investigación y los resultados que se esperan alcanzar al culminar la misma. También se presenta una evaluación de algunos métodos utilizados actualmente.

Licencia Creative Commons.

El segundo capítulo comprende todos aquellos aspectos teóricos necesarios para el entendimiento y desarrollo de la investigación.

En el tercer capítulo, se presenta un modelo de producto para una aplicación *web*, un modelo de producto de un sistema de negocios y un modelo de producto global.

El cuarto capítulo comprende la descripción detallada de métodos seleccionados para desarrollar cada una de las aplicaciones *web* seleccionadas, la evaluación de los aspectos comunes de cada una de ellos y la definición de un modelo de procesos genérico para desarrollar aplicaciones *web*, el cual será la base de nuestra propuesta.

El quinto capítulo contiene el conjunto de mapas intencionales que representan el modelo de proceso del método situacional para desarrollar aplicaciones *web*.

WWW.BDIGITAL.ULA.VE

CAPITULO 1

EL PROBLEMA

El presente capítulo tiene como finalidad exponer el problema objeto de estudio, formular el objetivo general y los específicos, así como presentar la justificación y metodología utilizada para el desarrollo de la presente investigación.

1. PLANTEAMIENTO DEL PROBLEMA

La Internet y la World Wide Web (WWW) han significado un gran impacto en los negocios, comercio, industria, banca, educación, gobierno, sectores de entretenimiento y hasta en nuestra vida personal, ya que ellas nos permiten descargar música, ver películas, obtener asesoramiento, hacer reservas de hoteles, comprar y vender artículos, reservar vuelos en líneas aéreas, conocer gente, hacer gestiones bancarias y recibir cursos universitarios, entre otras cosas.

En este sentido, se hace necesario el desarrollo de aplicaciones *web* funcionales, confiables y eficientes, que garanticen una respuesta adecuada a las peticiones remotas realizadas por el usuario a través del navegador, las cuales pueden implicar una mezcla de publicación impresa, mercadeo, comunicaciones internas, relaciones externas, arte, tecnología e informática.

Por esta razón, existe una gran variedad de aplicaciones *web* que van desde páginas meramente informativas hasta aplicaciones complejas que ofrecen diversidad de servicios al usuario, las cuales requieren para su desarrollo de metodologías sólidas, procesos sistemáticos e iterativos, herramientas de desarrollo y buenos lineamientos que permitan obtener un producto de alta calidad.

Actualmente, los desarrolladores *web* usan enfoques de desarrollo de aplicación *ad hoc*, lo que ocasiona pérdida de rigor, de técnicas sistemáticas y de aseguramiento de la calidad de las aplicaciones. Además, éstos consideran el desarrollo de aplicaciones *web* como un medio de manipulación y creación de presentaciones, dejando de lado, el análisis de necesidades, diseño, gerencia, métricas y mantenimiento de las aplicaciones *web*.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Tales causas originan el surgimiento de una nueva disciplina para el desarrollo de aplicaciones *web*: la Ingeniería Web, la cual es definida como el establecimiento y uso de conocimiento científico, principios de gerencia e ingeniería y el enfoque sistemático y disciplinado para desarrollar, entregar y mantener aplicaciones y sistemas basados en *web* de alta calidad (Ginige y Murigesan, 200(1).

En los últimos años, han surgido un conjunto de métodos para desarrollar aplicaciones *web*, entre los cuales se pueden mencionar: los métodos ágiles (Beck, 1999 y Mc Breen, 2000), los métodos para aplicaciones hipermedia (Montilva, 1996), los métodos para desarrollar aplicaciones de comercio electrónico (Fingar, 2000) y los métodos para el desarrollo de aplicaciones *web* en general (Pressman, 2002). Algunos de estos métodos, no toman en cuenta los principios de la Ingeniería de Métodos (Barrios, 2002), ni tampoco los proporcionados por la Ingeniería Web (Ginige y Murigesan, 2001).

La mayoría de estos métodos presentan en forma explícita su modelo de proceso, es decir, la integración de las actividades técnicas y gerenciales que son requeridas para el desarrollo de una aplicación *web* de alta calidad, y de manera implícita describen el producto, es decir, las características estructurales de las aplicaciones *web*. Sin embargo, ninguno de ellos guía al grupo de desarrollo en la construcción de aplicaciones *web* para múltiples dominios de aplicación independientemente de la experiencia de los miembros de dicho grupo.

Por lo dicho anteriormente, se plantea la necesidad de definir un método, que cumpliendo con los principios de la Ingeniería de Métodos (Kumar y Welke, 1992) y de la Ingeniería Web (Ginige y Murugesan), guíe al grupo de desarrollo en la construcción de aplicaciones *web* para múltiples dominios de aplicación, independientemente de la experiencia de los miembros de dicho grupo.

2. OBJETIVOS DE LA INVESTIGACIÓN

Objetivo General

Proponer un método que permita guiar el desarrollo de aplicaciones *web* para múltiples dominios de aplicación.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Objetivos Específicos:

- Caracterizar un método para desarrollar aplicaciones *web* basado en los principios de la Ingeniería de Métodos y de la ingeniería de *software* orientada a objetos.
- Evaluar los métodos para desarrollo de aplicaciones *web* existentes en el mercado con el fin de definir los aspectos que cumplen con los principios de la Ingeniería de Métodos y la Ingeniería Web.
- Analizar diferentes dominios de aplicación con el fin de identificar los elementos genéricos que los caracterizan.
- Definir un método para desarrollar aplicaciones *web*, tomando en cuenta los resultados obtenidos en la evaluación realizada y los elementos que caracterizan a cada dominio de aplicación analizado y la situación particular de cada proyecto.

3. JUSTIFICACION DE LA INVESTIGACION

Los continuos avances de la *World Wide Web* obligan a desarrollar aplicaciones que permitan satisfacer las necesidades de los usuarios (compras electrónicas, búsqueda de información, reservaciones de boletos, etc.) que utilizan este medio para recibir algún servicio. Por esta razón, es necesario construir aplicaciones de calidad que presenten al usuario el contenido de textos, gráficos, sonido o video, que sea actualizada constantemente, y que sea capaz de interactuar con otros elementos de *Internet* o efectuar transacciones automáticas en otros portales.

Para ello, se requiere que el grupo de desarrollo cuente con un método que contenga una descripción clara y explícita de un modelo de producto y de un modelo de proceso, que oriente en la manipulación que deben sufrir los distintos elementos para poder construir la aplicación *web*. Además, el método debe mostrar explícitamente la relación e integración entre ambos. El método debe adaptarse a los diferentes contextos de modelado a través de la toma de decisiones según la situación particular enfrentada en cada proyecto y, que

Licencia Creative Commons.

garantice una guía completa tomando en cuenta las necesidades particulares del dominio de aplicación seleccionado (comercio electrónico, negocio electrónico, etc.).

Las razones anteriormente descritas son las que justifican la “PROPUESTA DE UN METODO SITUACIONAL QUE PERMITA GUIAR AL GRUPO DE DESARROLLO DE APLICACIONES WEB PARA MULTIPLES DOMINIO DE APLICACIÓN”.

Este método, entre otros beneficios, ofrece la posibilidad al grupo de desarrollo de elegir entre múltiples alternativas la más conveniente en función del dominio de aplicación y de la situación del proyecto en desarrollo; es decir, los elementos del contexto de desarrollo, el tipo de aplicación, los usuarios, las herramientas, la tecnología y la experiencia de desarrollo en proyectos.

4. METODOLOGIA

Las actividades ejecutadas durante el desarrollo de esta investigación son:

1. Investigación bibliográfica
2. Modelado de diversos dominios de aplicación.
3. Evaluación de los métodos que existen actualmente para el desarrollo de diversos tipos de aplicaciones *web*.
4. Elaboración de un artículo científico para publicación en revistas técnicas
5. Selección de los aspectos positivos de los métodos evaluados para incluirlos en la nueva propuesta metodológica.
6. Formulación del modelo de producto genérico y del modelo de procesos adaptable a múltiples dominios de aplicación y situaciones de proyecto.
7. Redacción y exposición del trabajo de investigación.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

5. SELECCIÓN DE METODOS PARA EL DESARROLLO DE APLICACIONES WEB

Para seleccionar los métodos que fueron evaluados en la presente investigación, con el fin de extraer de éstos sus aspectos positivos, se tomaron en cuenta dos características fundamentales: el dominio de aplicación para el cual se desarrolla la aplicación *web* y la completitud de la documentación disponible.

Por lo tanto, los métodos seleccionados fueron:

- (1) Método para desarrollar S/W (Torrealba, 1998)
- (2) Marco de Referencia basado en componentes para *E-Commerce* (Fingar, 2000) *ACM*
- (3) Desarrollo de sitios *web* instruccionales – Un Enfoque de Ingeniería de *Software*. (Montilva, Sandia y Barrios, 2002)
- (4) *XP (Extreme Programming)* (Beck, 1999; Mc Breen, 2000).

Con el fin de lograr un buen método de evaluación, se diseñó un marco de referencia (Barrios, 2002) para evaluar métodos con el fin de definir los aspectos más relevantes a ser medidos.

Uno de los principios de la Ingeniería de Métodos establece que un método debe tener un propósito específico *el dominio*, el tipo de problema ser solucionado y las condiciones o restricciones de *uso*.

Basado en el marco de referencia para evaluar métodos es necesario considerar 4 aspectos fundamentales para evaluar al mismo: la vista del dominio, la vista de uso, la vista del producto y la vista del proceso. Dentro de estas vistas se consideran a su vez las facetas las cuales describen diferentes características de las vistas. Cada faceta está compuesta por un conjunto de atributos y sus valores correspondientes, los cuales son adaptados para evaluar los métodos de desarrollo de aplicaciones *web*.

La vista de dominio:

Está relacionada con el dominio de aplicación de los métodos que están siendo comparados. La vista del dominio está descrita en una faceta: El *alcance*. Esta faceta contempla todos los atributos relacionados con el área de aplicación, los

aspectos a ser cubiertos: técnicos, gerenciales o aquellos que están relacionados específicamente al dominio de aplicación.

El resultado de la evaluación de los métodos, se observa en la tabla No. 1.

En el atributo área de aplicación se puede observar que los métodos (1) Método para desarrollar S/W (Torrealba, 1998), (2) Marco de Referencia basado en componentes para *E-Commerce* (Fingar, 2000) y (3) Desarrollo de sitios *web* instruccionales – Un Enfoque de Ingeniería de *Software*, han sido diseñados para un dominio de aplicación en particular: Sistemas de información *web*, *e-commerce* y sitios *web* instruccionales, respectivamente. Sin embargo, el método (4) *XP (Extreme Programming)* (Beck, 2000), fue diseñado para desarrollar cualquier tipo de aplicación *web* por lo tanto puede ser utilizado en cualquiera de los dominios de aplicación considerados.

Los métodos (1) y (3) cubren los aspectos técnicos, gerenciales y relacionados con el dominio de aplicación (Sistemas de Información Web y Sitios *web* Instruccionales), lo cual permite guiar de una forma más clara al usuario para adaptar la aplicación a desarrollar al dominio de aplicación. A diferencia de los métodos (2) y (4) que únicamente cubren aspectos técnicos, dejando por fuera los aspectos gerenciales y los relacionados con el dominio de aplicación, lo que dificulta el desarrollo del proyecto y no garantiza que la aplicación se adapte a un dominio de aplicación específico.

Tabla No. 1. Vista del dominio (¿Para qué sirve? ¿para qué es concebido?)

Faceta	Atributos	Valores	(1)	(2)	(3)	(4)
Alcance	Área de aplicación	Hipermedia				✓
		Educativo			✓	✓
		e-commerce	✓	✓		✓
		e-business	✓			✓
		Portales	✓			✓
		Sistemas de Información web	✓			✓
		e-government	✓			✓
	Aspectos	Técnicos	✓	✓	✓	✓
		Gerenciales	✓		✓	
		Inherentes al área de Aplicación.	✓		✓	

Es importante aclarar, que el método (4) fue diseñado para desarrollar aplicaciones *web* en general, por lo tanto no puede cubrir aspectos inherentes a un dominio de aplicación en particular.

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. La Faceta *participación del usuario* especifica los tipos de usuario esperado y su grado de participación en el desarrollo de aplicaciones *web*. El resultado de evaluar estos aspectos se observa en la Tabla No. 2.

Tabla No. 2. Vista de Uso (Objetivos del que los usa)

Faceta	Atributos	Valores	(1)	(2)	(3)	(4)
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 web	✓		✓	✓
		Publicistas				
		Personal directivo de grandes organizaciones	✓			
	Grado de Participación	Total			✓	✓
		Parcial	✓	✓		
		Ninguno				

Los cuatro métodos cubren las fases generales del ciclo de vida de una aplicación *web* (análisis, diseño, desarrollo y evaluación). El mantenimiento, es cubierto únicamente por el método (1) Método para desarrollar S/W (Torrealba, 1998). Sin embargo, el método (4) cubre las etapas de manera ligera y superficial dándole la mayor importancia a las pruebas del *software*.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Cada método está orientado a un uso particular; el (1) Servicios el (2) comercial, el (3) académico, el (4) no posee una orientación definida ya que el mismo no está diseñado para ser usado en el desarrollo de alguna aplicación en específico.

En la faceta de uso, todos los métodos muestran qué hacer pero no todos enseñan cómo hacerlo. En cuanto a la eficiencia de uso, el método (1) es bastante rígido por lo que es poco adaptable y flexible; asimismo, el método (2) no se considera fácil de usar porque no indica cómo hacer las cosas y no está bien documentado.

La vista del producto:

Esta vista evalúa el modelo de producto usado por el método. Dos facetas son seleccionadas para describir el modelo del producto. La descripción del producto, que caracteriza al mismo por su notación, orientación y perspectiva; y la faceta de descripción conceptual, la cual describe los conceptos que son usados por el modelo. La evaluación de esta vista se puede observar en la tabla No. 3.

Tabla No. 3. Vista del Producto (Resultado de aplicar el método)

Faceta	Atributos	Valores	(1)	(2)	(3)	(4)	
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				✓	

Es indispensable que un método disponga de modelo de producto con el fin de garantizar el entendimiento del resultado que se obtendrá por el uso del método. El método (4) no cuenta con un modelo de producto, los métodos (1) (2) y (3) poseen un modelo explícito con notación semiformal (gráficos e imágenes) e informal (texto)

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

que permite caracterizar cada uno de los productos que se generarán al utilizar dicho método.

La vista del proceso:

Esta contiene un conjunto de facetas y atributos que describen el proceso de desarrollo de la aplicación que se desea producir. Esta vista evalúa el modelo de proceso usado por el método, a través de cinco facetas que consideran la orientación y el enfoque, las características y los procesos cubiertos por el modelo. La tabla No. 4 muestra los resultados de esta evaluación.

Uno de los aspectos más importantes a evaluar en esta vista es la faceta del modelo de proceso de desarrollo, específicamente lo relacionado con la orientación. Los modelos de proceso de desarrollo de los métodos (1), (3), y (4) están orientados a la actividad mientras que el del método (2) está orientado al producto. Esta característica demuestra una debilidad que existe en todos los métodos evaluados, debido a que son modelos de procesos que no permiten seleccionar alternativas de acuerdo al dominio de aplicación para el que se está desarrollando y a la situación del proyecto, lo que hace que los métodos se conviertan en un conjunto de pasos secuenciales que no permiten guiar adecuadamente el proceso de desarrollo de una aplicación. Esta falla es más grave aún en el método (4), ya que éste fue diseñado para cualquier tipo de aplicación, más sin embargo, el método no hace diferencia entre cómo se deben cubrir las etapas en los diversos dominios de aplicación.

Todos los métodos evaluados siguen un enfoque evolutivo (adaptativo), lo que se considera una gran fortaleza, debido a que permiten hacer cambios de requerimientos o de diseño a medida que se va evaluando la aplicación con el usuario y/o cliente.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Tabla No. 4. Vista del Proceso (Desarrollo del producto)

Faceta	Atributos	Valores	(1)	(2)	(3)	(4)
Modelo de proceso de desarrollo	Orientación	Orientada al producto		✓		
		Orientada a la actividad	✓			
		Orientada a la decisión			✓	✓
	Enfoque	Ad-hoc				
		Predictivo (cascada)				
Características del proceso	Claridad	Evolutivo (adaptativo)	✓	✓	✓	✓
		Estructura del proceso bien definido	✓	✓	✓	✓
	Estructura del proceso (qué hacer)	Fases/pasos/tareas	✓		✓	
		Modo de ejecución del proceso				
	Iterativo	Iterativo	✓	✓	✓	✓
		Secuencial estricto	✓			
	Incremental	Incremental				
		Definición clara del flujo de trabajo	✓		✓	
	Visibilidad	Progreso visible externamente	✓	✓	✓	✓
		Dependencia tecnológica	Independiente de las herramientas	✓	✓	✓
	Adaptabilidad	Modularidad		✓	✓	✓
		Procesos modificables			✓	✓
Guía	explícita			✓	✓	
	Completa	✓				
Gerencia de proyecto	Planificación y control	Implícita		✓	✓	✓
		Definición del plan del proyecto	✓		✓	✓
		Estimación de recursos	✓		✓	✓
	Equipo de trabajo	Monitoreo y evaluación del progreso	✓		✓	✓
		Tamaño del equipo				
	Individual	Estructura del equipo	✓		✓	✓
		Equipos pequeños (<= 5 personas)			✓	✓
	Equipos Grandes (> 5 personas)	Equipos Grandes (> 5 personas)	✓		✓	✓
		Calidad en la planificación y control	✓		✓	✓
	Aseguramiento de calidad	Calidad del equipo	✓		✓	✓
		Revisión de técnicas de validación y verificación	✓	✓	✓	✓
	Gestión de la configuración	Planificación y ejecución de la gestión de configuración del software	✓			
		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			✓	
		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).				
		Comunicación de objetos distribuidos.	✓	✓		
		Pruebas	Prueba de cada uno de los componentes	✓	✓	
Pruebas de integración			✓	✓		✓
Prueba de la aplicación web.	✓		✓	✓	✓	
Validación y verificación de la aplicación web.	✓			✓	✓	
Proceso de post-desarrollo	Implantación de la aplicación web	✓		✓		
	Soporte de mantenimiento					

Licencia Creative Commons:

Por otra parte, todos los métodos presentan una estructura del proceso bien definida, sin embargo, el (2) y el (4) no dividen sus fases en pasos y tareas, lo que hace que las actividades no se indiquen de manera clara y explícita al grupo de desarrollo. Los métodos (2) y (4) no consideran la gerencia del proyecto, lo que quiere decir que no incluyen dentro del modelo de proceso las actividades de planificación y control, organización del equipo de trabajo, gestión de la configuración del *software* y documentación. El único aspecto que ambos métodos consideran es la revisión de técnicas de validación y verificación debido a que todos ejecutan el proceso en forma iterativa y siguen enfoques evolutivos.

En cuanto a la faceta de cobertura del ciclo de desarrollo, el método (4) cubre, en forma muy ligera y simple, cada una de las etapas, pero no contempla un análisis de objetos del dominio. Sólo realiza una definición informal de requerimientos, sin detenerse mucho, ya que la idea es que a medida que los requerimientos vayan cambiando, dichos cambios sean incorporados rápidamente a la aplicación. El diseño es muy general y ligero y no contempla la etapa de implementación como tal, su mayor importancia radica en las pruebas, las cuales son rigurosas y se realizan de cada uno de los componentes, de integración, de la aplicación y finalmente una validación y verificación de la aplicación.

El método (2) cubre la etapa de análisis y definición de requerimientos; dándole especial importancia al diseño de las arquitecturas y a la comunicación de los objetos distribuidos. Además, cubre la etapa de pruebas de cada uno de los componentes, las de integración, las cuales son las de mayor atención, y las de la aplicación.

Por último, se puede observar que los métodos (1) y (3) son los que cubren completamente las etapas del ciclo de desarrollo (a excepción del mantenimiento que solo es cubierto por el (1)). El método (1) no guía al grupo de desarrollo en lo que a construcción de bases de datos se refiere, tampoco considera mecanismos de seguridad, ni establece como se comunicarán los objetos distribuidos. Esto representa una gran debilidad, ya que el mismo fue diseñado para el desarrollo de *Sistemas de Información Web (SIW)*.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

5. RESULTADOS ESPERADOS

Luego de realizada la evaluación de los métodos, se decidió seleccionar aquellos que están relacionados con el desarrollo de portales (Oliveira, 2000), aplicaciones de comercio electrónico (Oliveira, 2000), negocios inteligentes (Oliveira,2000), sistemas de información *web* (Torrealba, 1998) y gobiernos electrónicos (Valderrama, 2003) debido a que los dominios de aplicación para el cual se desarrollan las mismas presentan características similares, además de ser las aplicaciones más requeridas en la actualidad.

Por tal razón, el principal resultado que se espera lograr con el desarrollo de esta investigación es una propuesta metodológica situacional para desarrollar cualquiera de las aplicaciones *web* mencionadas anteriormente. Esta propuesta será formulada a través de un modelo de proceso y un modelo de producto que sean fácilmente adaptables a dichos dominios de aplicación.

Además, se espera lograr los siguientes resultados parciales:

1. Artículo científico, para publicación en las revistas técnicas de reconocida trayectoria en el área.
2. Descripción de las características que debe poseer un método, basado en los principios de la Ingeniería de Métodos y de la Ingeniería Web.
3. Modelos representativos de los diferentes dominios de aplicación seleccionados.
4. Modelos de productos adaptables a los diferentes dominios de aplicación.
5. Lista de estrategias que permitan orientar al grupo de desarrollo en la toma de decisiones relacionada con el camino a seguir para lograr el producto esperado, considerando la situación de proyecto.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

CAPITULO 2

MARCO TEORICO

En este capítulo, se presenta un breve estudio sobre las aplicaciones *web*, la Ingeniería de Métodos, Ingeniería de Dominio, Reutilización de *software* y el Modelo de Madurez de Capacidad Integrado (*CMMI*). Primeramente, se caracterizan y clasifican las aplicaciones *web*, asimismo, se describen algunos métodos para desarrollar este tipo de aplicaciones. Luego, se presenta una definición de Ingeniería de Métodos, sus principales enfoques y las técnicas para construir métodos nuevos. Además, se define la Ingeniería de Dominio y todo lo relacionado con los dominios de aplicación, las fases de la Ingeniería de Dominio y los métodos más conocidos.

Seguidamente, se explica la definición de componentes, el desarrollo de *software* basado en la Reutilización y las tecnologías de componentes vigentes en el mercado. Finalmente, se describe brevemente los niveles de madurez de una organización basado en el Modelo de Madurez de Capacidad Integrado (*CMMI*).

1. APLICACIONES *WEB*

Una aplicación *web* es definida como una aplicación de *software* que usa interfaz de usuario basada en *web* para proporcionar un conjunto de servicios a sus usuarios (Montilva y Barrios, 2003). Actualmente, una aplicación *web* permite que tanto la generación automática del contenido como el procesamiento de datos interactúen con las aplicaciones de negocios del cliente.

1.1. CARACTERIZACIÓN DE LAS APLICACIONES *WEB*

En general, una aplicación *web* se distingue por las siguientes características (Pressman, 2002):

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

1. **Intensivas de red:** Reside en una red y debe dar servicio a las necesidades de una comunidad diversa de clientes.
2. **Controlada por el contenido:** La función principal de una aplicación *web* es utilizar hipermedia para presentar al usuario el contenido de textos, gráficos, sonido y video.
3. **Evolución continua:** Las aplicaciones *web* se actualizan constantemente sin tener que esperar varios meses para producir una nueva versión. El mantenimiento es transparente al usuario, ya que es el programador quien mantiene la aplicación servidor y la configura.
4. **Inmediatez:** Para su acceso simplemente se requiere un navegador de Internet. Por lo que su acceso puede hacerse desde cualquier plataforma (*linux, windows, MAC, Unix* e incluso sistemas móviles o inalámbricos).
5. **Operatividad en Internet:** Las aplicaciones *web* pueden interactuar con otros elementos de *Internet*, como por ejemplo enviar alertas por *EMAIL* o Servicio de Mensajería corta (SMS), o efectuar transacciones automáticas en otros portales de Internet, etc.

Asimismo, es importante aclarar, que una aplicación *web* (además de las características mencionadas anteriormente) debe contar con la **lógica del negocio**, la cual proporciona los servicios de entrada, almacenamiento, actualización, acceso, recuperación y manipulación de objetos de datos que son almacenados en uno o más servidores.

Las características señaladas anteriormente únicamente diferencian una aplicación *web* de una local, pero no proporcionan una base útil para evaluar la calidad de las mismas. Es por ello que (Olsina, Lafuente y Pastor 2002) presentaron un “árbol de requisitos de calidad” que identifica un conjunto de atributos que debe contener una aplicación *web* de alta calidad (ver figura 1). Este “árbol de requisitos de calidad” está basado en el estándar ISO 9126 (*ISO/IEC 9126, 1991*) el cual provee marco de trabajo para evaluar la calidad de cualquier tipo de *software*.

Es importante mencionar que estas características no son consideradas por muchos desarrolladores, lo que origina un caos en el proceso de desarrollo de aplicaciones *web*, por no garantizarse la calidad del producto final y olvidarse de que

las aplicaciones *web* son sistemas que evolucionan rápidamente en su funcionalidad, alcance, contenido y uso. Por lo cual, para obtener información confiable de las mismas es necesario integrarlas con otras tecnologías, tales como bases de datos heterogéneas, sistemas de información legados y objetos distribuidos, las cuales le proporcionan a las aplicaciones *web* sus características propias.

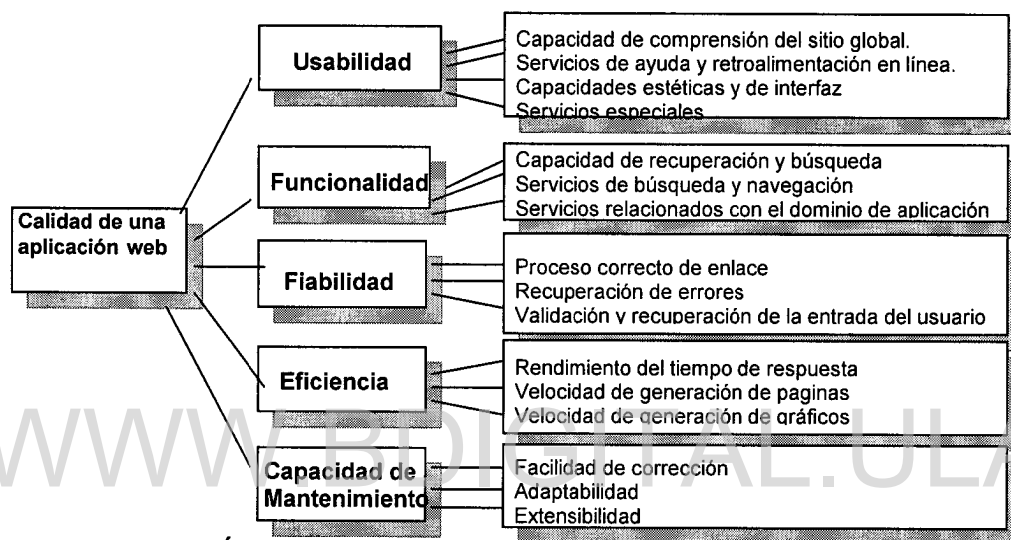


Figura 1. Árbol de Requisitos de Calidad (Fuente: Ingeniería de *Software* (Pressman, 2002))

Entre las tecnologías más importantes se mencionan:

- **Desarrollo basado en componentes:** Consiste en ensamblar un conjunto de componentes de *software* preconstruido y estandarizado siguiendo los patrones de un estilo arquitectónico específico para algún dominio de aplicación (Presman, 2001). En la actualidad existen tres estándares para la infraestructura: *CORBA*, *.NET* y *J2EE*. Estos estándares son los que proporcionan una infraestructura que permite a los diseñadores emplear y personalizar componentes de terceras partes garantizándoles así la comunicación entre ellos y con servicios a nivel de sistemas.
- **Mecanismos de Seguridad:** Mediante la infraestructura de red se proporciona una gran variedad de medidas de seguridad, tales como encriptación, cortafuegos y otras, con el fin de impedir los accesos sin autorización de personal interno de las organizaciones o de *hackers*.

- **Estándares de Internet:** El estándar dominante había sido el Lenguaje de Marcado de Hipertexto (*HTML*). Sin embargo, a medida que las aplicaciones crecen en tamaño y complejidad, se adoptó el estándar del Lenguaje de Marcado eXtensible (*XML*).

1.2. TIPOS DE APLICACIONES WEB

En nuestra investigación, se ha decidido categorizar las aplicaciones *web* de acuerdo al dominio de aplicación para el que han sido desarrolladas, ya que cada aplicación *web* tiene un propósito definido, lo cual permite diferenciarlas entre sí, es decir, cada una de estas aplicaciones ofrece servicios de acuerdo a las necesidades de los usuarios propios de cada dominio. Por tanto, una aplicación *web* puede ofrecer desde servicios informativos hasta un centro de compras electrónicas, lo que quiere decir que su complejidad puede variar ampliamente desde servicios informativos de pequeña escala a aplicaciones empresariales de gran escala que soportan varios procesos del negocio. Por tal razón, Ginige y Murugesan (2001), clasifica las aplicaciones *web* según el grado de complejidad, tal como se muestra en la tabla No. 5.

Tabla No. 5. Categorías de Aplicaciones Web

Categoría	Ejemplos
Informativas	Periódicos en línea, catálogos de productos, boletines de noticias (<i>newsletters</i>), manuales de servicio, avisos clasificados en línea, libros electrónicos en línea.
Interactivas (Información provista por el usuario o accesos personalizados)	Formularios de registro, presentación de información personalizada, juegos en línea.
Transaccionales	Compras electrónicas, ordenes de mercancía y servicios, banca en línea.
Workflow	Sistemas de planificación y programación en línea, Gestión de inventario, Monitoreo del estado.
Ambientes de trabajo colaborativo	Sistemas de autoría distribuida, herramientas de diseño colaborativo.
Mercados y comunidades en línea	Grupos de conversación (<i>chat</i>), sistemas para recomendar productos o servicios, mercados en línea, subastas en línea.
Portales Web	Centros de compras electrónicas, intermediarios en línea.

Fuente: Ginige y Murugesan, 2001. *Web Engineering: An Introduction*

De acuerdo a nuestra categorización, las aplicaciones *web* se dividen en:

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

1. **Informativas:** Páginas que presentan un contenido de solo lectura con navegación y enlaces simples (Pressman, 2002). Un ejemplo de ello lo constituyen las aplicaciones hipermedias, las cuales se definen formalmente como un grafo dirigido $G(N,E)$, en el que N es un conjunto de nodos de información y E es un conjunto de arcos denominados enlaces , cada uno de los cuales conecta dos nodos de N (Montilva 1996).
2. **Instruccionales:** Ambiente de enseñanza – aprendizaje implementado y distribuido a través de la tecnología *Web* para soportar cursos en línea (Montilva, Sandía y Barrios, 2001). Estos son un tipo especial de *software* que es creado y mantenido usando editores *web* de propósito general o sistemas de gestión de cursos especializados.
3. **Portales:** Sitios *web* que ofrecen al usuario, de forma fácil e integrada, el acceso a una serie de recursos y servicios *web* que pueden estar fuera del dominio de aplicación del portal. Entre los cuales se pueden mencionar los buscadores, foros, comercio electrónico, etc. (Oliveira, 2000).
4. **Comercio electrónico (*e-commerce*):** Portales que se desarrollan con la finalidad de que el usuario pueda efectuar exclusivamente transacciones de compra, venta, remate, subasta u otra; que involucran pago a través de soluciones electrónicas o fuera de línea entre un proveedor y un consumidor (Oliveira, 2000).
5. **Negocios inteligentes (*e-business*):** Son portales, que además de permitir realizar transacciones de compra, venta; tienen la opción de hacer uso de servicio de valor agregado tales como: preguntas y respuestas, opinión/comentarios, ideas, foros/grupos de discusión, diálogos con otros consumidores, referencias de los consumidores (Oliveira, 2000). Los portales de *e-business* se clasifican en tres grandes grupos:
 - **"B2C", "*Business-to-Customer*":** Cuando el portal permite realizar transacciones monetarias directamente (sin intermediarios) con el consumidor, por ejemplo: Dell, Amazon.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

- **“B2B”.** **“Business-to-Business”:** Cuando el portal solo permite interactuar con los mercados de ofertantes y de demandantes, sin ocurrir entre ellos transacciones del tipo monetario.
 - **“B2E”.** **“Business-to-Employee”:** Cuando el portal permite interactuar con los empleados de una organización con el fin de recibir orientación a través de las ayudas de escritorio (*helpdesk*), crear espacios de trabajo colaborativo, realizar operaciones internas, o administrar las finanzas, entre otras.
- 6. Sistemas de Información web (SIW):** Los mismos no son más que sistemas de información tradicionales que emplean las tecnologías de *Internet* y las capacidades de un navegador (visualizador de paginas *web*), lo cual le permite integrar diversos sistemas de *software* heterogéneos (*DataWarehouse*, *e-Business*, *sitios web*, *groupware*, sistemas *ERP* (*Enterprise, Resource Planning*), Sistemas de información Geográfica) en una única interfaz de usuario basada en el *web* (Torrealba, 1998).
- 7. Gobiernos electrónicos (e-government):** aplicaciones *web* que pueden ir desde sitios *web* con poca interacción hasta un único punto de contacto para todos los servicios (ventanillas únicas) que son ofrecidos a un gran número de usuarios (ciudadanos, negocios y empleados del gobierno) (Valderrama, 2003).

La infraestructura tecnológica en la que está sustentada se basa en estándares abiertos, que pueden incluir una gran variedad de tecnologías, entre las cuales se mencionan: *web sites*, *e-mail newsletter*, sistemas de información geográfica (GIS), realidad virtual, herramientas para toma de decisiones, tecnología para participación pública, portales de planificación en línea, sistemas de gestión de contenidos, sistemas de zonificación y permisología, aplicaciones para dispositivos *handhelds*.

Los portales de *e-government* se desarrollan en cuatro escenarios diferentes:

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

- **Gobierno a Ciudadano (G2C):** Proporciona servicios al ciudadano como acceso a documentos estáticos, sistemas transaccionales para pago de impuestos, registro de vehículo y solicitud de permisos.
- **Gobierno a Empleado (G2E):** Portales empresariales que permiten compartir conocimientos y aplicaciones para automatizar los procesos internos y dar soporte a las operaciones realizadas por el gobierno.
- **Gobierno a Gobierno (G2G):** Permite compartir datos y transacciones con otras organizaciones gubernamentales para incrementar la eficiencia operacional.
- **Gobierno a Negocio (G2B):** Proporciona acceso a portales para interactuar con compañías, como por ejemplo los proveedores.

1.3. MÉTODOS PARA EL DESARROLLO DE APLICACIONES *WEB*

A continuación, se presenta una breve descripción de los métodos para el desarrollo de aplicaciones *web* que existen en la actualidad. Estos métodos se seleccionaron tomando en cuenta la categorización de aplicaciones *web* realizada para esta investigación y descrita en la sección anterior.

1.3.1. Métodos Ágiles para el Desarrollo de Aplicaciones

Los métodos ágiles constituyen una solución a la medida para el desarrollo de proyectos pequeños, aportando así, una elevada simplificación de actividades, y asegurando la calidad del producto a través de la implementación de prácticas esenciales. Estos métodos dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del *software* con iteraciones muy cortas. Se consideran muy efectivos en el desarrollo de proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo, pero manteniendo una alta calidad.

El ciclo de desarrollo de estos métodos consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar

Licencia Creative Commons:

2. El programador estima el esfuerzo necesario para su implementación
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio..

Entre los métodos ágiles más conocidos se encuentran:

XP (Extreme Programming) (Beck, 1999; Mc Breen, 2000): La programación extrema toma las pruebas del *software* como base para el desarrollo, donde cada programador que escribe código también escribe los casos de prueba. Estos forman parte del proceso continuo de generación de código y se integra continuamente, lo que garantiza una plataforma estable para el futuro desarrollo. Sobre dicha plataforma se genera un proceso de diseño evolutivo que es la base del sistema y que se enriquece con cada iteración.

Este método surge por la necesidad de desarrollar aplicaciones y sitios *web* que proporcionen la explotación de otros mercados y servicios como el comercio electrónico, la enseñanza virtual, etc. El mismo puede ser utilizado para aplicaciones que no se encuentran en la *web*.

Open Source (Raymond's y Fogel's, 2000): *Open Source* apuesta por la distribución de trabajo entre diferentes equipos, al igual que ocurre con la mayoría de los procesos adaptativos. La mayoría de los proyectos cuentan con supervisores de código, quienes son las únicas personas autorizadas para realizar un cambio en el repositorio del código fuente.

Su principal ventaja es que la depuración es altamente paralelizable, aunque un gran número de personas puedan verse involucradas. Este método, al igual que el anterior, surge para tratar con requisitos desconocidos o variables, lo cual ocurre con el desarrollo de aplicaciones *web*, que requieren colocarse, en el menor tiempo posible, en producción sin conocer completamente los requerimientos de los usuarios.

1.3.2. Métodos para el Desarrollo de Sistemas de Información Web (SIW)

Metodología para desarrollar Sistemas de Información Web (MIDAS)

(Cáceres y Marcos, 2001): Es una "metodología" ágil y genérica para el desarrollo

(análisis y diseño) de Sistemas de Información *Web* (SIW). Esta metodología posee las mismas características de los métodos ágiles mencionados anteriormente, con la diferencia de que la misma tiene un propósito definido: el desarrollo de Sistemas de Información *Web* (SIW). Por tal razón, al igual que los métodos ágiles, se caracteriza por la priorización de requisitos a través de secuencia de fases o incrementos, por ser un proceso iterativo que va generando pequeñas versiones, no forzar la realización de técnicas y actividades innecesarias, por realizar pruebas por producto generado en cada iteración, producir productos por pares de personas, incorporar al cliente al equipo de trabajo y utilizar lenguajes estándares.

Método para desarrollar SIW (Torrealba, 1998): Este método tiene los siguientes objetivos: guiar al grupo de profesionales que participan en el desarrollo de un *SIW*, contribuir a la planificación de proyecto mediante la definición de las fases, pasos y tareas requeridas para desarrollar un *SIW*, asegurar la calidad del *SIW* producido.

El método está basado en los modelos de procesos y conceptos de la ingeniería del *software*, cubriendo todo el ciclo de desarrollo de *SIW*. Además, el método es independiente de las herramientas existentes para el desarrollo de *SIW*, emplea técnicas de análisis y diseño orientado a objetos las cuales permiten un modelado de la estructura y contenido del *SIW* mucho más natural, elegante y de fácil comprensión.

1.3.3. Métodos para el Desarrollo de Aplicaciones Hipermedia

Aplicando modelos de procesos de *software* al desarrollo de aplicaciones hipermedia (Montilva, 1996): El método tiene por objetivos guiar al grupo de profesionales que participan en el desarrollo de un hiperdocumento, contribuir a la planificación del proyecto mediante la definición de las fases, pasos y tareas requeridas para desarrollar un hiperdocumento y asegurar la calidad del hiperdocumento producido.

El ciclo de desarrollo es evolutivo, ya que se repite varias veces, en base a la evaluación de hiperdocumento, hasta alcanzar una alta calidad del producto y un alto

Licencia Creative Commons.

grado de satisfacción del usuario. El método asume la existencia de tres actores: el cliente, el grupo de desarrollo y los usuarios.

Hypertext Design Model (HDM) (Garzotto, Paolini y Schawabe, 1993):

Este método introdujo muchos conceptos fundamentales nuevos en el mundo del desarrollo de aplicaciones hipermedia. *HDM* permitió romper la brecha para representar la misma información en varias formas alternadas, pero no hizo el salto hacia las diferentes clases de usuario. Con este método fueron definidos los conceptos de entidad y tipo de entidad y estructura de acceso.

Relationship Management Methodology (RMM) (Isakowitz, Stohr y Balasubramanian 1995): Está basado en *HDM*, anteriormente señalado. Contiene un modelo de diseño. Introduce la noción de *slice*, lo cual permite limitar la vista sobre la información acerca de entidades "ricas" con muchos atributos. Los cinco pasos del método (diseño entidad-relación, diseño de *slices*, diseño de la navegación, diseño de la conversión de protocolos y diseño de la interfaz de usuario) están guiados por los datos y cubre completamente el ciclo de vida de un sitio web típico.

1.3.4. Métodos para el Desarrollo de Sitios Web Instruccionales

Desarrollo de sitios web instruccionales – Un Enfoque de Ingeniería de Software (Montilva, Sandia y Barrios, 2002): Es un método de ingeniería de *software* que ayuda a los diseñadores de cursos y profesores a desarrollar sitios web instruccionales de alta calidad. El diseño del método está basado en la integración de principios bien conocidos, conceptos y modelos de procesos provenientes de diseño instruccional, Ingeniería de Métodos e ingeniería de *software* orientada a objetos. El método dirige los procesos gerenciales requeridos para planificar, organizar y controlar el proyecto, así como las actividades técnicas e instruccionales involucradas en el desarrollo de sitios web instruccionales. El método está compuesto por un modelo de producto que describe un marco de trabajo conceptual necesario para el modelo de procesos, el que a su vez describe las fases y los pasos que se deben ejecutar.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

1.3.5. Métodos para el desarrollo de aplicaciones de comercio electrónico (e-Commerce)

Marco de Referencia basado en componentes para E-Commerce (Fingar, 2000) ACM: Este marco de referencia (“*framework*”) está basado en la ingeniería del *software* basada en componentes, la cual tiene por objetivo producir componentes de alta calidad. El grupo de desarrollo debe buscar componentes prefabricados para adaptarlos a los procesos de negocio que han sido modelados y luego ensamblarlos a la aplicación. Los componentes *e-commerce* proporcionan funcionalidades interempresariales para *e-commerce* (negociación, mediación, gerencia de flujos de trabajo “*workflow*” interempresariales y notificaciones de eventos); por tanto la principal tarea del grupo de desarrollo consiste en adecuar y extender el marco de referencia (“*framework*”) para incorporar las reglas del negocio y procesos de una compañía en particular. Debido a esto, el grupo de desarrollo debe concentrar sus esfuerzos en el conocimiento y carácter único de la compañía para sacarle punta a sus ventajas competitivas.

El *framework* está basado en el método estrategia código (*strategy-to-code method*) para desarrollar los sistemas de *e-commerce*, fusionando de esta manera, la arquitectura del negocio con la arquitectura de la tecnología. La estrategia del negocio involucra a los expertos del dominio quienes definen las iniciativas a ser procesadas basado en el análisis de las limitaciones, oportunidades y características de la compañía. Las estrategias de negocio determinan qué problema será resuelto, generando requerimientos, metas y restricciones. Las aplicaciones basadas en componentes implementan los procesos de negocio nuevamente diseñados apoyados en la infraestructura tecnológica. Todo el proceso es guiado por un enfoque arquitectural que genera el desarrollo rápido de soluciones de negocio a través del Reutilización de todos los elementos en un gran repositorio de modelos y componentes de *software*.

1.3.6. Metodologías para el Desarrollo de Aplicaciones Web:

Método WATCH (Montilva, 2000): El método WATCH es el resultado de la integración de varios modelos de procesos de la ingeniería del *software*, incluyendo

los modelos orientados a objetos, Reutilización y modelo de prototipos. El modelo apunta hacia las aplicaciones de *software* de negocios, tales como sistemas de información de negocios, sistemas de información basados en *web* y aplicaciones de comercio electrónico.

La metáfora del reloj es usada para diseñar un método que guía al equipo de desarrollo en la complejidad de gerenciar y desarrollar aplicaciones de negocios en ambientes de desarrollo pequeños. Esta caracterizado por emplear el Lenguaje de Modelado Unificado (UML) como lenguaje de modelado.

Está estructurado en 4 grupos de procesos: Ingeniería de Métodos, procesos gerenciales, procesos de desarrollo y procesos de post-desarrollo, que a su vez se dividen en fases.

Un marco de trabajo para la Ingeniería Web (IWEB) (Pressman, 2002):

El marco de trabajo está basado en modelos de procesos incremental e iterativo con la finalidad de poder elaborar aplicaciones *web* muy rápidamente. El marco de trabajo sugiere que el equipo de desarrollo esté conformado por personas técnicas y no técnicas (redactores publicitarios, diseñadores gráficos, etc.).

El proceso de Ingeniería Web (*IWeb*) comienza con la formulación, actividad que identifica las metas y los objetivos de las aplicaciones *web* y establece el ámbito del primer incremento.

La planificación estima el coste global del proyecto, evalúa los riesgos asociados con el esfuerzo del desarrollo y define una planificación del desarrollo bien granulada para el incremento final de la aplicación *web*. El análisis establece los requisitos técnicos para la aplicación *web* e identifica los elementos del contenido que se van a incorporar. Además, se definen los requisitos del diseño gráfico (la estética).

La actividad de ingeniería incorpora dos tareas paralelas: el diseño del contenido y la producción, las cuales son llevadas por personas no técnicas del grupo de desarrollo. La generación de páginas, es una actividad de construcción que hace uso de las herramientas automatizadas (editores *web*) para la creación de la aplicación *web*. Durante esta actividad también se lleva a cabo la integración con el *software* intermedio (*middleware*) de componentes (CORBA, J2EE, .NET). Las

pruebas ayudan a asegurar que la aplicación *web* funcionara correctamente en diferentes entornos. Cada incremento producido en el proceso *lweb* es revisado durante la actividad de evaluación del cliente.

2. INGENIERA DE METODOS

La Ingeniería de Métodos es una disciplina donde los mismos métodos son el destino del ingeniero, como por ejemplo el explorar técnicas de ingeniería para construir métodos. De acuerdo a Brikkemper (Brikkemper, 1995), la Ingeniería de Métodos es “una disciplina para diseñar, construir y adaptar métodos, técnicas y herramientas para el desarrollo de sistemas de *software*”.

La Ingeniería de Métodos representa el esfuerzo por mejorar la utilidad de los métodos para desarrollar sistemas, con el fin de crear un marco de trabajo de adaptación, a través del cual métodos situacionales son creados para resolver situaciones problemáticas específicas.

En esta investigación, la Ingeniería de Métodos facilitará la construcción de un método situacional para el desarrollo de aplicaciones *web*, a través del Reutilización y ensamblaje de fragmentos de métodos ya existentes, los cuales permiten resolver una situación problemática específica. Es importante aclarar, que un fragmento de método está conformado por un fragmento de producto y un fragmento de proceso, los cuales son dependientes entre si, ya que un fragmento de producto requiere de un fragmento de procesos.

Asimismo, un fragmento de método puede ser representado en tres (3) niveles de abstracción, los cuales se muestran en la figura 2. El Nivel superior está constituido por los meta-modelos. Un meta-modelo es un modelo conceptual de un método de desarrollo (Brinkemper, 1990), es decir, que un meta-modelo captura información acerca de los conceptos, formas de representación y uso de un método. Un meta-modelo está compuesto de un meta-modelo de producto y un meta-modelo de proceso.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Un meta-modelo de producto representa el conjunto de conceptos genéricos que sirven de base a la definición de modelos de producto. Un meta-modelo de proceso describe formalmente un conjunto de conceptos genéricos (y sus interrelaciones) los cuales sirven de base para la definición de modelos de procesos (Barrios, 2002).

El siguiente nivel está constituido por los modelos. Un modelo de producto representa el conjunto de conceptos que se pueden utilizar para construir un producto o sistema (Barrios, 2002), y es la instanciación de un meta-modelo de producto. un modelo de proceso indica las actividades a ejecutar durante el desarrollo de un proyecto, y es la instanciación de un meta-modelo de procesos (Kraiem, Bourguiba y Selmi, 2000).

El nivel inferior está constituido por los productos y procesos. El producto es la instanciación del modelo de producto y el proceso es la instanciación del modelo de procesos para un proyecto particular (Kraiem, Bourguiba y Selmi, 2000).

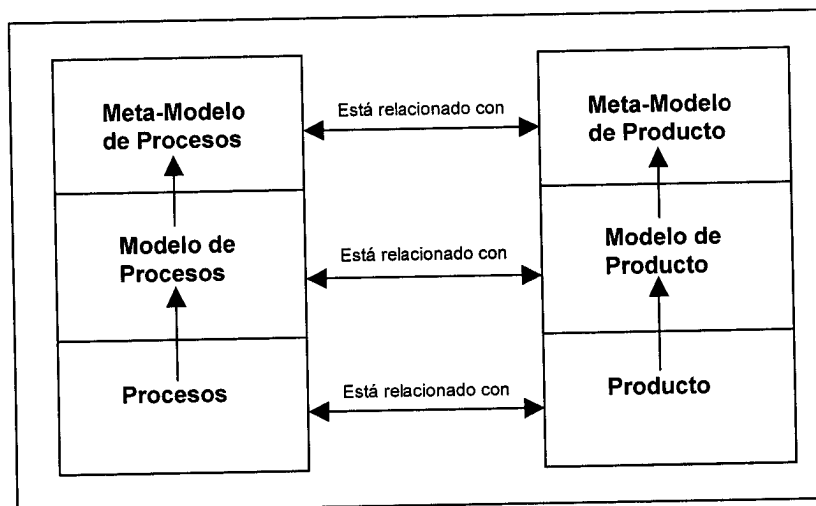


Figura 2. Niveles de Abstracción de un Fragmento de Método. Adaptación de: *Situational Method for Information System Project* (Kraiem, Bourguiba y Selmi, 2000).

2.1. ENFOQUES DE LA INGENIERÍA DE MÉTODOS

Dentro de la Ingeniería de Métodos existen dos (2) enfoques predominantes: la meta-modelización y la orientación situacional (Kraiem, Bourguiba y Selmi, 2000).

2.1.1. Meta-modelización

La meta-modelización consiste en la representación de modelos de producto y modelo de procesos como instanciación del mismo meta-modelo.

a) Meta-modelo de Producto:

Un meta-modelo de producto es un conjunto de conceptos independientes de cualquier metodología. Como se explicó anteriormente, un meta-modelo de producto representa el conjunto de conceptos genéricos que sirven de base a la definición de modelos de producto.

El modelo de producto es la instanciación de un meta-modelo de producto. El mismo está compuesto de un conjunto de conceptos los cuales tienen propiedades que son relacionados a través de asociaciones.

Muchos investigadores han propuesto varias técnicas de meta-modelado para modelar y representar métodos, una de ellas es el modelo entidad-relación. El modelo entidad-relación incluye versiones extendidas tal como el modelo de objeto-propiedades rol (modelo OPR) el cual es ampliamente usado para describir partes de productos de un método (Saeki, 2000).

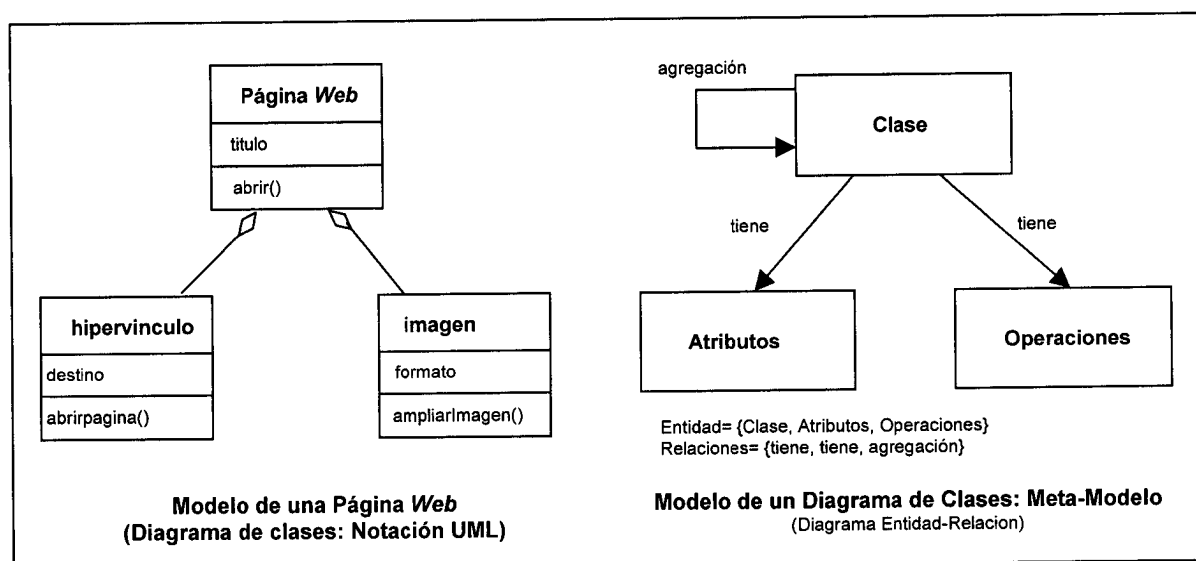


Figura 3. Modelo y Meta-Modelo de un Diagrama de Clases. Adaptación de: (Software Specification & Design Methods And Method Engineering (Saeki, 2000).

En la figura 3, una parte del diagrama de clases es especificado en forma de diagrama de entidad-relación. Los rectángulos y las flechas expresan las entidades y relaciones.

b) Meta-modelo de procesos

Un meta-modelo de procesos es un conjunto de actividades independientes de cualquier metodología (Kraiem, Bourguiba y Selmi, 2000). Un meta-modelo de procesos debe proporcionar mecanismos para: la argumentación, definición de criterios de selección de pasos, adaptación del proceso a situaciones particulares del modelado y la utilización de heurísticas (Barrios, 2002).

Un modelo de procesos es la instanciación de un meta-modelo de procesos, ya que, como se mencionó anteriormente, un meta-modelo de procesos describe formalmente el conjunto de conceptos genéricos (y sus interrelaciones) los cuales sirven de base para la definición de los modelos de procesos (Ver figura 4).

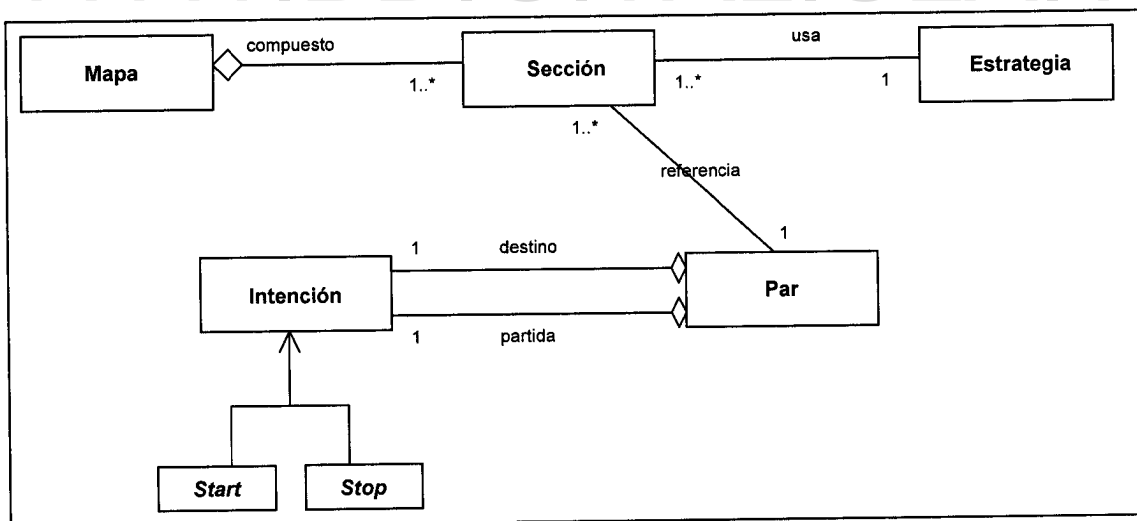


Figura 4. Meta-Modelo para Construir Mapas de Procesos. Adaptación de: (Barrios, 2002).

Un mapa muestra un conjunto de intenciones y sus correspondientes estrategias de ejecución, las cuales pueden ser seleccionadas, durante el proceso de modelado para construir un producto (Barrios, 2002). El mapa proporciona una vista estratégica de los procesos que pueden ser ejecutados.

Un mapa es representado por un grafo dirigido etiquetado en el cual los nodos son las intenciones y los arcos entre intenciones son las estrategias (Kraiem, Bourguiba y Selmi, 2000). El orden de las intenciones y las estrategias no es determinístico, ya que su selección depende de ciertos factores que deben evaluarse. Los arcos en los grafos son dirigidos y muestran cuales intenciones pueden ejecutarse.

Se utilizan, además, un conjunto de directivas que guían el proceso de modelado. Para cada sección del mapa hay directivas asociadas: una DEI (Directiva de Ejecución de Intención), una DSI (Directiva de Selección de Intención) y cero o una DSE (Directiva de Selección de Estrategia). Las DEI indican la manera de ejecutar la intención. La DSI indica la próxima intención que puede ser ejecutada. La DSE ayuda en la selección de la estrategia más apropiada cuando hay varias intenciones, una intención fuente y una destino (Barrios, 2002).

Dado que, para el desarrollo de la presente investigación, necesitamos un modelo de procesos que permita orientar el razonamiento asociado a la ejecución de las actividades de modelado, se propone utilizar un meta-modelo de procesos orientado a la decisión. Un meta-modelo de procesos orientado a la decisión es aquel que permite expresar procesos no secuenciales, repetitivos y complejos, donde múltiples estrategias de solución pueden aplicarse, en una diversidad de situaciones de modelado, donde múltiples procesos de modelado pueden seguirse (Barrios, 2002).

El meta-modelo, está basado en cuatro (4) elementos fundamentales: situación, intención, estrategia y decisión (Barrios, 2002). Una situación puede ser un estado particular del producto en construcción, un conjunto de aspectos que describen el ambiente de modelado. Una intención es un objetivo que el modelador desea satisfacer en un momento dado del proceso, una estrategia es la manera como se ejecuta una intención y una decisión es ejecutar una intención mediante una estrategia.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

El modelo de proceso, instancia del meta-modelo de la figura 4, se representa gráficamente por mapas y directivas (ver figura 5).

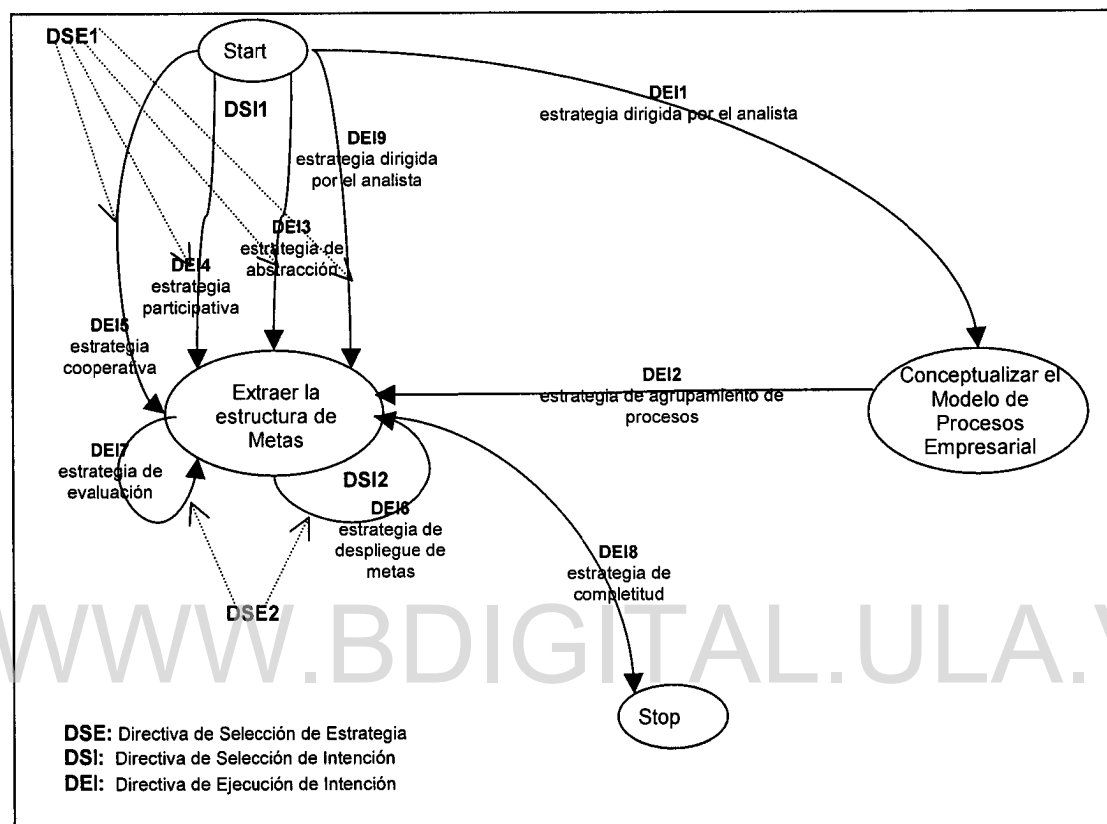


Figura 5. Ejemplo de un Mapa de Procesos. Adaptación de: Modelado de Procesos (Barrios, 2002).

2.1.2. Orientación situacional

El enfoque situacional encuentra su justificación en el análisis de las prácticas de los métodos, éstas muestran que un método nunca es seguido al pie de la letra, sino que, por lo contrario, es adaptado a la situación particular de cada proyecto. Un método situacional es hecho de acuerdo a situaciones particulares de un proyecto por capas de fragmentos de métodos predefinidos, validados y accesibles en una base de métodos.

El enfoque situacional recomienda la construcción de un método por ensamblaje de bloques llamados "fragmentos de métodos", los cuales son tomados de métodos existentes (ver figura 6). Estos fragmentos de métodos son ensamblados

Licencia Creative Commons.

en un “método situacional” que permite resolver una situación de proyecto particular en un dominio dado (Kraiem, Bourguiba y Selmi, 2000).

2.2. TÉCNICAS PARA LA CONSTRUCCIÓN DE MÉTODOS SITUACIONALES

El enfoque del método situacional propone tres técnicas para la construcción de métodos situacionales: estrategia basada en el ensamblaje, estrategia basada en la extensión y estrategia basada en el paradigma (Ralité, Deneckere y Rolland, 2002).

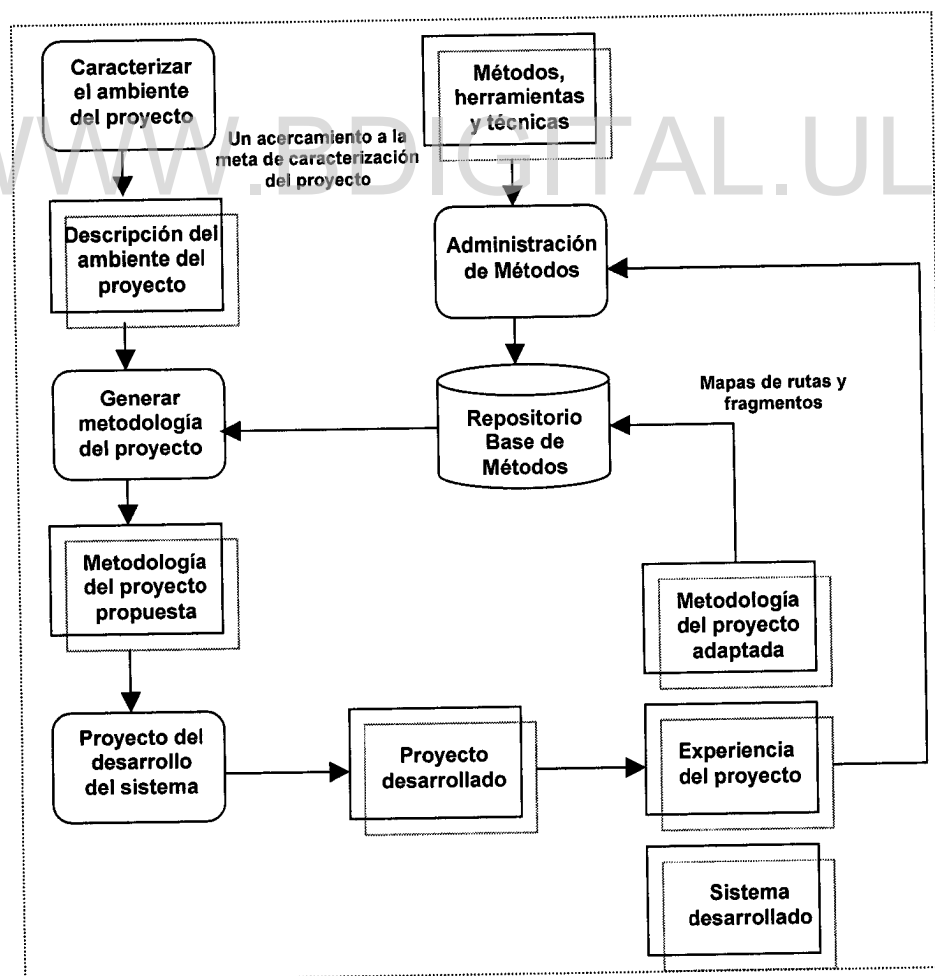


Figura 6. El proceso de la Ingeniería de Métodos Situacional. Fuente: *Situational Method for Information System Project* (Kraiem, Bourguiba y Selmi, 2000). Adaptación: (Mendoza, 2004).

2.2.1. Ingeniería de Métodos basada en el ensamblaje.

Esta técnica se enfoca en construir un método “sobre el vuelo” con el fin de resolver tan bien como sea posible la situación del proyecto que se está desarrollando. La misma consiste en seleccionar los componentes (*method chunks*) de métodos existentes para satisfacer algún requerimiento situacional y su ensamblaje (Ralité, Deneckere y Rolland, 2002).

Esta técnica es dirigida por requerimientos, lo que implica que el ingeniero de métodos debe comenzar por extraer los requerimientos para el método. Luego, un *method chunk* debe corresponder con ese requerimiento, el cual puede ser recuperado de una base de fragmentos de métodos. Finalmente, el *chunk* seleccionado debe ser ensamblado para conformar un nuevo método o para ampliar uno que ya existe. Como consecuencia de esto, las tres intenciones a ejecutarse el proceso de Ingeniería de Métodos basada en el ensamblaje son: Especificación del requerimiento del método, selección del *method chunk* y Ensamblaje de los *method chunks* (ver fig. 7).

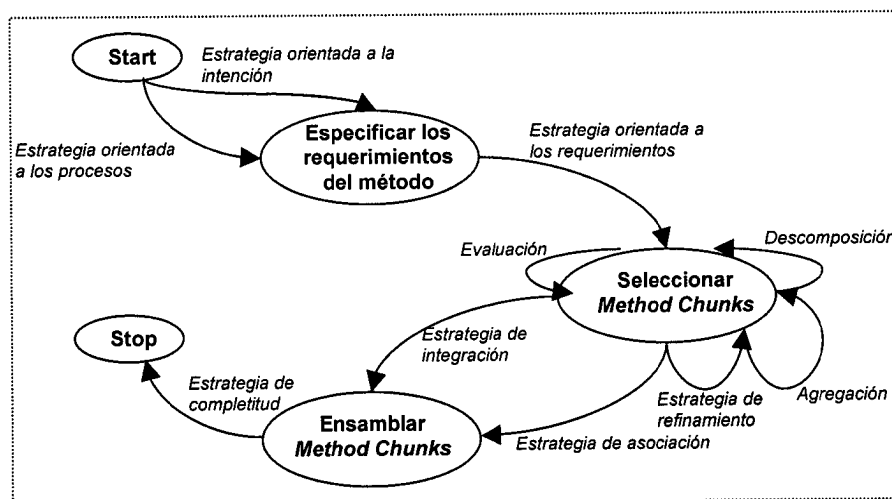


Figura 7. Modelo de Procesos Basado en el Ensamblaje para Ingeniería de Métodos Situacional.
Fuente: *Towards a Generis Model for Situational Method Engineering* (Ralité, Deneckere y Rolland, 2002)

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

2.2.2. Ingeniería de Métodos basada en la Extensión

Esta técnica consiste en adaptar localmente un método para la contingencia de un proyecto que se esté desarrollando. La misma guía al ingeniero de métodos a proporcionar patrones de extensión que ayudan a identificar situaciones típicas de la extensión y a dar orientación para ejecutar la extensión requerida. (ver fig. 8).

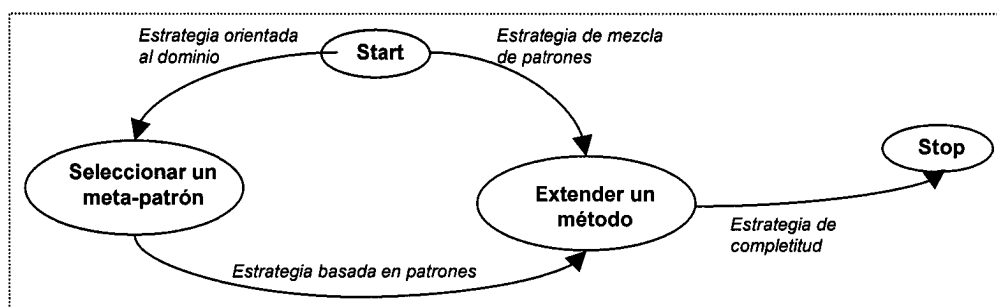


Figura 8. Modelo de Procesos Basado en Extensión para Ingeniería de Métodos Situacional. Fuente: *Towards a Generis Model for Situational Method Engineering* (Ralyté, Deneckère y Rolland, 2002)

Esta técnica puede ser ejecutada de dos maneras distintas: a través de la estrategia de mezcla de patrones o el uso de algún conocimiento genérico relacionado con el dominio para el cual la extensión es hecha a través de la selección de un meta-patrón y de la extensión de un método con la estrategia basada en patrón.

2.2.3. Ingeniería de Métodos Basada en el Paradigma:

Esta técnica utiliza el enfoque del meta-modelado de la Ingeniería de Métodos. Esta es la técnica más genérica para construir un método. La hipótesis de este método es que un nuevo método es obtenido de abstraer de un modelo existente o por instanciar un meta-modelo. El meta-modelado es conocido como una técnica para capturar conocimiento acerca de métodos. Uno de los resultados obtenidos es la definición de que un método está compuesto por un modelo de producto y un modelo de procesos. Por lo tanto, la construcción del método está centrada en estos dos modelos (ver figura 9).

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

La construcción del modelo de producto depende de las metas de la Ingeniería de Métodos. Por ejemplo, la meta de la Ingeniería de Métodos podría ser construir un método:

- Por subir o bajar el nivel de un modelo dado
- Por instanciar un meta-modelo seleccionado
- Por adaptar un meta-modelo a una circunstancia específica
- Por adaptar un modelo,

Cada uno de estos casos define una estrategia para construir un modelo de producto: abstracción, instanciación, adaptación y utilización. Cada una de estas es soportada por una directiva. De acuerdo a estas directivas, la construcción del modelo de producto consiste en definir diferentes elementos del modelo de producto tales como objetos, enlaces y propiedades, lo cual comienza con la abstracción de algunos elementos del modelo del paradigma. Luego, el modelo es refinado a través de las estrategias de generalización, especialización, agregación, descomposición y transformación.

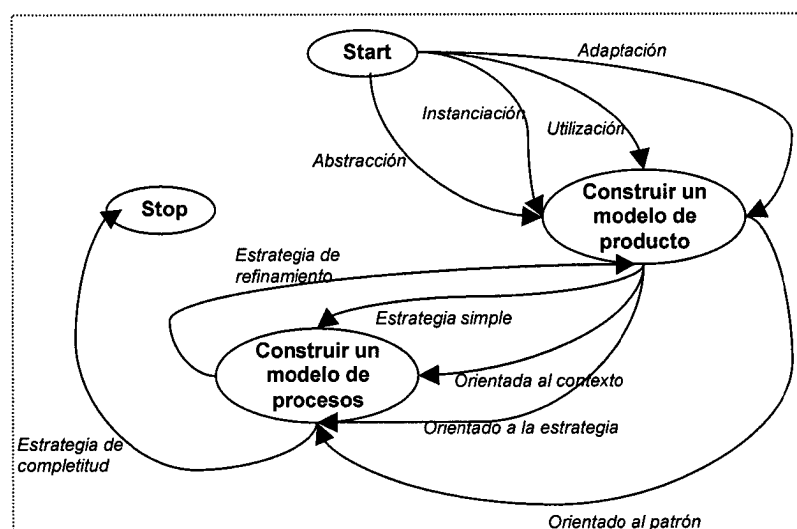


Figura 9. Modelo de Procesos Basado en Paradigma para Ingeniería de Métodos Situacional. Fuente: *Towards a Generis Model for Situational Method Engineering* (Ralyté, Deneckère y Rolland, 2002)

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

El modelo de procesos debe estar conforme con el modelo de producto. Un modelo de procesos puede tomar múltiples formas diferentes. Este puede ser una directiva informal simple, un conjunto de actividades ordenadas para ejecutar algo, un conjunto de patrones de procesos a ser seguidos por una directiva de multiprocesos combinando diferentes alternativas de trabajar. Estos cuatro casos son representados en el modelo de procesos basado en el paradigma por cuatro estrategias: simple, orientado al contexto, orientado por patrones y orientado por la estrategia.

En la construcción del método situacional para desarrollar aplicaciones *web* fue necesario utilizar una combinación de las tres técnicas para la construcción de métodos situacionales de la Ingeniería de Métodos: el ensamblaje de fragmentos de métodos, la Ingeniería de Métodos basada en la extensión y la Ingeniería de Métodos basada en el enfoque basado en el paradigma, lo que garantiza que nuestro método cumple con los principios de la Ingeniería de Métodos y posee un enfoque orientado al contexto.

3. INGENIERIA DE DOMINIO

El propósito fundamental de la Ingeniería de Dominio consiste en proporcionar Reutilización entre aplicaciones similares. La Ingeniería de Dominio es un proceso sistemático que proporciona una arquitectura base común para esas aplicaciones (Harsu, 2002). Esta puede ser aplicada tanto para sistemas nuevos como para los existentes.

3.1. DOMINIO

Un dominio o un dominio de aplicación puede definirse como un área de conocimiento o familia de sistemas en el cual se observan los conceptos relacionados, se determinan los aspectos comunes y las variaciones con el fin de resolver problemas mediante la realización de artefactos de *software* (Hamar, 2004).

Licencia Creative Commons:

Un dominio puede ser caracterizado por un lenguaje común para describir los conceptos y problemas en el mismo.

El término dominio puede ser usado en varias asociaciones (Schmid,2000):

- Área de negocio
- Colección de problemas (dominio del problema)
- Colección de aplicaciones (dominio de la solución)
- Área de conocimiento con terminología común.

Los dominios pueden ser considerados desde dos puntos de vistas: como un contexto orientado a objetos y como un conjunto de sistemas (Simos, 97). Primero, son considerados como **un contexto orientado a objetos**, es decir, un dominio puede ser visto como un mundo real. Este enfoque se concentra en el fenómeno y sus procesos. Los requerimientos de diferentes aplicaciones no son considerados. Este tipo de Ingeniería de Dominio es algo similar al modelado conceptual, y cubre a las áreas del negocio y el dominio del problema. El modelado de dominio, derivado de esta alternativa, es altamente reusable, porque los conceptos de un área de problema son algo estable. Ellos no necesariamente varían, a pesar de que los requerimientos causen cambios en los sistemas.

La otra forma de considerar un dominio es cuando es visto como **un conjunto de sistemas**. Este punto de vista se concentra en las familias de aplicaciones. El dominio es abordado de acuerdo a las similitudes entre las aplicaciones. Por tanto, este enfoque cubre los dominios solución. Esta última alternativa es más cerrada para productos de ingeniería en línea, mientras que el anterior también satisface la ingeniería de sistemas simple.

Los dominios pueden ser divididos en verticales y horizontales (Czarnecki, 1997). Los dominios verticales son sistemas de *software* clasificados de acuerdo a las áreas de negocio, como por ejemplo, sistemas de reservaciones aéreas, sistemas de registro medico, sistemas de gestión de clientes, sistemas de procesamiento de órdenes y sistemas de gestión de inventario.

Licencia Creative Commons:

Los dominios horizontales son partes de sistemas de *software* clasificados de acuerdo a su funcionalidad, como por ejemplo, sistemas de base de datos, librerías contenedoras, sistemas *workflow*, Interfaces Gráficas de Usuario (*GUI*) y librerías de código numérico. Cuando se aplica Ingeniería de Dominio a un dominio vertical, el resultado podría ser *software* reutilizable que puede ser presentado en forma de marco de trabajo reutilizable y sus componentes. En el caso de los dominios horizontales, la Ingeniería de Dominio podría producir componentes reutilizables.

3.2. FASES DE LA INGENIERÍA DE DOMINIO

La Ingeniería de Dominio es un proceso que, a través de una serie de fases definidas y los productos de las mismas, permiten que el conocimiento de un dominio sea capturado y vaya evolucionando (Hamar, 2004). Este proceso está conformado por tres fases: análisis del dominio, diseño del dominio e implementación del dominio (Czarnecki, 1997).

3.2.1. Análisis del Dominio

En la fase de análisis del dominio se estudia el dominio del problema de una familia de aplicaciones. El análisis del dominio está asociado con Reutilización, su propósito es capturar información involucrada con el dominio a ser reusado en el desarrollo de otras aplicaciones del mismo dominio. El resultado de esta fase es el modelo del dominio, el cual está conformado por los siguientes elementos:

- **Alcance del dominio (definición del dominio, análisis del contexto):** Esta actividad proporciona ejemplos de aplicaciones que pertenecen al dominio, así como ejemplos de aplicaciones que están fuera del dominio. Además, da las reglas de inclusión y exclusión.
- **Análisis de concordancias:** Considera las concordancias y variaciones de las aplicaciones en el dominio. El propósito de esta actividad consiste en estudiar los requerimientos y propiedades de las

aplicaciones y los conceptos en el dominio. El análisis de concordancias produce como resultado un documento de concordancias.

- **Diccionario del dominio (vocabulario del dominio):** Proporciona y define los términos concernientes al dominio. El propósito de esta actividad es hacer la comunicación entre el grupo de desarrollo y los expertos de la organización (*stakeholders*) más fácil y precisa.
- **Notaciones (modelado del concepto, representación del concepto);** Proporciona una manera uniforme de representar los conceptos usados en el modelado de dominio. Ejemplo de tales notaciones son: diagramas de objetos, diagramas de estado-transición, diagramas entidad-relación y diagramas de flujo de datos.
- **Ingeniería de requerimientos (modelado de características):** Comprende reuniones, definición, documentación, verificación y gestión de los requerimientos que especifican las aplicaciones en el dominio. En el contexto línea de producto, el propósito es reusar y configurar los requerimientos entre aplicaciones individuales. Estos requerimientos pueden ser llamados características, la cual es definida como un atributo del sistema que es visible al usuario final, o puede comprender también atributos que son relevantes para algún experto de la organización (*stakeholder*) (no necesariamente visible).

3.2.2. Diseño del Dominio

En esta fase, se diseña la arquitectura base para una familia de aplicaciones. Esto comprende la selección del estilo arquitectural y su representación utilizando diferentes vistas. La arquitectura base podría proporcionar variaciones entre aplicaciones. Es en este momento, cuando se decide cómo permitir esta variabilidad o flexibilidad de configuración. De acuerdo a las características del modelo y al documento de concordancias, esto podría ser seleccionado con componentes o

ítems (tales como requerimientos) los cuales son provistos en la arquitectura base y los cuales son implementados como variaciones en aplicaciones individuales.

El diseño del dominio produce un plan de producción que indica como la aplicación concerniente puede ser derivada de la arquitectura base y de los componentes reutilizables. El plan de producción comprende la descripción de los sistemas con sus interfaces para cada cliente, directivas para el proceso de ensamblaje de los componentes, y directivas para gestionar los cambios requeridos por los clientes. El proceso de ensamblaje puede ser manual, semi-automático o automático.

3.3.3. Implementación del Dominio

Esta fase cubre la implementación de la arquitectura, componentes, y herramientas diseñadas en la fase previas. Esta comprende la escritura de la documentación y la implementación de lenguajes de dominio específico y generadores.

El propósito de la Ingeniería de Dominio es producir activos reutilizables que son implementados en esta fase. Así, el resultado del ciclo completo de la Ingeniería de Dominio comprende componentes, modelos de características, modelos de análisis y diseño, arquitecturas, patrones, marcos de trabajo, lenguajes de dominio específico, planes de producción y generadores.

3.3. MÉTODOS DE LA INGENIERÍA DE DOMINIO

A continuación se presentan los métodos más conocidos de la Ingeniería de Dominio.

3.3.1. FODA (*Feature-Oriented Domain Analysis*) (Kang Cohen y otros, 1990): Considera las características de las aplicaciones similares en un dominio. Las características son capacidades de las aplicaciones consideradas desde el punto de

vista del usuario final. Las características cubren tanto los aspectos comunes como los variables entre los sistemas relacionados. *FODA* incluye diferentes análisis tales como el análisis de requerimientos y el análisis de características. Este produce un modelo de dominio que abarca las diferencias entre las aplicaciones relacionadas. *FODA* es extendido a *FORM* (*Feature-Oriented Reuse Method*) (Kang, Kim y otros, 1999) para incluir las fases de diseño e implementación de *software*. *FORM* cubre tanto el análisis de las características del dominio como el uso de esas características en el desarrollo de activos del dominio reutilizables.

3.3.2. *ODM (Organization Domain Modeling)* (Simos, Creps y otros, 1996):

Se concentra, principalmente, en la Ingeniería de Dominio de los sistemas legados. Sin embargo, puede ser aplicada para los requerimientos de un nuevo sistema. *ODM* combina diferentes activos tales como requerimientos, diseño, código y procesos de varios sistemas legados con elementos comunes reutilizables. *ODM* es soportado por *DAGAR* (*Domain Architecture-based Generation for Ada Reuse*) (Klingler y Solderitsch, 1996). El proceso *DAGAR* incluye actividades tanto para la Ingeniería de Dominio como para la ingeniería de aplicación.

3.3.3. *RSEB (Reuse-Driven Software Engineering Business)* (Jacobson,

Griss y Jonson, 1997): Es un método sistemático de Reutilización orientado a modelos. Está compuesto de un conjunto de aplicaciones relacionadas de un conjunto de componentes reutilizables. *RSEB* usa *UML* para especificar sistemas de aplicaciones, sistemas de componentes reutilizables y arquitecturas en capas. Las variabilidades entre sistemas son expresadas con puntos de variación y variaciones unidas. *FeatuRSEB* (*Features RSEB*) (Griss, Favaro y d'Alessandro, 1998) conecta características (de *FODA*) con *RSEB*. Actualmente, *FODA* y *RSEB* tienen mucho en común: ambos son métodos orientados al modelo, proporcionando así, varios modelos correspondientes a diferentes puntos de vista del dominio.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

3.3.4. PuLSE (Product Line Software Engineering) (Bayer, DeBaud y otros, 1999): divide el ciclo de vida de la línea de productos en tres partes. En la fase inicio, *PuLSE* es personalizado para ajustar aplicaciones particulares. La adaptación es afectada por la naturaleza del dominio, estructura del proyecto, contexto organizacional y la Reutilización. En la segunda fase, es construida la infraestructura de la línea de productos. El próximo paso incluye alcance, modelado y arquitectura de la línea de productos. En la tercera fase, La infraestructura de la línea de productos es utilizada par crear productos individuales.

PuLSE tiene varios componentes: *PuLSE-DSSA (PuLSE – Domain-Specific Software Architecture)* (Anastasopoulos, Bayer y otros, 2000), el cual permite desarrollar una arquitectura específica de dominio basada en el modelo de la línea de productos, *PuLSE-Eco* se concentra en el alcance económico y *PuLSE-EM* en la gerencia y evolución.

WWW.BDIGITAL.ULA.VE

3.3.5. FAST (Family-Oriented Abstraction, Specification, and Translation) (Weiss y Robert, 1999): Cubre todo el proceso de la ingeniería de productos en línea. Sin embargo, divide a la Ingeniería de Dominio en dos partes: Análisis del dominio e implementación del dominio. De esta manera, las actividades que involucran diseño del dominio son consideradas en la fase de análisis del dominio. *FAST* proporciona directivas sistemáticas para cada uno de los pasos durante la ingeniería de línea de productos. Estos pasos pueden ser representados como transiciones entre estados. *FAST* también describe los activos que son producidos en cada fase.

En esta investigación, la Ingeniería de Dominio facilitó el proceso de análisis de los dominios de aplicación de las siguientes aplicaciones *web*: portales, *e-commerce*, *e-business*, sistemas de información *web* y *e-government*. El método utilizado en esta etapa fue el FODA, ya que permite agrupar las características comunes de las aplicaciones en un dominio.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

4. REUTILIZACION DE SOFTWARE

Reutilización de *software* es el proceso de crear sistemas de *software* a partir de *software* existente, en lugar de desarrollarlo desde el comienzo (Sametinger, 1997). La Reutilización de *software* es un proceso de ingeniería de *software* que conlleva el Reutilización de activos de *software* en: la especificación, análisis, diseño, implementación y pruebas de una aplicación o sistema de *software* (Montilva, 2002).

4.1. PRODUCTOS REUTILIZABLES

Un activo de *software* reutilizable es un producto de *software* diseñado expresamente para ser reutilizado en el desarrollo de muchas aplicaciones (Montilva, 2002). Un activo puede ser: un componente de *software* (ejemplo: un módulo, una clase, un procedimiento, una función, etc.), una especificación de requerimientos, un modelo o especificación de diseño, un algoritmo, un patrón de diseño, una arquitectura de dominio, un esquema de base de datos, una especificación de prueba, la documentación de un sistema o un plan.

Un componente de *software* reutilizable es un artefacto auto-contenido y claramente identificable que describe y/o ejecuta funciones específicas, tiene interfaces claras, una documentación apropiada y un *status* de Reutilización definido (Sametinger, 1997).

Los productos que pueden ser reutilizados son:

1. **Algoritmos:** elaborados, generalmente, en un pseudocódigo que debe ser luego convertido a un programa en un lenguaje de programación. Por lo general, requieren de modificaciones para su Reutilización en otra aplicación o dominio.
2. **Librerías de definiciones:** colección de subrutinas, funciones o procedimientos escritos en un lenguaje de programación, Ejemplo: librerías estadísticas, librerías matemáticas, librerías de un lenguaje.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

3. **Librerías de clases:** colección de clases escritas en un lenguaje de programación orientado a objetos (*Java*, *C++*, *Smalltalk*, etc.).
4. **Arquitecturas y diseños de *software*:** establece la manera en que los componentes del sistema están interconectados.
5. **Marcos (*frameworks*):** es una colección flexible de clases semielaboradas escritas en un lenguaje de programación y elaboradas con el propósito de ser extendidas o refinadas a través de su Reutilización.
6. **Patrones de diseño:** describe como resolver ciertos problemas recurrentes de diseño observando los elementos comunes a una colección de aplicaciones en un dominio dado.

4.2. PROCESO DE DESARROLLO DE SOFTWARE BASADO EN LA REUTILIZACION DE COMPONENTES

El proceso de desarrollo de *software* basado en la Reutilización de componentes se clasifica en: desarrollo de *software* con Reutilización de componentes y desarrollo de componentes de *software* reutilizables (Sametinger, 1997).

El desarrollo de *software* con Reutilización de componentes consiste en el desarrollo de una nueva aplicación involucrando el Reutilización de un conjunto de componentes existentes. Este enfoque maximiza la Reutilización de componentes de *software* existentes y/o reduce el número de componentes que requieren ser desarrollados desde el comienzo.

El desarrollo de componentes de *software* reutilizable consiste en la adaptación o desarrollo de componentes con el propósito expreso de ser reutilizados en futuras aplicaciones. Su objetivo es producir repositorios de activos o componentes de *software* reutilizables que puedan ser reutilizados en el desarrollo de *software*.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

4.3. TECNOLOGIAS DE COMPONENTES

Existen varias tecnologías de componentes en el mercado. El propósito de las mismas es el de proporcionar a los desarrolladores una arquitectura de componentes distribuidos que permita la comunicación entre componentes de diferentes vendedores en diferentes plataformas. A continuación se explican brevemente cada una de estas tecnologías.

4.3.1. *Common Object Request Broker Architecture (CORBA)*

Es una arquitectura y especificación para crear, distribuir y manejar objetos programados distribuidos dentro de una red. *CORBA* fue creado por un consorcio de vendedores a través de *Object Management Group (OMG)*, el cual actualmente incluye más de 800 compañías (CORBA, 2002).

Las aplicaciones *CORBA* están compuestas por objetos, unidades individuales de *software* corriendo que combinan funcionalidad y datos, y que representan algo en el mundo real. Típicamente son instancias de un único objeto de un único tipo. Por ejemplo, una aplicación de comercio electrónico puede tener muchas instancias de carritos de compra, todos idénticos en funcionalidad pero asignados a diferentes compradores, y contienen datos que representan la mercancía del comprador seleccionado.

Los objetos son conectados a través de interfaces *IDL* usando un *ORB*. El Lenguaje de Definición de Interfaces (*IDL*) se utiliza para definir un objeto *CORBA*, el mismo es utilizado para especificar las interfaces entre los objetos *CORBA*. El *Object Request Broker (ORB)* es la infraestructura de comunicación a través de la cual el cliente accede a los servicios del objeto distribuido.

En la arquitectura *CORBA*, cualquier cliente *CORBA* sobre cualquier red puede acceder a cualquier objeto *CORBA* sin importar su ubicación, plataforma de ejecución y protocolo de red.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

4.3.2. Java 2 Platform Enterprise Edition (J2EE)

Es una plataforma *Java* diseñada para grandes empresas, creada por *Sun Microsystems* para crear aplicaciones complejas que pudieran atender a miles de usuarios en forma simultánea, 24 horas al día, siete días a la semana, sin ningún tiempo de inactividad. *J2EE* simplifica la creación de las aplicaciones empresariales, ya que su funcionalidad se encapsula en los componentes de *J2EE*.

J2EE es una tecnología versátil ya que los componentes de las aplicaciones que se construyen con *J2EE* se comunican entre sí a través de métodos estándares de comunicación como son: *HTTP*, *SSL*, *XML*, *RMI* e *IIOP*.

Los principales componentes de *J2EE* son: los *Enterprise Java Beans (EJB)*, los *servlets* y los *JavaServer Pages (JSP)*. Además, *J2EE* incluye siete (7) servicios más, los cuales son:

- a) **Compatibilidad con CORBA:** *Java IDL* y *RMI-IIOP* permiten comunicarse con cualquier sistema empresarial compatible con tecnología *CORBA* e interactuar con sistemas heredados. *JavaIDL* se utiliza para conectar programas *Java* con sistemas *CORBA*. *RMI-IIOP* es la unión de la *API* de invocación remota de métodos (*RMI*) de *Java* con el *Internet Inter-ORB Protocol (IIOP)* que utiliza *CORBA*, la cual permite conectar programas *Java* con sistemas heredados.
- b) **JavaMail:** es la *API* que permite comunicarse mediante el envío y recepción de mensajes de correo electrónico para que los clientes y los sitios de comercio electrónico puedan intercambiar información.
- c) **Servicio de Mensajes de Java (JMS):** se utiliza para incluir en los programas *Java* un enlace de transmisión y recepción en forma asíncrona de mensajes tolerantes a fallos entre los componentes.
- d) **Interfaz de Nombres y Directorios para Java (JNDI):** Permite a los programas *Java* localizar objetos en forma sencilla.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

- e) **API Java para Transacciones (JTA):** permite incluir código para gestionar las transacciones dentro de los componentes.
- f) **Java DataBase Connectivity (JDBC):** permite a un programa conectarse e interactuar con cualquier sistema manejador de bases de datos comercial.
- g) **Descriptores de despliegue XML:** permite crear herramientas y componentes que interactúan con documentos XML.

4.3.3. .NET

Es un marco de modelo de programación creado por *Microsoft*, que permite a los desarrolladores construir aplicaciones basadas en *web*, aplicaciones inteligentes, aplicaciones servicios *web XML* utilizando protocolos estándares como *SOAP* y *HTTP*.

El lenguaje común del tiempo de corrida (C#) es responsable por los servicios en el tiempo de ejecución, tales como: el lenguaje de integración de seguridad, memoria procesos y manejo de hilos. Además, en el tiempo de desarrollo maneja el ciclo de vida y la construcción dinámica, lo que permite reducir el tiempo de codificación para introducir la lógica del negocio dentro de un componente reutilizable.

Las librerías de clases proporcionan una funcionalidad estándar tal como entradas/salidas, manejo de cadenas, manejo de seguridad, comunicación en red y otras funciones. Las clases de datos de *Microsoft ADO.NET* soportan el manejo de la persistencia de datos e incluye las clases *SQL* para manipular datos persistentes y búsquedas *XML*. Las clases de *Microsoft ASP.NET* soportan las aplicaciones *web* y los servicios *web XML*.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

5. MODELO DE MADUREZ DE CAPACIDAD INTEGRADO (CMMI)

El proyecto *CMMI* fue formado para solucionar el problema de utilizar múltiples *CMMs*. El producto de la misión del equipo de *CMMI* fue combinar tres modelos: 1) Modelo de Madurez de Capacidad para Software (*SW-CMM*), 2) Alianza de Industrias Electrónica/Estándar Intermedio (*EIA/IS*), y 3) Desarrollo de Producto Integrado - Modelo de Madurez de Capacidad (*IPD-CMM*) en un simple marco de trabajo para ser usado por la organización para el mejoramiento de procesos de la misma.

5.1. NIVELES DE MADUREZ

El nivel de madurez de una organización proporciona una manera de predecir el desempeño futuro de la misma en una disciplina dada o en conjunto de disciplinas. Un nivel de madurez es definido como el mejoramiento evolutivo de procesos. Cada nivel de madurez establece una parte importante de los procesos de la organización (CMU/SEI,2002).

El nivel de madurez consiste en un conjunto de áreas de proceso predefinidas. El nivel de madurez es medido por el logro de las metas específicas y genéricas que aplican a cada área de proceso predefinida. En el modelo *CMMI*, existen cinco (5) niveles de madurez, cada una de las capas es el inicio para ir a un nuevo proceso de mejora. Las mismas están designadas con los números desde el 1 hasta el 5: 1)Inicial, 2)Gestionado, 3)Definido, 4) Gestionado Cuantitativamente y 5)Optimizado.

1) Nivel de Madurez 1: Inicial

Los procesos son usualmente *ad hoc* y caóticos. La organización no proporciona un ambiente estable. Los logros de la organización dependen de las competencias heroicas de su personal y no de los procesos probados. En este nivel, generalmente las organizaciones producen productos y servicios para trabajar, sin embargo frecuentemente exceden el tiempo y recursos de sus proyectos.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

2) Nivel de Madurez 2: Gestionado

A este nivel una organización ha logrado todos sus objetivos genéricos y específicos de las áreas de procesos del nivel de madurez 2. En otras palabras, los proyectos de la organización tienen asegurado que sus requerimientos son gestionados, y sus procesos son planificados, ejecutados, medidos y controlados.

Se establece comunicación con los expertos de la organización más importantes (*stakeholders*) para revisar sus necesidades, controlar los productos de trabajo y servicios y para verificar si satisfacen sus requerimientos, estándares y objetivos de la organización.

3) Nivel de Madurez 3: Definido

A este nivel la organización ha logrado todos los objetivos genéricos y específicos de las áreas de procesos asignadas a los niveles de madurez de 2 y 3. En el nivel de madurez 3 los procesos están bien caracterizados y entendidos, y son descritos en estándares, procedimientos, herramientas y métodos.

4) Nivel de Madurez 4: Gestionado Cuantitativamente

A este nivel, una organización ha logrado todos los objetivos específicos de las áreas de procesos asignadas a los niveles de madurez 2, 3 y 4 y los objetivos genéricos asignados a los niveles de madurez 2 y 3. Los subprocesos que contribuyen significativamente en la ejecución de todos los procesos son seleccionados y controlados usando estadísticas y otras técnicas cuantitativas.

Los objetivos cuantitativos de calidad y ejecución de procesos son establecidos y usados como criterio en la gestión de procesos. Los objetivos cuantitativos están basados en las necesidades del cliente, usuarios finales, organización e implementadores de procesos. La calidad y la ejecución de procesos son entendidos en términos estadísticos y son gestionados a través de la vida de un proceso.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

5) Nivel de Madurez 5: Optimizado

A este nivel una organización ha logrado todos los objetivos específicos de las áreas de procesos asignadas a los niveles de madurez 2, 3, 4 y 5 y los objetivos genéricos asignados a los niveles de madurez 2 y 3. Los procesos son continuamente mejorados basados en el entendimiento cuantitativo de las causas comunes de variación inherentes a los procesos.

El nivel de madurez 5 se enfoca en el mejoramiento continuo de la ejecución de procesos a través de las mejoras tecnológicas incrementales e innovadoras. Los objetivos cuantitativos del mejoramiento de procesos para la organización son establecidos, continuamente revisados para reflejar cambios en los objetivos del negocio y usados como criterio en la gestión del mejoramiento de procesos.

Para esta investigación, el nivel de madurez de una organización nos proporciona una manera de predecir el desempeño de la misma en el proceso de desarrollo de una aplicación *web*. Por tal razón, nos ayudó a considerar estrategias de acuerdo al nivel de madurez donde estuviese ubicada una organización.

6. CONCLUSION

La revisión documental y los aportes realizados en este capítulo permiten asegurar que existe una relación muy estrecha entre aplicaciones *web*, Ingeniería de Métodos, Ingeniería de Dominio y Reutilización de software. Asimismo, el proceso desarrollo de aplicaciones *web* esta directamente relacionado con el nivel de madurez de una organización de acuerdo al *CMMI*.

La Ingeniería de Dominio permite capturar información de las aplicaciones para ser reutilizada en otras que pertenezcan al mismo dominio. La Ingeniería de Métodos tiene como objetivo fundamental la construcción de nuevos métodos a través del Reutilización de fragmentos de métodos existentes.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Dado que el objetivo de esta investigación es diseñar un método situacional para el desarrollo de aplicaciones *web*, es decir, que se pueda adaptar fácilmente a varios proyectos que pertenecen a un mismo dominio o a dominios similares, se requiere del conocimiento de las aplicaciones *web* para poderlas describir, de la Ingeniería de Métodos como guía fundamental para construir el método situacional y de la Ingeniería de Dominio como apoyo fundamental para capturar, modelar y reutilizar todos los aspectos comunes de cada una de las aplicaciones *web* consideradas.

Asimismo, se requiere de la Reutilización de *software* para implementar mecanismos de Reutilización de componentes (patrones de diseño, componentes gráficos, etc.) y la incorporación de las tecnologías de componentes vigentes en el mercado a nuestro método situacional. Además, se requiere del *CMMI* para gestionar el proyecto de desarrollo de la aplicación *web* tomando como base el nivel de madurez de la organización.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

CAPITULO 3

MODELO DE PRODUCTO DE UNA APLICACIÓN WEB

Tal como se explicó en el capítulo anterior, el modelo de producto es una abstracción de las propiedades comunes que se encuentran en cualquier producto de desarrollo. Este representa, a través de notación formal o gráfica, los conceptos generales que son necesarios para construir un producto. En este capítulo se presenta un modelo de producto para una aplicación *web*, un modelo de producto de un sistema de negocios, un modelo de producto asociado al ciclo de vida de una aplicación *web*, los mismos se representan a través de la notación proporcionada por el Lenguaje de Modelado Unificado (UML) en su versión 1.5.

El modelo de producto para una aplicación *web* describe los conceptos necesarios que deben estar presentes en una aplicación *web*. El modelo de producto de un sistema de negocios describe el conjunto de conceptos genéricos que deben ser encontrados en cualquier sistema de negocios.

El modelo de producto asociado al ciclo de vida de una aplicación *web* está conformado por tres (3) sub-modelos: el sub-modelo de gestión de proyectos, el sub-modelo de la aplicación *web* y el sub-modelo de implantación y mantenimiento de la aplicación *web*.

Dado que el objetivo del capítulo es presentar el modelo de producto de una aplicación *web*, el resto de los modelos y sub-modelos sólo se presentan de manera muy general.

1. CARACTERÍSTICAS DE LAS APLICACIONES WEB SELECCIONADAS.

De las aplicaciones *web* estudiadas en el capítulo anterior, se han seleccionado cinco (5), las cuales servirán de base para el diseño del método situacional para el desarrollo de aplicaciones *web*. Este método garantizará una guía completa al equipo de desarrollo, tomando en cuenta las necesidades particulares del dominio de aplicación seleccionado.

En este caso particular, se han seleccionado los métodos que tienen relación con el dominio de aplicación para el cual son desarrolladas las aplicaciones *web*. El

hecho de que cada aplicación tenga un propósito definido permite diferenciarlas entre sí, es decir, cada una de ellas ofrece distintos servicios según las necesidades de los usuarios propios de cada dominio sin tomar en cuenta el grado de complejidad de dicho servicio.

Los métodos seleccionados permiten desarrollar: portales, comercio electrónico, negocios inteligentes, sistemas de información *web* y gobiernos electrónicos.

Es importante aclarar, que cada una de estas aplicaciones *web* presenta una arquitectura similar, la cual se divide en 3 capas: Presentación, lógica del negocio y datos (Montilva y Barrios, 2002). La capa de presentación contiene los elementos necesarios para generar la Interfaz Gráfica de Usuario (GUI) adecuada para mostrar las páginas *web* que han sido solicitadas por un programa cliente. Estas páginas se encuentran almacenadas en un servidor *web*, el cual a su vez contiene páginas dinámicas que se presentarán de acuerdo a las solicitudes del usuario.

La capa de lógica del negocio, contiene los objetos del negocio que son comunes en una aplicación *web*, y los objetos *web*. Los objetos del negocio representan el mundo real, es decir, las personas, lugares, cosas y eventos que pueden ser encontrados en el dominio de aplicación. Estos son ensamblados para garantizar la funcionalidad de la aplicación *web*.

La capa de datos está conformada por el repositorio de componentes y el repositorio de datos. El repositorio de componentes contiene almacenados componentes, los cuales son paquetes auto-contenidos, que ofrecen una funcionalidad claramente definida, con interfaces abiertas para ofrecer la posibilidad de “*plug-and-play*” en la capa de lógica del negocio. El repositorio de datos contiene las bases de datos, las cuales pueden ser de relacionales, orientadas a objetos y/o mixtas o datos en *XML*.

A continuación se caracterizarán cada una de las aplicaciones *web* seleccionadas:

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

1.1. PORTALES

El concepto “portal” se comenzó a utilizar para referirse a lugares donde era posible encontrar una oferta de servicios de valor añadido, tales como correo electrónico, noticias, información, etc (Colado, 2003). Un portal provee un punto único y seguro de interacción con información diversa, procesos de negocios y personas, personalizando las necesidades de sus usuarios y atributos. Entre las principales características de un portal se mencionan las siguientes:

- Seguridad a nivel de base: *Login* único, autorización, *Public Key Infrastructure (PKI)*.
- Servicios de directorio para administrar usuarios: escalables y administrables.
- Ambiente de desarrollo “sin desarrollo”: Agregar información y servicios es simple y requiere poco código.
- Arquitectura uniforme.
- Motor de perfilación (roles) y personalización (preferencias).
- Administración, métrica, auditoría.
- Administración de contenido: flujo de publicación, agregación, búsquedas, categorización.

La importancia de un portal radica en los valores esperados de los tres actores que intervienen en el mismo:

- **El cliente:** Persona, organización o parte de ella que promueve, patrocina o requiere el portal. Los valores que espera obtener son: acortar los ciclos de entrega de información, facilitar el acceso a información agregada, aumentar la satisfacción de los usuarios, permitir integrar diferentes organizaciones proveedoras, contar con una infraestructura básica de manejo de información segura y funcional e integrar la cadena de valor entre proveedores, personal interno, administrativo y clientes en un único medio.
- **El Usuario:** Visitante o navegante que interactúa con el portal. El espera contar con un único punto de acceso a información y servicios, encontrar un sitio proactivo (alarmas, semáforos), información de

interés específico, visitar un sitio personalizado, con una Interfaz Gráfica de Usuario (GUI) estandarizada y no necesitar de múltiples claves.

- **Los desarrolladores:** Personas dentro de la misma organización o externa a ella, que tienen a cargo la ejecución del proyecto del portal. Esperan lograr escalabilidad, flexibilidad, administración, seguridad, apoyo en estándares, desacoplo de sistemas, a través de la separación de presentación, servicios de infraestructura, *porlets* y *webservices*, soporte multidispositivo, maximización de la Reutilización.

Lo dicho anteriormente, se resume en que cada usuario busca una funcionalidad diferente del portal de acuerdo a su necesidad particular, tal como se puede observar en la figura 10.

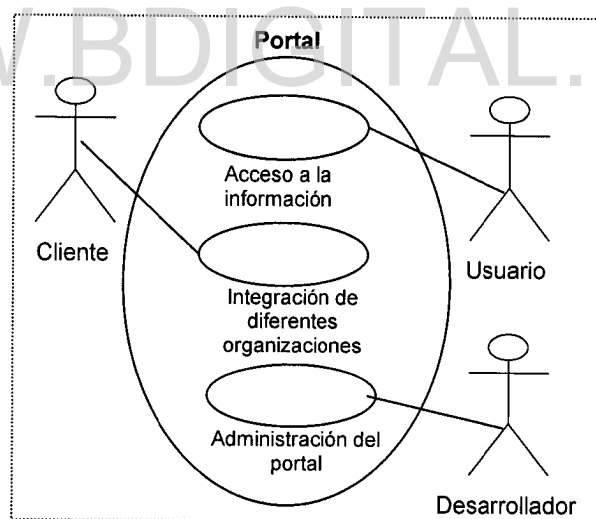


Figura 10 Diagrama de caso de uso de un portal

1.1.1. Infraestructura de componentes de un portal, está basada en los siguientes componentes:

- **Capa de Presentación:** Interfaz para el usuario *web* con soporte de dispositivos ampliados.
- **Personalización:** Capacidad de atención dinámica al usuario basado en sus perfiles personales.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

- **Colaboración:** Herramientas que permiten intercambios entre usuarios.
- **Portlets:** Marco de trabajo para unir fácilmente módulos y servicios.
- **Aplicaciones y workflow:** Integración entre el sistema legado y nuevas aplicaciones.
- **Búsqueda y navegación:** Clasificación de los repositorios de contenido y búsqueda de la información relevante.
- **Publicación y suscripción:** Capacidad de autoría y publicación de nuevo contenido a los suscriptores.
- **Administración y seguridad:** Servicios *web* básicos tales como diseñadores *web*, monitores de desempeño, servicios de *clusters*, y administración de metadatos.
- **Integración:** Metadatos, *XML*, conectores estándares, *EAI* (*Enterprise Application Integration*).

Los portales pueden ser horizontales o verticales (Colado, 2003). El horizontal está constituido por la infraestructura base sobre la cual se construye un portal; y el vertical se construye sobre una capa horizontal y representa un caso de portal específico, definido generalmente por un tópico principal o un dominio. Los más comunes son: motores de búsquedas, Sistemas de Información Geográfica (GIS), gestión de contenidos, administración y sistemas de alertas.

1.1.2. Arquitectura de un Portal. Como se dijo anteriormente, el desarrollo de este tipo de aplicaciones *web* es expresado en términos de las capas de presentación, de lógica de aplicación y de datos. Sin embargo, muchos de los contenidos y servicios que se presentan en un portal se desarrollan como transversales sobre estas capas. Tal es el caso de los Sistemas de Información Geográfica (GIS), los motores de búsqueda, etc. Este tipo de portales requieren de lógica y datos propios, y se integran en el portal mediante la capa de lógica de aplicación.

En la figura 11, se muestra el modelo de producto de un portal. En este se observa que en la capa de lógica del negocio se combina la información y se adapta a las necesidades de cada usuario. Asimismo, en la capa de datos es donde se

almacena de forma persistente toda la información necesaria para facilitar los servicios ofrecidos por el portal: el perfil de los usuarios del portal, índices de motores de búsqueda y contenidos de agregación e información sobre la publicación.

El estilo arquitectónico de un portal establece la forma en la que se ensamblan los componentes del portal. De manera general, se pueden diferenciar los siguientes:

a) **Arquitectura basada en los datos.** Se emplea en la mayor parte de portales con tecnología “*Java Server Pages*” (JSP) o “*Active Server Pages*” (ASP) para poder ofrecer contenidos dinámicos. En esta arquitectura se invoca un componente capaz de acceder a la información almacenada en el portal y presentarla directamente. Tanto la lógica de acceso a los recursos como la lógica de aplicación se encuentran en este componente. La ventaja de este tipo de arquitectura es su simplicidad, aunque se hace difícil la escalabilidad y la Reutilización de componentes (Colado, 2003).

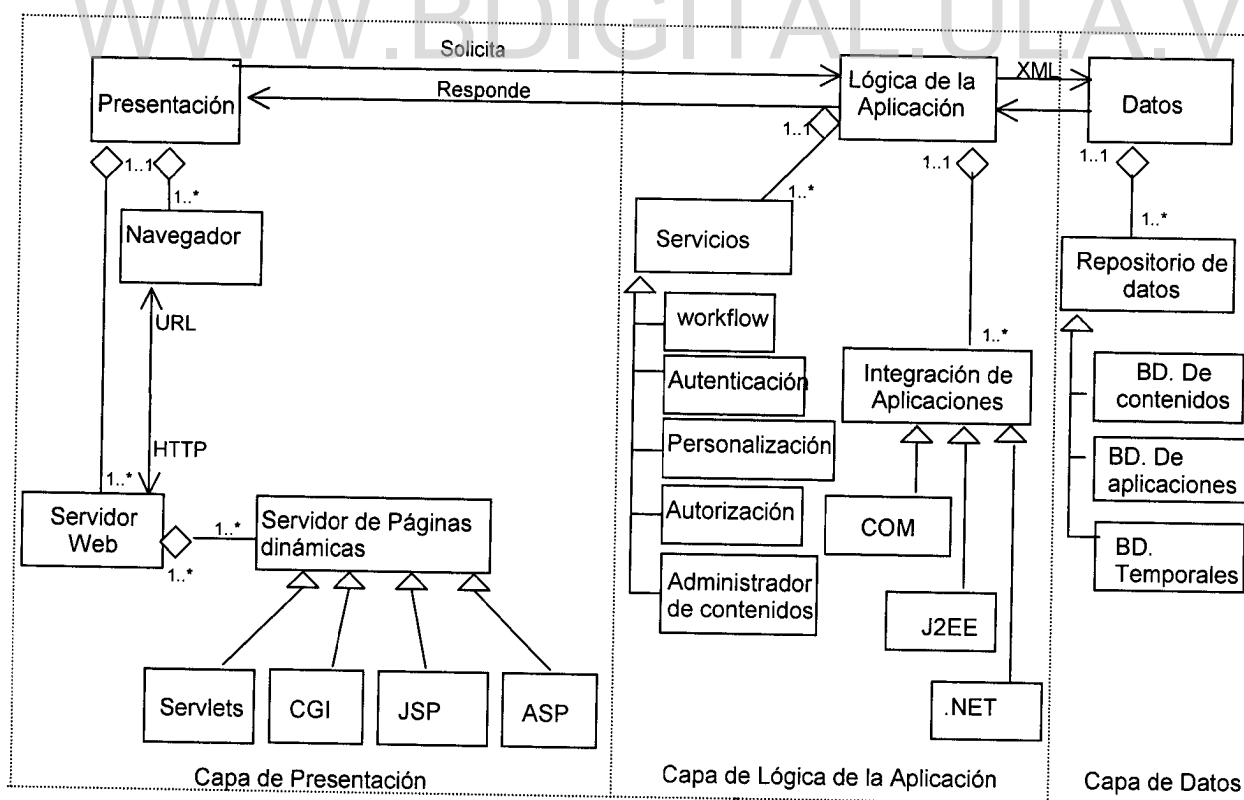


Figura .11. Modelo de Producto de un Portal

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

b) Arquitectura centrada en el gestor de contenidos. Es el caso de portales que se adaptan a la funcionalidad proporcionada por un gestor de contenidos. La ventaja de este tipo de arquitecturas es la reducción en el desarrollo, ya que, prácticamente es una solución cerrada. Sin embargo, el crecimiento de este tipo de soluciones no siempre es posible, a pesar de ser modulares. Es muy costoso integrar elementos nuevos en el portal (Oracle 2003).

c) Arquitectura Modelo-Vista-Controlador (MVC). Este tipo de arquitectura proporciona una separación entre lo que es la lógica de negocio y la vista (presentación). El controlador funciona redirigiendo la petición, por lo que es fácil cambiar la presentación para una determinada petición, así como tener una aplicación con diferentes vistas para distintos mecanismos de acceso que comparten la lógica de la aplicación (Colado, 2003).

d) Arquitectura de *Portlets*. Un *portlet* es un componente *web* capaz de generar contenido dinámico e interactuar con los usuarios a través del portal. La ventaja que plantean estos elementos es su versatilidad a la hora de ser desarrollados y desplegados. Los *portlets* pueden generar distintos tipos de contenidos en función del mecanismo de acceso, información a la que tienen acceso. Un *portlet* debe ser visto como la encapsulación de una aplicación *web* para poder ser utilizada como un componente dentro de un portal (Jakarta, 2003).

1.2. APLICACIONES DE COMERCIO ELECTRÓNICO (*e-commerce*).

Generalmente, las aplicaciones de comercio electrónico están asociadas con la compra y venta de información, productos y servicios vía *Internet*. Sin embargo, el *e-Commerce*, también es utilizado para transferir y compartir información en organizaciones, a través de *Intranets*, para mejorar la toma de decisiones y eliminar la duplicación de esfuerzos.

El nuevo paradigma de *e-Commerce* es construir aplicaciones basadas en las transacciones, pero construyendo, sosteniendo y mejorando las relaciones, tanto las existentes como las potenciales. *E-commerce* es una infraestructura para extender una compañía enfocada a su interior con procesos de negocio únicos a clientes,

Licencia Creative Commons.

socios de negocio, proveedores y distribuidores a nuevas aplicaciones enfocadas al exterior.

Las aplicaciones de comercio electrónico pueden ser categorizadas en cuatro grupos principales: Gestor de Ventas (*Vendor Management*), Mercados Electrónicos (*I-Market*), Valor Extendido/Cadena de Suministro (*Extended Value/Supply Chain*) y Atención al Cliente (*Customer Care*). Por el lado de las ventas, las aplicaciones de *I-Market* incluyen gestión de catálogos en línea, gestión de órdenes, comunidades de comercio, publicidad y mercadeo; mientras que las aplicaciones *customer care* involucran auto-servicio, gestión de relaciones con el cliente y soporte a negocios inteligentes.

En el lado de las compras, el sistema *Vendor Management* automatiza la procura de recursos operativos indirectos, incluyendo información, oferta/demanda de compras y catálogos de los proveedores; mientras que las aplicaciones *Extended Value/Supply Chain* incluyen pronóstico y planificación colaborativa, agenda y logística.

En tal sentido, en una aplicación de comercio electrónico, intervienen los siguientes actores: clientes, socios de negocio, proveedores y distribuidores; teniendo cada uno un conjunto de funcionalidades asociadas a cada una de las categorías mencionadas anteriormente (ver figura 12).

1.2.1. Etapas de las Aplicaciones de Comercio Electrónico. Las aplicaciones de comercio electrónico han pasado por tres etapas, tal como se puede observar en la figura 13. En la primera etapa el *e-commerce* era más pequeño que un “catálogo de productos”, la información de productos y servicios era distribuida fácilmente y accedida desde cualquier lugar, a cualquier hora, obteniendo así lo más pequeño que puede tener un negocio, “presencia en la *web*”.

En la segunda etapa, las compañías obtienen la capacidad de realizar transacciones electrónicamente, es decir, los sitios *web* permiten la procura y venta de catálogos en línea.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

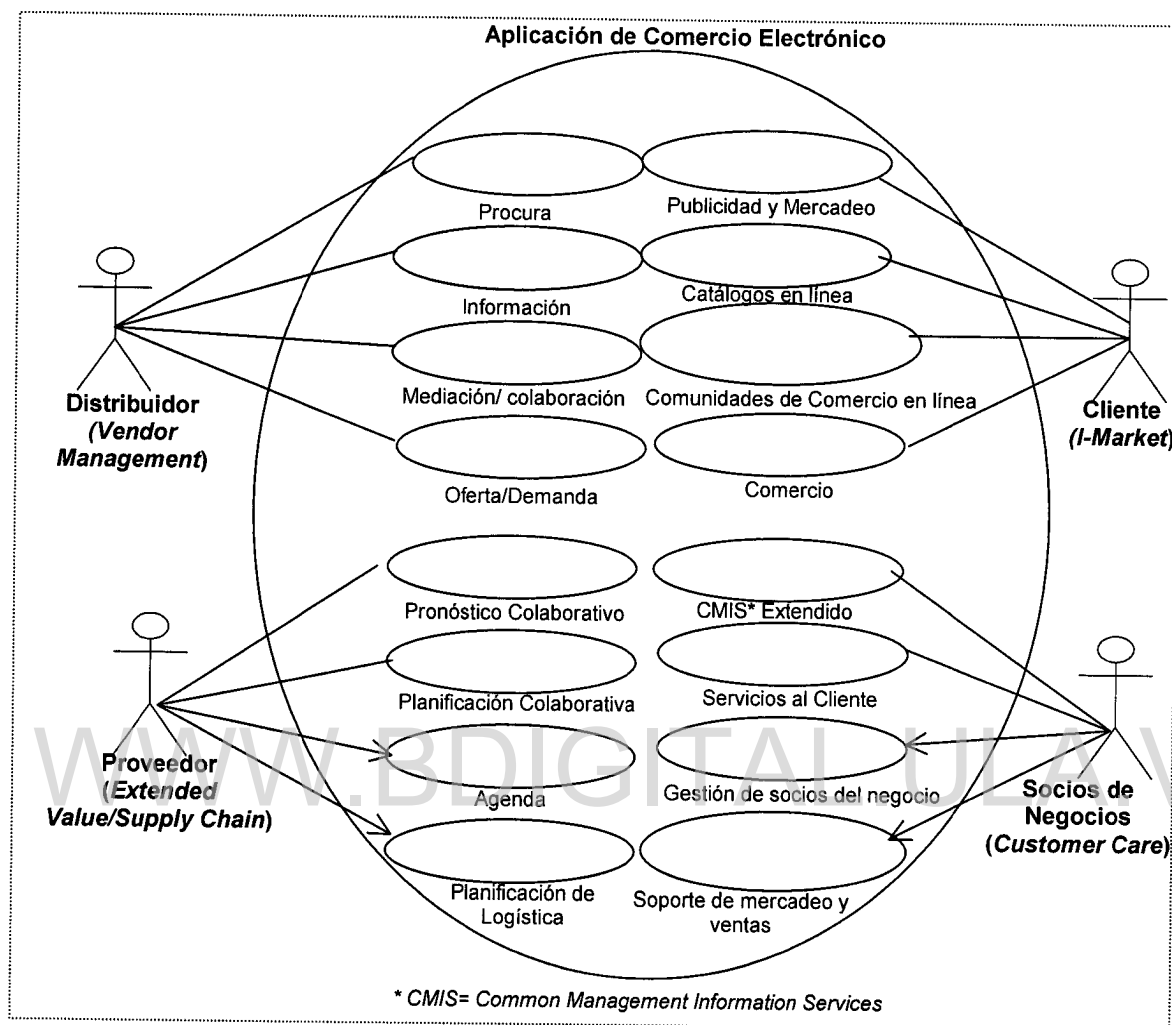


Figura 12. Diagrama de Casos de Uso de Una Aplicación de Comercio Electrónico

En la tercera etapa las compañías están rediseñando los procesos de negocio para que traspasen los límites de la empresa, erradicando así, la duplicación de procesos, eliminando la ineficacia y conectándose entre empresas. Se comienzan a crear las “corporaciones virtuales” para compartir las metas del negocio, tener una planificación común y herramientas de gestión de desempeño. Todo esto, con la ayuda de la Ingeniería de Procesos Interempresarial (IPE) (Stefan, 2000). Esta rama de la ingeniería, guía y orienta el diseño y la implementación de procesos de negocios “hyper-eficientes”, integrados en tiempo real con los proveedores y clientes y las mejores ofertas, constituyendo así el nuevo significado de las corporaciones del siglo XXI.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

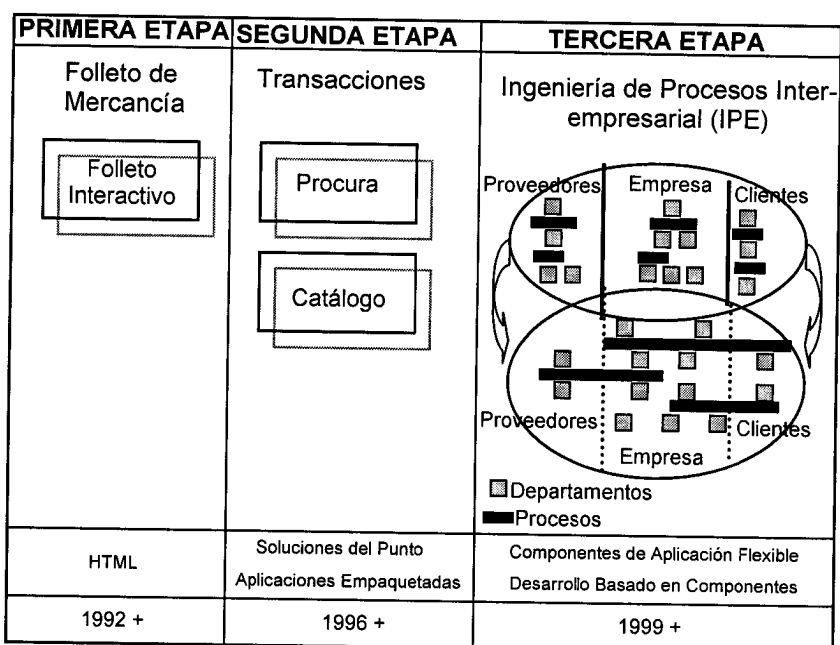


Figura 13. Etapas de las Aplicaciones de Comercio Electrónico (Fuente: *e-commerce: The Third Wave* (Meghan y Kiffer, 2000))

1.2.2. Arquitectura de una aplicación de comercio electrónico.

Las aplicaciones de comercio electrónico deben operar sobre una arquitectura que permita integrar los procesos del negocio y los componentes del *software*. El núcleo de las mismas está constituido por los procesos de negocio los cuales se encuentran empotrados en sistemas propietarios, sistemas de Planificación de Recursos Empresariales (ERP) y sistemas cliente/servidor.

El modelo de producto de una aplicación de comercio electrónico se muestra en la figura 14.

El modelo de producto describe la arquitectura de una aplicación de comercio electrónico. Los objetos de negocio claves en un *e-commerce* son: el servicio del perfil de usuario, el servicio de autenticación, el servicio de *workflow*, el servicio de integración de datos, el servicio de control de accesos y el servicio de notificación de eventos.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

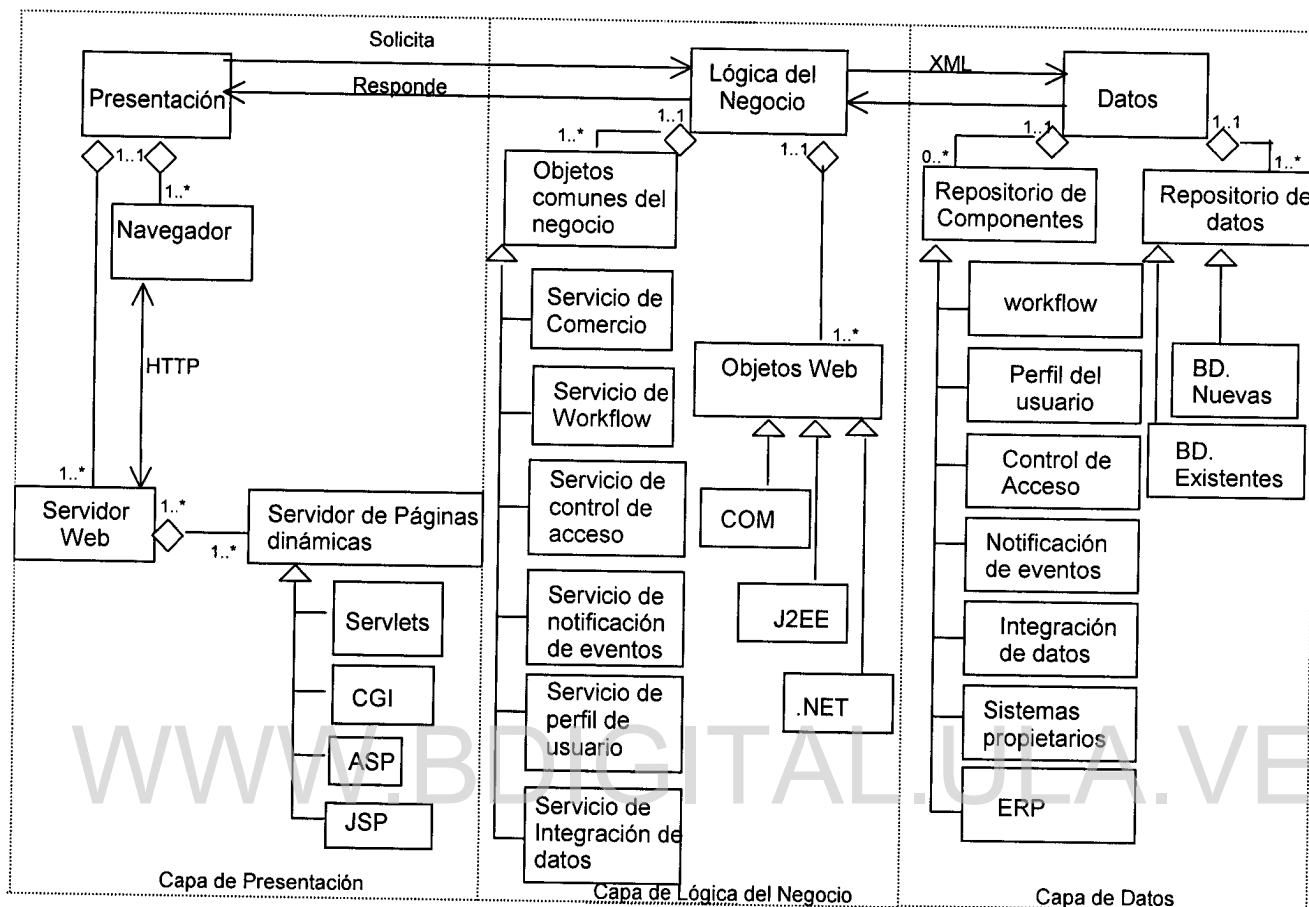


Figura 14. Modelo de Producto de una Aplicación de Comercio Electrónico

1.3. Aplicaciones de negocios Inteligentes (e-Business).

Las aplicaciones de negocios inteligentes (*e-Business*) son aplicaciones que incluyen procesos externos (*e-commerce*), procesos internos de producción, gestión de inventario, desarrollo de productos, gestión de riesgo, finanzas, desarrollo de estrategias, gestión del conocimiento y recursos humanos (Kruidof y Ferguson, 2003).

Las aplicaciones *e-business* se distinguen por poseer los siguientes atributos (Kuhn, Junginger y Bayer, 2000):

1. **Desarrollo Rápido:** Las aplicaciones *e-business* tienen que ser desarrolladas y adaptadas muy rápidamente, a través del uso de prototipos y cambios en las aplicaciones mientras están en producción.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

2. **Sistemas heterogéneos y complejos:** Para construir una aplicación *e-business* es necesario integrar diferentes tecnologías y sistemas, por lo cual es necesario realizar un modelado de calidad, con el fin de poder reducir la complejidad de este tipo de aplicaciones.
3. **Seguridad en las aplicaciones y datos:** La principal plataforma del *e-Business* es el *Internet*, lo cual es inseguro, por tal razón estas aplicaciones deben proporcionar mecanismos de seguridad muy sofisticados.
4. **Desempeño de la aplicación:** Las aplicaciones *e-Business* deben ser escalables, con el fin de garantizar la disponibilidad de los servicios *e-Business* en *Internet* y tener la capacidad de soportar la gran cantidad de clientes que pudiesen accederlas.

Tal como se explicó en el capítulo anterior, en las aplicaciones *e-business* pueden darse tres (3) tipos de relaciones diferentes (ver figura No. 15): *Business to Consumer (B2C)*, *Business to Employee (B2E)* y *Business to Partner (B2B)*.

En las aplicaciones *B2C* las actividades están centradas en la promoción, venta y atención al cliente, lo cual está relacionado directamente con los procesos de las aplicaciones de comercio electrónico (*e-commerce*). Los principales objetivos de las aplicaciones *B2C* son: proporcionar información acerca de la organización, productos y servicios, vender productos, liberar servicios, gestionar las relaciones con los clientes y comunicación y mercadeo.

En una aplicación *Business to Business (B2B)*. Un *Partner* (socio de negocio) es una persona o compañía proveedor de un bien o servicio, con el cual se crea una relación colaborativa para lograr las metas establecidas en el negocio. El principal objetivo de las aplicaciones *B2B* es la eficiencia en las operaciones y en los costos, lo cual se ve reflejado en las siguientes actividades: Integración del negocio, sucursales digitales, gestión de la cadena de suministro, *E-Procurement* y *Marketplaces*.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

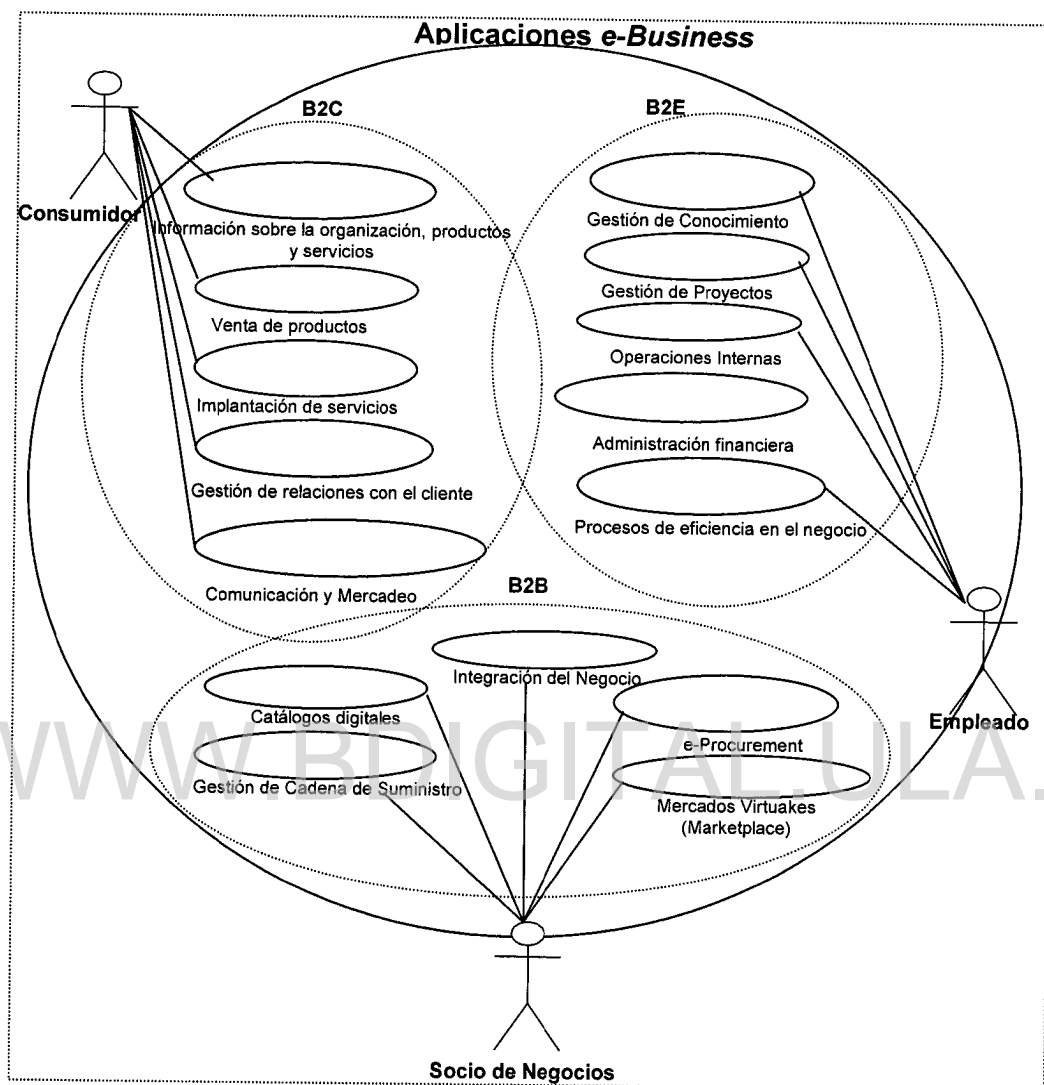


Figura 15. Diagrama de Casos de Uso de las Aplicaciones e-Business

Las aplicaciones *Business to Employee (B2E)* están dirigidas al capital humano de la organización, con el fin informar y educar al empleado para utilizar sus conocimientos en la automatización de los procesos del negocio. Las actividades que se desarrollan en las aplicaciones *B2E* son: Gestión de conocimiento, gestión de proyectos, operaciones internas, administración financiera y eficiencia en los procesos del negocio.

Es necesario aclarar, que las tecnologías relacionadas con *Internet* son usadas para fortalecer estos tipos de relaciones en las aplicaciones e-Business, con el fin de garantizar interacción entre cualquiera de las partes involucradas. Las

diferentes maneras de implantar este tipo de relaciones son: Internet para las aplicaciones B2C, *Extranet* para las B2B e *Intranet* para las B2E.

1.3.1. Etapas de las aplicaciones e-Business.

En la figura 16 se presentan las cuatro (4) etapas por las que ha pasado el desarrollo de aplicaciones e-Business (IICD, 2003). La primera etapa es la presencia básica, en la cual las organizaciones estratégicamente implantan un sitio *web* de su organización en *Internet*, con el fin de compartir su información y dar a conocer su cartera de productos y servicios. La segunda etapa es la perspectiva, en ésta las organizaciones exploran a Internet como un canal de distribución, involucra información extensiva, comienzo de relaciones interactivas, personalización de información y facilidades de búsqueda.

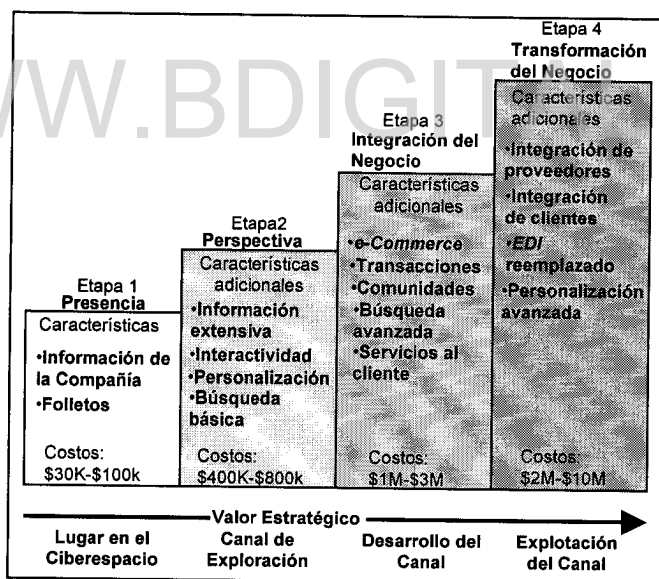


Figura 16. Etapas de las Aplicaciones de Negocios Inteligentes (E-Business). (Fuente: *Making Sense of E-business in Developing Countries* (IICD, 2003)).

La tercera etapa consiste en la integración de negocios, a este nivel la organización con los otros negocios y sus clientes de una manera más proactiva: ya se habla de actividades de *e-Commerce*, es decir, de transacciones de pagos y ventas, servicios al cliente búsquedas avanzadas y construcción de comunidades virtuales con los clientes.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

En la última etapa, ocurre una transformación del negocio, es realmente cuando las organizaciones comienzan a aprovechar todas las bondades que ofrece *Internet* como un canal de comunicaciones con sus clientes, proveedores y otros socios del negocio de manera integrada. A este nivel existen características avanzadas de personalización a través del uso del Intercambio Electrónico de Datos (EDI).

1.3.2. Arquitectura de las Aplicaciones e-Business.

Las aplicaciones de negocios inteligentes, desde el punto de vista arquitectónico, pueden ser descritas desde dos puntos de vista distintos (IICD, 2003):

1. El negocio.

Está conformada por la arquitectura de negocios y la arquitectura de información. La arquitectura de negocio proporciona una descripción de las estrategias del negocio y la información necesaria para implementarlas. La descripción del negocio está compuesta por el propósito o misión de la organización, objetivos, principios operacionales, procesos del negocio y aspectos de datos relevantes.

La descripción de la información está basada en la descripción del negocio. La descripción de la información está conformada por el *framework* que contiene los elementos que indican la forma de cómo será presentada la misma (estrategia de comunicación y flujo de información): comunicación, estructura de datos, estructura de la información, aplicaciones.

2. La Infraestructura Tecnológica (IT).

Desde el punto de vista de la IT una aplicación de negocio electrónico consiste en la descripción de una aplicación automatizada y la infraestructura tecnológica requerida. La descripción de los sistemas de información incluye los servicios de aplicación, componentes y la interacción humano-computador. La descripción de la infraestructura tecnológica incluye los recursos tecnológicos y la infraestructura física.

El modelo de producto de una aplicación de negocio inteligente (*e-Business*) se muestra en la figura 17. El modelo de producto describe la arquitectura y los

objetos fundamentales dentro de cada proceso del negocio. Estos son: la promoción, venta y atención al cliente, la integración del negocio y gestión de la cadena de suministro y la gestión de conocimientos y proyectos.

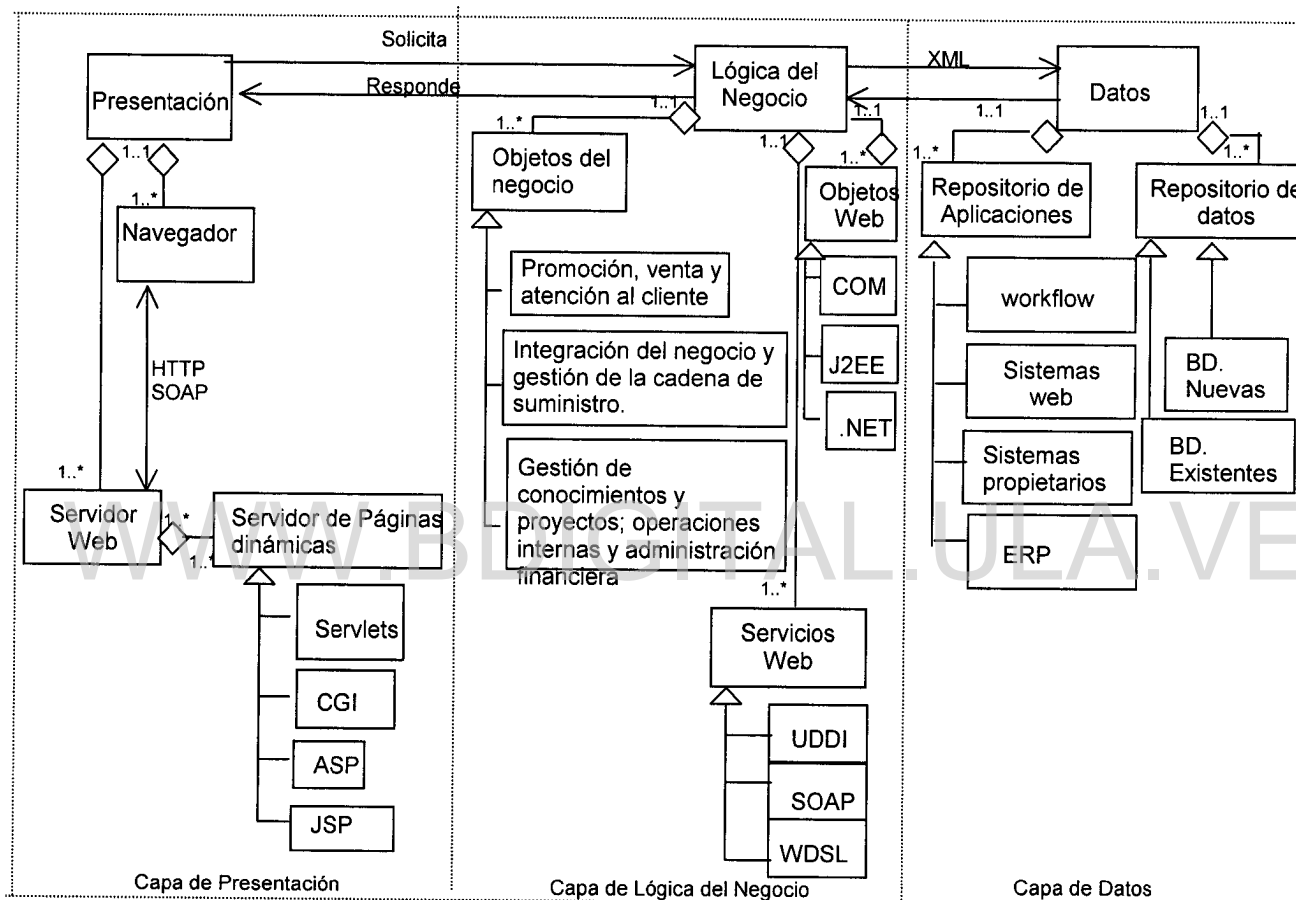


Figura 17. Modelo de Producto de una Aplicación de Negocios Inteligentes (*e-Business*)

1.4. SISTEMAS DE INFORMACIÓN WEB (SIW).

Un Sistema de Información Web (SIW) puede ser definido desde perspectivas diferentes: La primera es que un SIW es cualquier aplicación de software que depende de la *web* para su correcta ejecución (Gellersen, 1999), asimismo, un SIW puede ser definido como un sistema de información que proporciona facilidades para acceder datos complejos y servicios interactivos vía *web* (Gnaho, 2001), y finalmente, un SIW representa una sub-categoría de sistemas de información que típicamente soportan información en línea, tareas de rutina y recuperación a través

del autoservicio para un gran número de usuarios ocasionales sin depender de su ubicación geográfica. (Scharl, Gebauer y Bauer, 2001).

Para este caso particular se definirá como un sistema de información soportado por la computadora, que utiliza la tecnología del *World Wide Web* (*WWW*), y por tanto, es accedido a través de un navegador vía *Internet* o *Intranet*. Estos sistemas son capaces de soportar el trabajo de una organización y, por lo general, están fuertemente integrados con otros sistemas o elementos, tales como bases de datos, sistemas de procesamiento de transacciones, etc. Además, se pueden integrar con otras organizaciones y otros SIW.

Un SIW es utilizado tanto por los usuarios internos de la organización (empleados) como por usuarios externos (otras organizaciones o clientes), con el fin de solicitar algún requerimiento que es ejecutado por el mismo, como puede ser un trámite administrativo, un balance financiero, una consulta a base de datos, una compra de un artículo, la ejecución de un trabajo colaborativo o la consulta de un documento electrónico de la organización (ver figura 18).

1.4.1. Componentes básicos de un SIW.

.Un sistema de información *web* está conformado por los siguientes componentes de *software* (Torrealba, 1998):

- **Sistemas corporativos legados (*Legacy systems*)**. Sistemas operacionales que son usados para soportar las actividades que se realizan a diario dentro de la organización. Estos sistemas también incluyen aplicaciones más recientes tales como los sistemas ERP (*Enterprise Resource Planning*) y los Sistemas de Información Geográfica (SIG).
- **Sistema de almacenamiento corporativo (*Data Warehouse – DW*)**. Colección de bases de datos integradas orientadas al asunto diseñadas para soportar las funciones del sistema de soporte a las decisiones (DSS) donde cada unidad de datos es relevante en algún momento en el tiempo. El *DW* contiene datos atómicos y datos ligeramente resumidos.

Licencia Creative Commons:

- **Sistema de Groupware.** Herramienta que permite a un grupo de personas trabajar juntas a través del desarrollo de la comunicación (intercambio de mensajes electrónicos, *e-mail*), colaboración (compartir espacio de trabajo, conferencias) y coordinación (agregar la estructura de los procesos de negocio para la comunicación y colaboración, implementando las políticas de la empresa, logrando ambientes de desarrollo de trabajo o *workflow*). La plataforma *Groupware* tiene la capacidad de soportar el dinamismo entre la comunicación, la colaboración y la coordinación.

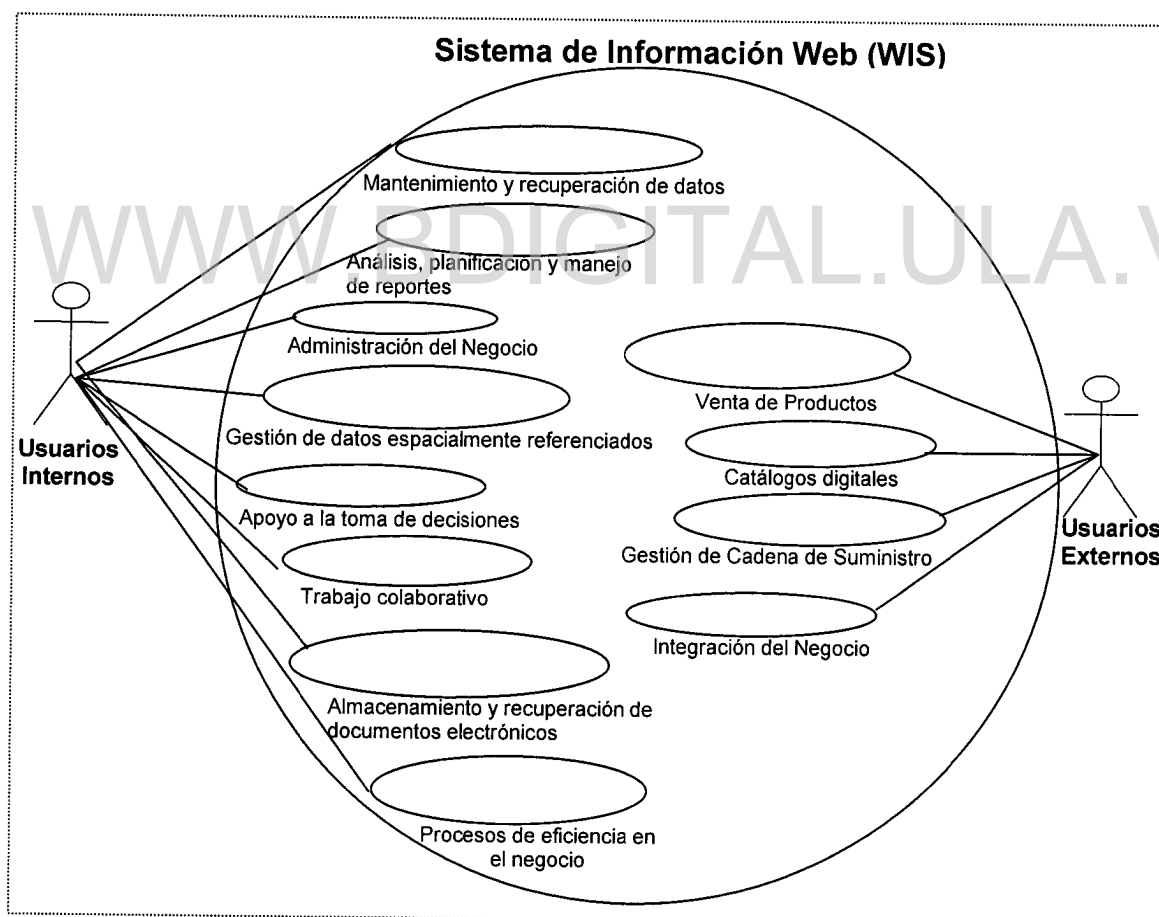


Figura 18. Diagrama de Casos de Uso de un Sistema de Información Web (WIS)

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

- **Sistema de Manejo Electrónico de Documentos (EDMS o *Electronic Document Management System*)**. Compila todos los documentos creados por medio del *software Groupware* y los organiza dentro de una base de datos.
- **Sistema de Autoría y Publicación *Web***. Sistema centralizado de manejo de documentos usado para el desarrollo *web*.
- **Sistemas de Inteligencia de Negocios (BIS *Business Intelligence Systems*)**. Un conjunto de conceptos y metodologías para mejorar la toma de decisiones en los negocios a través del uso de hechos y sistemas basados en hechos. Estos sistemas pueden estar ubicados dentro de la mayoría de las organizaciones actuales y poseen una serie de componentes que pueden ser reutilizados para la construcción de un SIW, ya que los mismos no se construyen desde cero.

WWW.BDIGITAL.ULA.VE

1.4.2. Arquitectura de un SIW

La figura 19 ilustra el modelo de producto de un WIS, el cual describe la arquitectura de este tipo de aplicaciones. La arquitectura de los WIS se fundamenta en los componentes de *software* que lo conforman, asegurando la integración de los mismos en una única interfaz de usuario a través de los componentes *web* (*J2EE*, *COM*, *.NET*). Estos se encuentran en la capa de lógica del negocio.

De la misma manera, en la capa de datos se concentran todos los repositorios de datos propios de los *software* que conforman al SIW resaltando la importancia del directorio de información, el cual proporciona acceso universal a los datos e información desde el SIW, ya que contiene de definición y ubicación de los datos (metadatos) que son manipulados por el mismo. Este directorio es producto de la integración de los esquemas de bases de datos y demás repositorios existentes.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

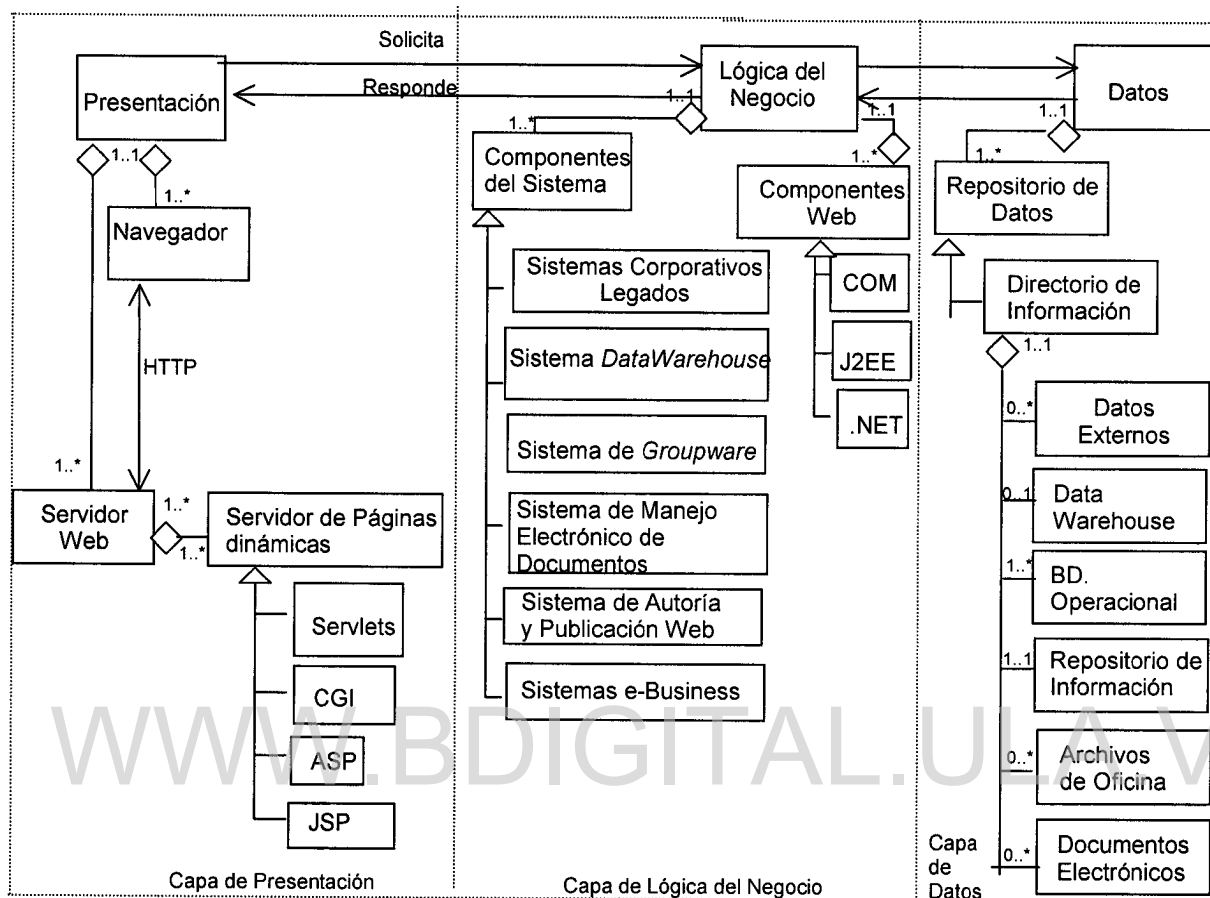


Figura 19. Modelo de Producto de un Sistema de Información Web (SIW)

1.5. GOBIERNOS ELECTRÓNICOS (e-Government).

Es definido como el uso de las tecnologías de información y comunicaciones para mejorar los servicios e información ofrecidos a los ciudadanos, aumentar la eficiencia y la eficacia de la gestión pública e incrementar sustantivamente la transparencia del sector público y la transparencia ciudadana (Valderrama, 2003).

Las soluciones de gobierno electrónico requieren considerar las relaciones existentes entre los diferentes actores que intervienen para que sus servicios sean efectivos, útiles y de calidad. Tal como se explicó en el capítulo anterior, las relaciones que pueden darse en este tipo de aplicaciones son: Gobierno a Ciudadanos (G2C), Gobierno a Negocios o Empresas (G2B), Gobierno a Empleados (G2E) y Gobierno a Gobierno (G2G) (ver figura 20).

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

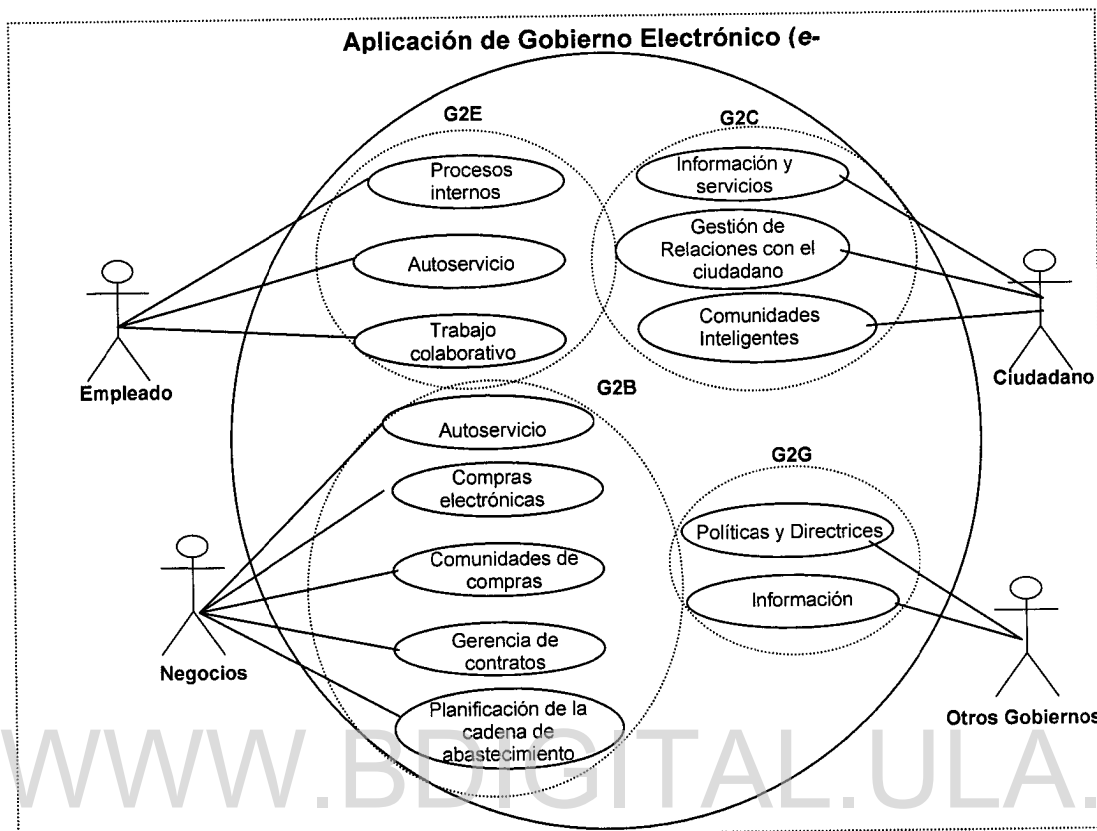


Figura 20. Diagrama de Casos de Uso de una Aplicación de Gobierno Electrónico (e-Government)

Las soluciones G2C cuentan con un área de atención para los ciudadanos que busca la manera de mejorar y acercar los servicios con el fin de satisfacer sus requerimientos, a través de aplicaciones de Gestión de Relaciones con el Ciudadano (*Citizen Relationship Management*), tales como: ventanas únicas (portales), centros o quioscos para interacción de los ciudadanos, gobierno en línea y comunidades inteligentes.

Las aplicaciones G2B están enfocadas a los negocios, y el mismo utiliza aplicaciones como Compras Electrónicas (*Internet Procurement*), Comunidades/grupos que hacen compras para todos los miembros de la comunidad o grupo (*Buying Communities*), Gerencia de Contratos, Planificación de la Cadena de Abastecimiento (*Supply Chain Planning*) y Autoservicio para Proveedores (*Suppliers Self-Service*).

Las soluciones G2E están relacionadas con los servicios internos donde se analizan y optimizan las operaciones y transacciones dentro del mismo gobierno. Para ello, el

G2E debe facilitar la captura, acceso y manipulación de la información para ser analizada. El G2E busca optimizar los procesos, utilizar efectivamente el portal (ventana única) para beneficiar al empleado del gobierno, hacer énfasis en el uso de autoservicios para los empleados. De tal forma que tengan acceso a la información todo el tiempo y desde cualquier lugar, busquen como aprovechar nuevas tecnologías como la telefonía celular y la computación móvil, utilicen más el “*Strategic Enterprise Management*” y sus indicadores “PKIs (*Public Key Infraestructure*)” para apoyar a la toma de decisiones, hagan énfasis en la colaboración y compartan información dentro del gobierno.

En las aplicaciones G2G cada orden del gobierno se relaciona dentro de si mismo y su vez con los otros niveles de gobierno y con sus principales proveedores. Estas relaciones implican la centralización, estandarización y redefinición del papel del organismo central y la forma en que interactúa en si mismo con las empresas y con los ciudadanos.

1.5.1. Evolución del Gobierno Electrónico. Generalmente una aplicación de gobierno electrónico atraviesa cada una de las siguientes etapas (ver fig. 21) con el fin de proporcionar mayor competitividad y ofrecer mayor retorno (Alanis, 2003).

- **Presencia primaria.** Aproximadamente el 85% de las aplicaciones de gobierno electrónico que existen en el mundo se encuentran en esta etapa; la cual consiste en la creación de un sitio de *Internet*. Pero no son ni siquiera portales con capacidad de gerencia de información, solo tienen páginas estáticas que ofrecen información básica.
- **Presencia mejorada.** En esta fase se encuentran el 5 ó 7% de los países, los cuales ofrecen mayor información al público en forma más avanzada. Se empieza a utilizar un portal con capacidad de gerencia de conocimientos, ya que, además, de poder ver información publicada del gobierno, los ciudadanos tienen la capacidad de buscar información e imprimir documentos que generan en el portal.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

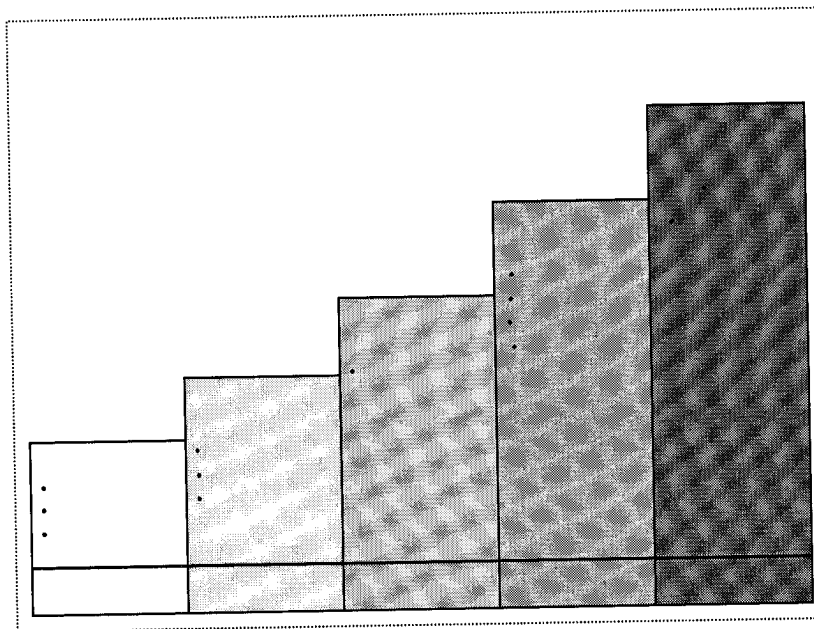


Figura 21. Evolución de las Aplicaciones de Gobierno Electrónico (e-Government). (Fuente: Adaptada de Diplomado de Gobierno Electrónico (Adanis, 2003))

- **Presencia interactiva.** En esta fase se puede solicitar e intercambiar información con el gobierno. Se empiezan a desarrollar sistemas donde se encuentran todos los datos sensibles de las diferentes áreas de la institución con la finalidad de tener la información reunida en un solo lugar “*business warehouse*” para que sea útil en el momento de tomar decisiones y tener sistemas de apoyo a decisiones e informes gerenciales.
- **Presencia transaccional.** En esta fase se empiezan a realizar transacciones monetarias en línea y servicios más integrados.
- **Presencia totalmente integrada.** Reúne los servicios de varios ministerios, secretarías, gobiernos o municipios, donde los procesos de negocios están coordinados y rediseñados y donde la infraestructura tecnológica requiere ser integrada. En la actualidad el único gobierno electrónico que ha alcanzado esta etapa es el de Singapur.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

1.5.2. Arquitectura de una Aplicación de Gobierno Electrónico:

Al igual que el resto de las aplicaciones *web* seleccionadas, una aplicación de gobierno electrónico (*e-Government*) se encuentra estructurada en tres capas: Presentación, lógica del negocio y datos (ver figura 22).

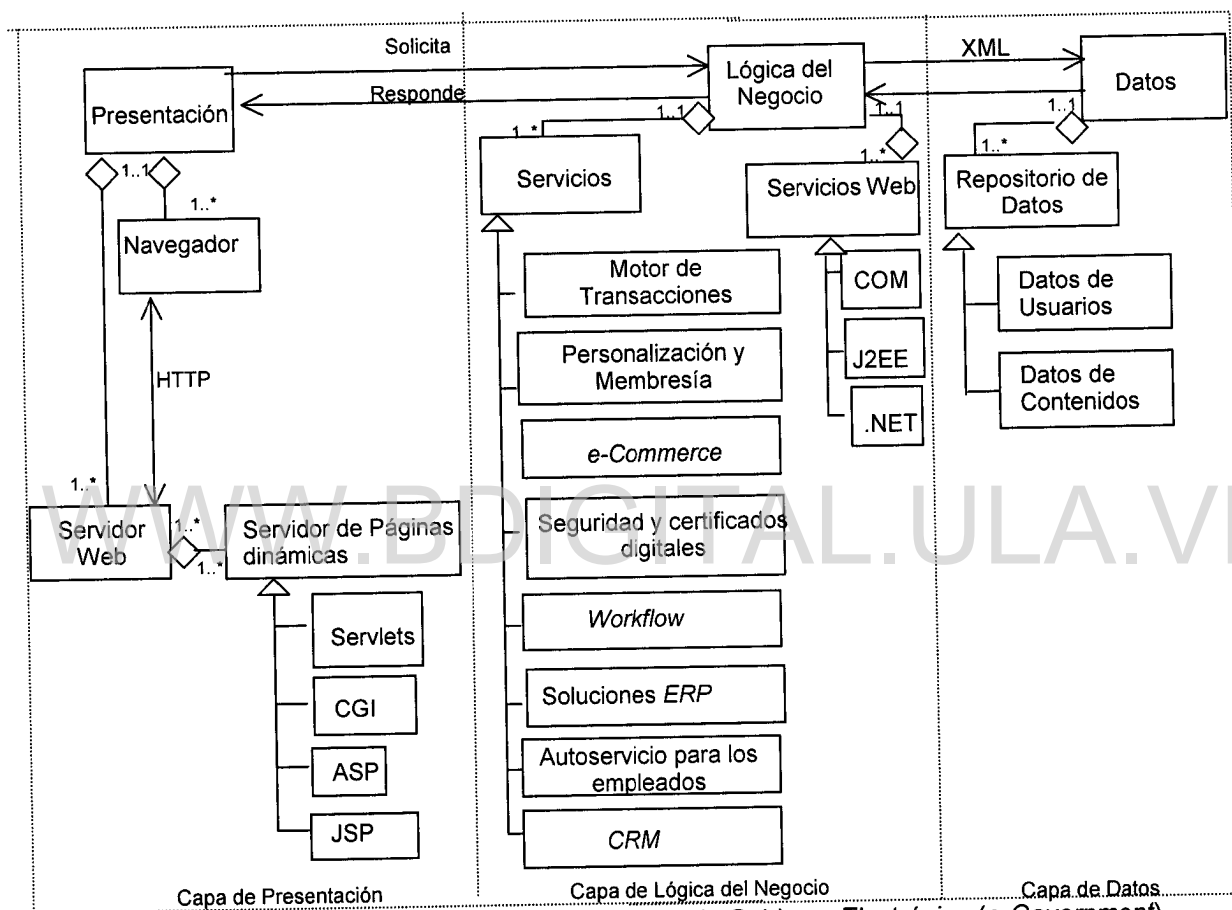


Figura 22. Modelo de Producto de una Aplicación de Gobierno Electrónico (*e-Government*)

La capa de lógica del negocio contiene todas las aplicaciones que generan los servicios ofrecidos a los usuarios, entre los cuales se mencionan: motor de transacciones, personalización y membresía, *e-Commerce*, seguridad y certificados digitales, *workflow*, soluciones ERP, autoservicio para los empleados y CRM.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

2. ASPECTOS COMUNES ENTRE LAS APLICACIONES *WEB* SELECCIONADAS:

Una vez descritas las aplicaciones *web* seleccionadas, se comparan con el fin de derivar los aspectos comunes entre ellas, los cuales se mencionan a continuación:

- Todas las aplicaciones *web* seleccionadas presentan una arquitectura de tres (3) capas: presentación, lógica del negocio y datos.
- La capa de presentación de cada una de las aplicaciones está conformada por el navegador y el servidor *web*, el cual a su vez contiene un servidor de páginas dinámicas, con el fin de mostrar al usuario únicamente la información requerida.
- Todas las aplicaciones requieren de objetos distribuidos (*COM*, *J2EE* o *.NET*) que garanticen la integración entre los diversos *software* que lo componen a través de los estándares *XML*.
- Todas las aplicaciones contienen en la capa de lógica del negocio todos los servicios que son ofrecidos a los usuarios para satisfacer sus requerimientos (compras, *workflow*, trabajo colaborativo, etc.).
- Una aplicación de gobierno electrónico en su última etapa de evolución (presencia totalmente integrada) contiene en su arquitectura soluciones de *e-Commerce* y portales.
- Una aplicación de Gobierno Electrónico (*e-Government*) es muy similar a una aplicación de *e-Business*, ya que ambas manejan las mismas funcionalidades y grupos de actores similares; lo que distingue una de la otra es, precisamente la organización para la cual son desarrolladas, debido a que el *e-Government* es para los gobiernos municipales, regionales y centrales y el *e-Business* está dirigido a las organizaciones privadas. En este caso, los principios y el marco legal hacen totalmente diferentes las reglas del negocio para cada tipo de aplicación.
- Uno de los componentes fundamentales de las aplicaciones *e-Business*, es el comercio electrónico. Debido a que este tipo de aplicaciones requiere de las transacciones de comercio electrónico para

Licencia Creative Commons.

los usuarios externos a la organización (proveedores, clientes y otros socios negocio).

- Las aplicaciones *e-Commerce*, *e-Business*, sistemas de información *web* y *e-Government* requieren de los portales, como un componente fundamental dentro de cada una de las mismas, con el fin de garantizar la interacción usuario/sistema a través de las páginas web que son presentadas a través del navegador.

En conclusión, se puede decir que, las aplicaciones *web* seleccionadas se agrupan de la forma como lo muestra el siguiente diagrama de capas (figura 23), y pueden ser descritas a través de un modelo de producto genérico, el cual se presenta en la siguiente sección.

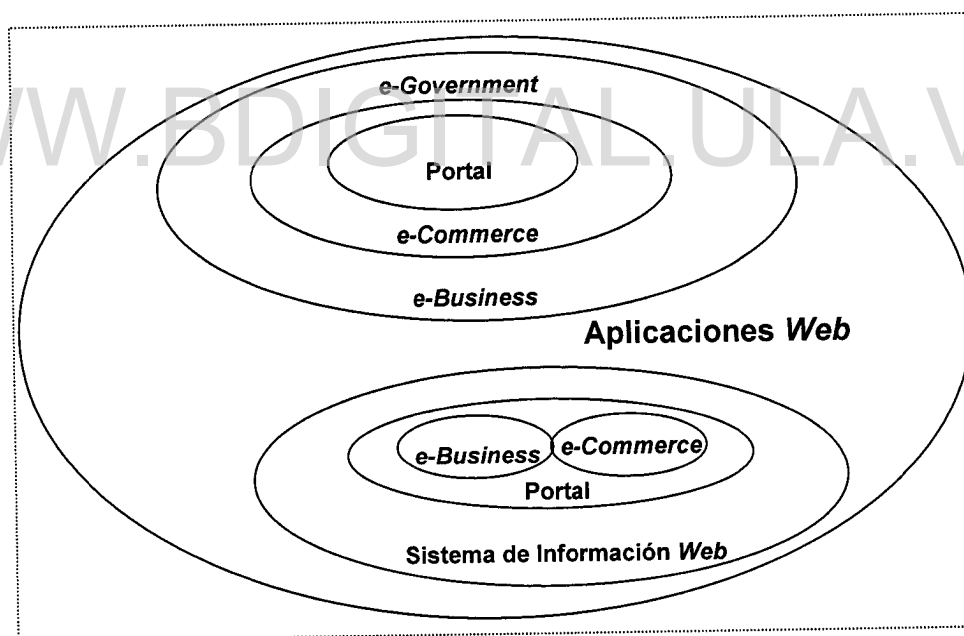


Figura 23. Diagrama de Capas que agrupa a las Aplicaciones Web Seleccionadas

3. MODELO DE PRODUCTO DE UNA APLICACIÓN WEB.

El modelo de producto que describe a una aplicación *web* genérica permite capturar los conceptos y patrones arquitectónicos genéricos de cualquier aplicación *web* (Montilva y Earríos, 2002). Estos conceptos al ser identificados contextualmente,

permiten definir un modelo de proceso flexible y ajustable a un contexto particular de desarrollo.

Con el objeto de conceptualizar una aplicación *web* y considerando que en la literatura actual, la mayoría de los modelos de producto están definidos implícitamente, se diseña un modelo de producto genérico que representa los conceptos más comúnmente encontrados en una aplicación *web* (fig. 15,17,19 y 21). Este modelo de producto está organizado en tres capas (figura 24) presentación, lógica del negocio y datos, debido a que este es el estilo arquitectónico que presentan todas las aplicaciones *web* estudiadas.

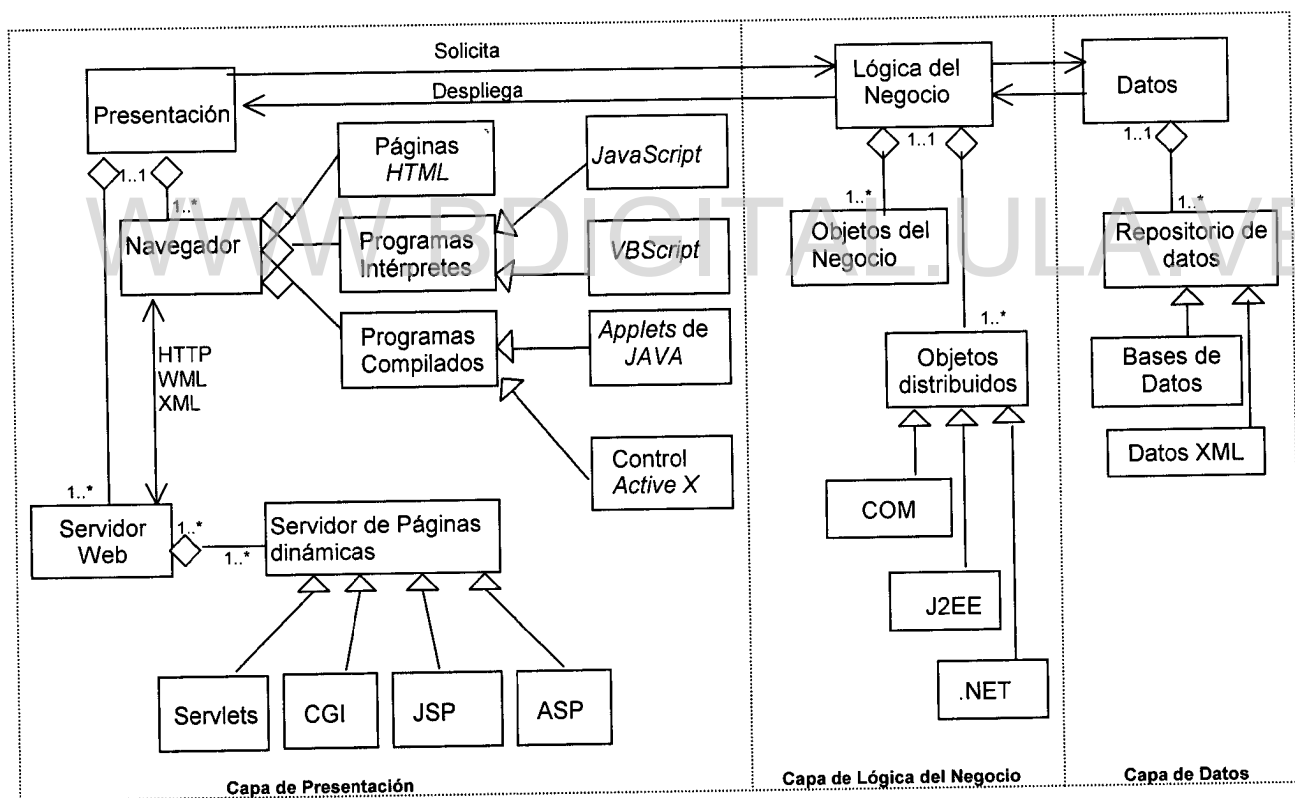


Figura 24. Modelo de Producto de una Aplicación *Web* en general

De acuerdo a este modelo de producto, se dice que el desarrollo de una aplicación *web* involucra la especificación, diseño, implementación y prueba de tres grupos de componentes distribuidos en cada una de las capas.

La Capa de Presentación, describe cómo la información debe ser desplegada. La misma está conformada por el navegador y el servidor web. El navegador

contiene páginas *HTML*, programas intérpretes (*JavaScript*, *VBScript*) y programas compilados (*Applets* de *JAVA*) para facilitar la validación de los datos, ayudar en la navegación y animar la interfaz del usuario. El servidor *web* contiene un servidor de páginas dinámicas que posee código embebido (*SERVLETS*, *CGI*, *ASP*, *JSP*) para preparar el formato de las páginas *HTML* que son requeridas por el navegador y solicitar a la capa de lógica del negocio los servicios que son demandados por el usuario.

La Capa de Lógica del Negocio crea la correspondencia entre los datos almacenados y la información que será presentada al usuario luego de haber solicitado un servicio a través del servidor *web*, es decir, actúa como un *middleware* para soportar la integración y comunicación entre los componentes (*J2EE*, *.NET*, *COM*). Además, esta capa contiene las clases de objetos que son propios del negocio (un carrito de compras, un cliente, etc.).

Por último, la Capa de Datos se encarga de describir la representación física de los datos. La misma está compuesta por los repositorios de datos que pueden ser bases de datos relacionales, orientadas a objetos y/o mixtas o datos en *XML*.

4. MODELO DE PRODUCTO DE UN SISTEMA DE NEGOCIOS

Dado que el modelado de negocios es la actividad central para diferentes áreas incluyendo, reingeniería de procesos, modelado empresarial, desarrollo de Sistemas de Información *Web* (SIW), comercio electrónico, Integración de Procesos y Gestión de Procesos de Negocios, el mismo es el punto de partida para desarrollar una aplicación *web*, ya que constituye la base para entender un negocio y obtener requerimientos de información de la organización.

Por tal razón, a continuación se presenta de manera muy general el modelo de producto de un sistema de negocios.

El modelo de producto de un sistema de negocios identifica y representa el conjunto de conceptos genéricos que deben ser encontrados en cualquier sistema de negocios (Barrios y Montilva, 2003) (ver figura 25). Estos conceptos se describen a continuación:

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

- **Metas del Negocio:** Constituyen la razón de ser de cualquier sistema de negocio. Estas son establecidas por actores de alto nivel para cumplir con los intereses del negocio y necesidades demandadas por el ambiente. Las metas pueden ser clasificadas de acuerdo a su alcance en: visión, misión y objetivos.
- **Procesos del Negocio:** Es un conjunto de actividades estructurado y jerárquico diseñado para alcanzar las metas del negocio. Los procesos pueden ser organizados en varias categorías. Primarios y procesos de soporte, las cuales son categorías que están asociadas a la cadena de valor de una empresa (Porter, 1985).
- **Actores:** Son los que ejecutan los procesos del negocio. Un actor puede ser una persona o una máquina capaz de ejecutar acciones o tareas definidas (ej., un componente de *software*, un programa de computadora o un Sistema de Información Empresarial). El objetivo principal, no es modelar los actores como persona, sino representar el rol que ellos juegan dentro del sistema de negocios. Cada rol es responsable de la ejecución de un conjunto de tareas, las cuales son parte de uno o más procesos del negocio. Los actores son asignados de acuerdo a sus roles a las unidades de negocios. Las **unidades de negocios** son organizadas en una estructura de negocios jerárquica.
- **Tecnología:** Es usada por los procesos del negocio para ejecutar sus actividades más efectivamente. Por ejemplo, tecnologías de información y comunicación, tales como redes de computadoras, Sistemas de Información Empresarial (EIS) y bases de datos son comúnmente utilizadas para soportar la ejecución eficiente de los procesos del negocio en muchas empresas.
- **Reglas del Negocio:** Delimitan los procesos del negocio. Un sistema de negocios debe adherirse, por ejemplo, a las regulaciones gubernamentales y a las leyes impuestas por el ambiente operativo. Similarmente, un sistema de negocios debe satisfacer las políticas, planes y estándares establecidos internamente por los gerentes de la empresa. Del mismo modo, los procesos son regulados o controlados por un conjunto de reglas del negocio de diferentes tipos.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

- **Objetos del Negocio:** Un objeto de negocio es una cosa concreta o abstracta que es relevante en el sistema del negocio. Por ejemplo, recursos tales como personas, dinero, insumos, equipamiento y datos son tipos de objetos de negocio que son requeridos para ejecutar procesos. Los clientes y proveedores son también otro tipo de objetos de negocio que participan en muchos procesos. Cada objeto del negocio tiene asociado un tipo que describe su estructura y comportamiento.
- **Eventos:** Un evento es una acción de muy corta duración que toma lugar dentro o fuera del sistema de negocios. Los eventos pueden ser programados o no programados (casual). La llegada de una orden de servicio es un ejemplo de un evento causado por un actor u otro sistema de negocios.

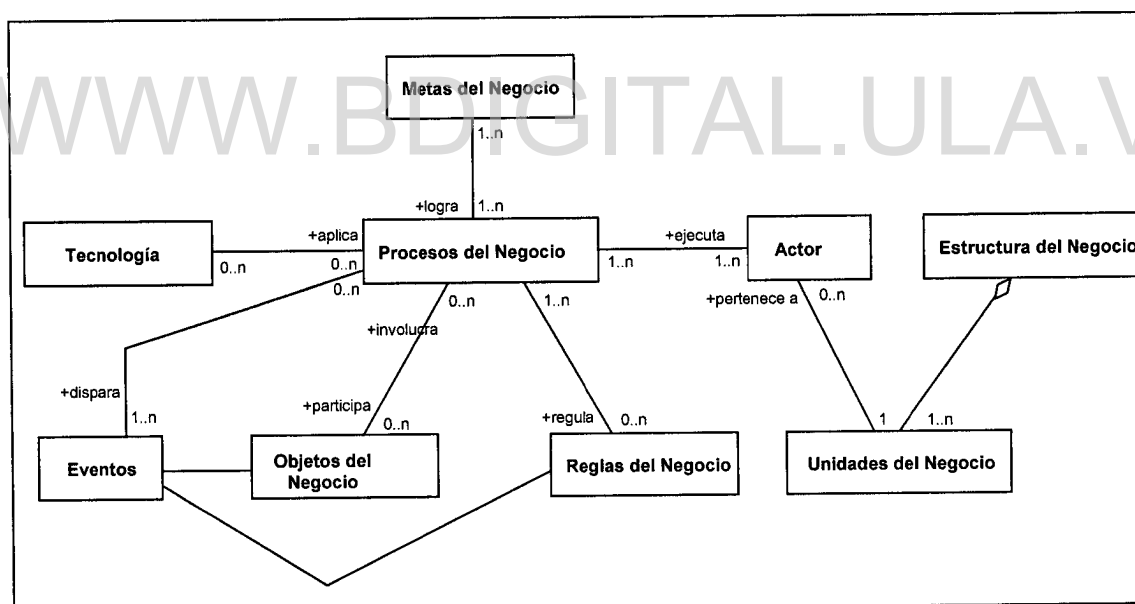


Figura 25. Modelo de Producto de un Sistema de Negocio. Fuente: *Business Modelling Through Roadmaps* (Barrios y Montilva, 2003)

5. MODELO DE PRODUCTO ASOCIADO AL CICLO DE VIDA DE UNA APLICACIÓN WEB

En la sección anterior se explicó de manera muy general el modelo de producto de un sistema de negocios, por ser el modelado de negocios la base para entender un negocio y obtener requerimientos de información de la organización.

A continuación se presenta el modelo de producto asociado al ciclo de vida de una aplicación *web*.

Este modelo de producto está conformado por tres (3) elementos fundamentales: el sub-modelo de gestión de proyecto, el sub-modelo de la aplicación *web* y el sub-modelo de implantación y mantenimiento de la aplicación *web* (ver figura 26).

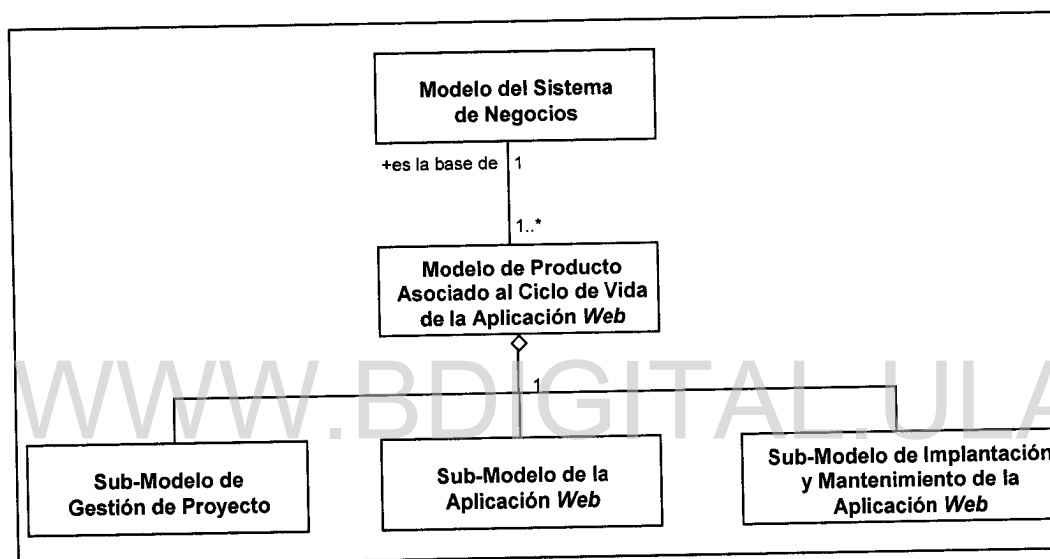


Figura 26. Modelo de Producto Asociado al Ciclo de Vida de la Aplicación *Web*.

A continuación, se describen brevemente cada uno de los sub-modelos que constituyen al modelo de producto asociado al ciclo de vida de la aplicación *web*..

5.1. SUB-MODELO DE GESTIÓN DE PROYECTO

La gestión de un proyecto tiene como objetivo principal la definición de los procesos mediante los cuales se establece el proyecto, el cómo este es coordinado y el cómo hacer uso efectivo de los recursos (tiempo, personas, dinero) para lograr los objetivos del proyecto.

La misma se centra en cuatro (4) elementos fundamentales: *personal*, *producto*, *proceso* y *proyecto* (Pressman, 2002). El modelo de producto de la gestión de un proyecto identifica y representa cada uno de estos elementos, los cuales se describen a continuación (ver figura 27):

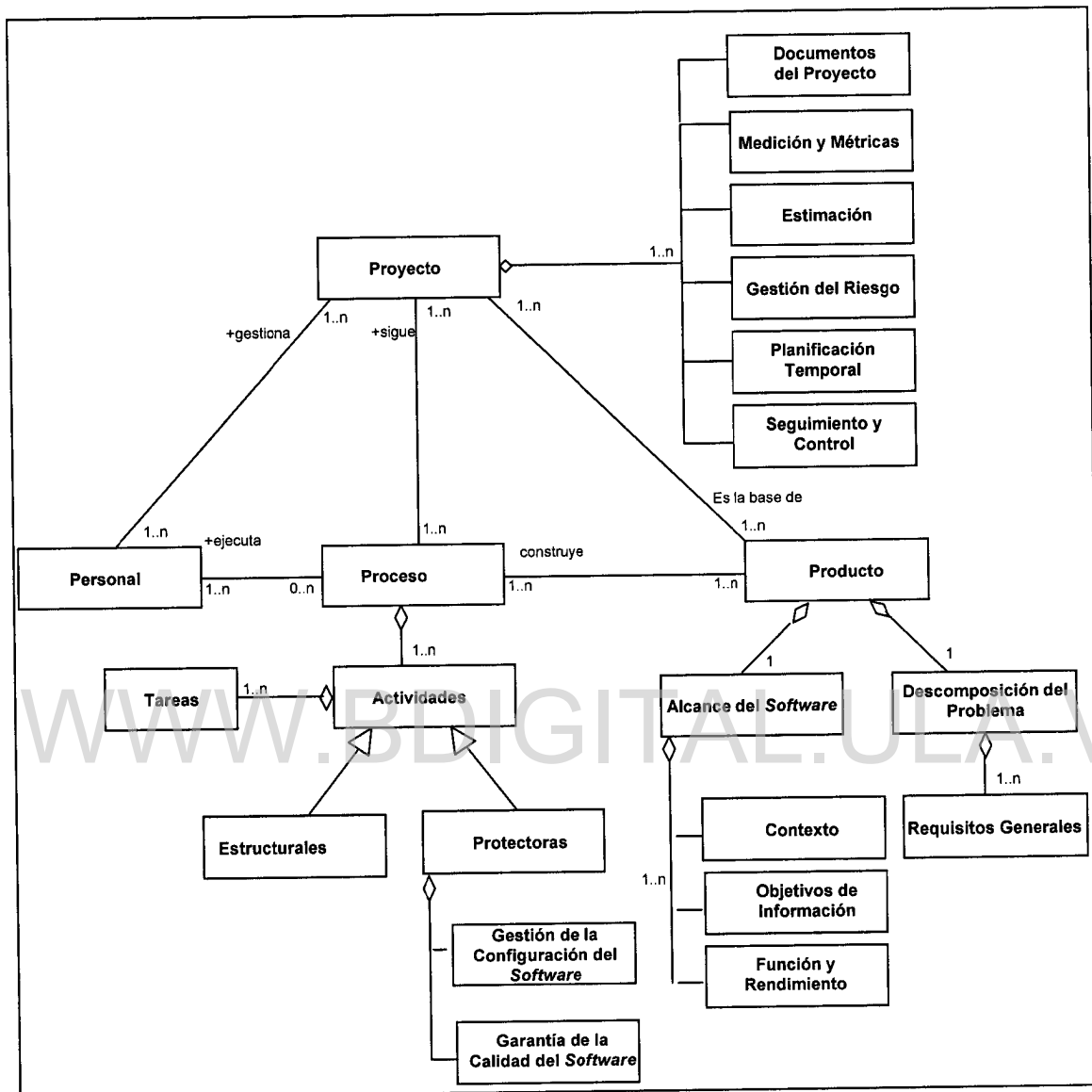


Figura 27. Sub-Modelo de Gestión de Proyecto.

- **Personal:** Constituye el principal factor de éxito en el desarrollo de un proyecto. El personal debe organizarse en equipos eficaces, motivados para hacer un software de alta calidad y coordinados para alcanzar una comunicación efectiva. El personal lo componen los gerentes, el grupo de desarrollo, los clientes, y los usuarios finales.
- **Proceso:** Proporciona la estructura para establecer un plan detallado para el desarrollo de *software*. Está conformado por las actividades estructurales y protectoras. Las actividades estructurales se pueden aplicar a todos los

proyectos de *software*, sin tener en cuenta su tamaño y complejidad; estas pueden adaptarse a las características del proyecto de *software* y a los requisitos del equipo del proyecto. Las actividades protectoras cubren el modelo de proceso. Las actividades protectoras son independientes de las estructurales y tienen lugar a lo largo del proceso.

- **Producto:** Es la base fundamental para definir unas estimaciones razonables del costo: una valoración efectiva del riesgo, una subdivisión realista de las tareas del proyecto. El producto esta descrito a través del ámbito del producto y la descomposición del problema. El ámbito del producto define el contexto los objetivos de información y la función y rendimiento. La descomposición del problema está constituida por los requisitos del producto, los cuales son considerados la base de trabajo del equipo de desarrollo.
- **Proyecto:** Está constituido por las métricas de calidad, estimación, lista de riesgos, plan del proyecto e informes de seguimiento y control. Las métricas de calidad proporcionan una visión profunda del proceso, de la comprensión del proyecto y de los atributos con que debe cumplir el producto para ser considerado de calidad. El informe de riesgos contiene la probabilidad de ocurrencia de cada riesgo y el daño que puede causar si ocurre. El plan del proyecto consiste en una tabla que indica las tareas a desarrollar, las funciones a implementar, y el costo, esfuerzo y tiempo necesario para la realización de cada tarea. El seguimiento y control es un informe que contiene los avances y logros del proyecto, tomando como base la planificación del mismo.

5.2. SUB-MODELO DE LA APLICACIÓN WEB

El producto que se obtiene luego de desarrollar la aplicación *web*, es la aplicación *web* operativa y probada. El sub-modelo de producto de una aplicación *web* identifica y representa todos los elementos asociados a los procesos de definir y especificar requerimientos, diseñar, implementar y probar la aplicación *web* (ver figura 28)

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

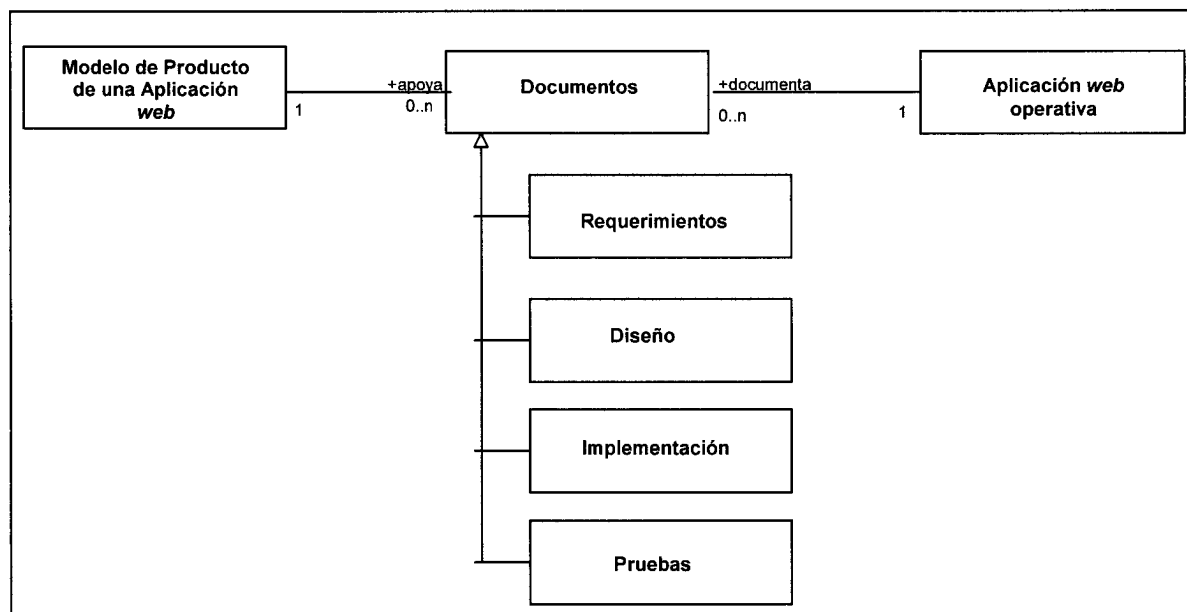


Figura 28. Sub-Modelo de la Aplicación Web.

Los principales elementos que constituyen el sub-modelo de una aplicación web son: documentos de requerimientos, documentos de diseño, modelo de producto de una aplicación web, la aplicación operativa y los documentos de la implementación y las pruebas.

Los documentos de requerimientos constituyen la base del modelo de producto de una aplicación web. Los mismos están conformados por el documento de definición de requerimientos y el documento de especificación de requerimientos.

Los documentos de diseño describen la arquitectura de la aplicación web. Los mismos están conformados por los documentos del diseño de las capas de presentación, lógica del negocio y datos.

El modelo de producto de una aplicación web describe los conceptos necesarios que deben estar presentes en la misma. El mismo fue descrito en este capítulo en la sección 3. La aplicación web operativa es la aplicación ejecutable, lista para ser implantada en su ambiente de ejecución.

La documentación de la implementación contiene todo lo relacionado al enlace de los componentes de las capas que conforman una aplicación web y la integración con cualquier otra aplicación existente.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

5.3. SUB-MODELO DE IMPLANTACION Y MANTENIMIENTO DE LA APLICACIÓN WEB

La implantación y mantenimiento de una aplicación *web* consiste de la instalación, mantenimiento, operación y soporte y adiestramiento de los usuarios y operadores que manipularán la misma. El principal producto que genera es la aplicación *web* instalada y operando, así como los operadores y usuarios adiestrados y la documentación del mantenimiento. El modelo de implantación y mantenimiento de una aplicación *web* identifica y representa todos estos elementos (ver figura 29).

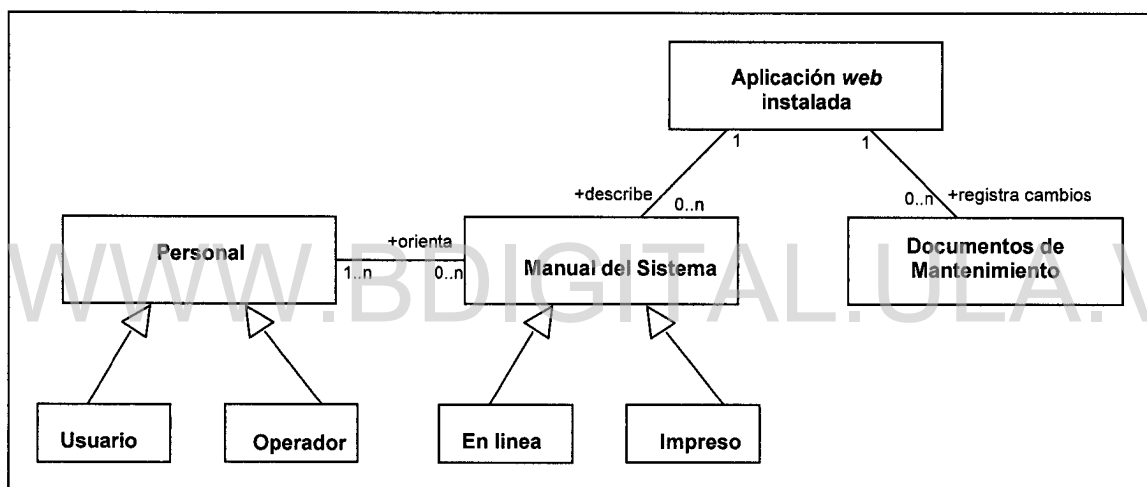


Figura 29. Sub-Modelo de Implantación y Mantenimiento de la Aplicación Web.

La aplicación *web* instalada y operada en su ambiente de ejecución constituye el producto fundamental de la implantación y mantenimiento de la misma. El personal adiestrado puede ser clasificado en usuario u operador. Este puede ser guiado u orientado por los manuales del sistema, los cuales describen las características y la forma de operar la aplicación *web*. Estos manuales pueden ser en línea o impresos. Los documentos de mantenimiento registran todos los cambios realizados a la aplicación *web*. Los mismos no serán detallados en este proyecto.

6. CONCLUSIÓN.

Dado que el propósito de esta investigación es diseñar un método situacional, es decir, que se adapte a situaciones particulares de proyectos para el desarrollo de

aplicaciones *web*, la construcción de un modelo de producto genérico de una aplicación *web* es parte fundamental y básica de este método situacional.

El modelo de producto presentado permite mostrar la arquitectura base y el conjunto de conceptos necesarios para capturar los aspectos comunes de las diferentes aplicaciones *web*, los cuales podrán ser utilizados en el desarrollo de diversos proyectos.

El modelo de producto asociado al ciclo de vida de una aplicación *web* permite mostrar los elementos más importantes que están asociados con el ciclo de vida de una aplicación *web*. El mismo está conformado por tres (3) sub-modelos: el sub-modelo de gestión de proyectos, el sub-modelo de la aplicación *web* y el sub-modelo de implantación y mantenimiento de la aplicación *web*.

El modelo de proceso correspondiente a nuestro método situacional será el objeto de los capítulos siguientes.

WWW.BDIGITAL.ULA.VE

CAPITULO 4

GENERALIZACION DE LOS MODELO DE PROCESOS PARA DESARROLLAR APLICACIONES WEB

Tal como se explicó en el capítulo 2, el modelo de proceso describe las actividades que el grupo de desarrollo debe seguir para desarrollar una solución de *software*. En este capítulo se presenta la generalización de modelos de proceso en la cual se recopilan los aspectos comunes de los modelos de proceso asociados los métodos seleccionados.

De las aplicaciones *web* estudiadas en el capítulo 2, se han seleccionado cinco (5), las cuales sirvieron de base para la construcción del modelo de producto genérico de una aplicación *web*, presentado en el capítulo anterior. Por tal razón, fue necesario seleccionar otros métodos distintos a los evaluados en el capítulo 2; debido a que los métodos evaluados deben tener relación con el dominio de aplicación para el cual son desarrolladas las aplicaciones *web* seleccionadas.

Los aspectos comunes de los modelos de proceso asociados a estos métodos constituirán la base necesaria para construir el modelo de proceso de nuestro método situacional para el desarrollo de aplicaciones *web*.

Los métodos seleccionados permiten desarrollar: portales, comercio electrónico, negocios inteligentes, sistemas de información *web* y gobiernos electrónicos.

1. METODOS PARA DESARROLLAR LAS APLICACIONES WEB SELECCIONADAS

De las aplicaciones *web* estudiadas en el capítulo anterior, se seleccionaron las cinco (5) siguientes: portal, *e-Commerce*, *e-Business*, Sistema de Información Web (SIW) y *e-Government*. Los aspectos comunes entre los dominios de aplicación para los cuales son desarrolladas y la necesidad que existe hoy en día de contar con este tipo de aplicaciones *web*, fueron los criterios utilizados para realizar esta selección. En este caso particular, se han seleccionado los métodos que tienen

relación con el dominio de aplicación para el cual son desarrolladas las aplicaciones *web*, ya que cada aplicación tiene un propósito definido, lo cual permite diferenciarlas entre sí, es decir, cada una de ellas ofrece servicios de acuerdo a las necesidades de los usuarios propios de cada dominio, sin tomar en cuenta el grado de complejidad de dicho servicio. A continuación se describen cada uno de ellos.

1.1. MÉTODO PARA EL DESARROLLO DE PORTALES

Este método es presentado por Oliveira (Oliveira, 2000). El mismo considera algunos factores claves tales como:

- Los diferentes puntos de vista de los actores: desarrolladores, clientes y usuarios.
- El rol de cada actor en el proyecto
- El valor del diseño y del contenido de la solución final
- La actualización constante para el usuario.

El método está estructurado en cuatro (4) fases: estrategia y tácticas, diseño y desarrollo del contenido, producción y desarrollo técnico y lanzamiento y mantenimiento.

En la **fase de estrategia y táctica**, los desarrolladores y el cliente trabajan juntos para definir el problema. Los diseñadores ven primero las necesidades del negocio, para luego concentrarse en el contexto *web*. Luego trabajan con el cliente para decidir cual de las opciones se deberá implantar para conducir a una solución *web* exitosa.

En esta fase se presentan la investigación y los documentos que deben conducir a un portal exitoso. Las principales actividades a realizar son:

1. **Planificación estratégica.** Estrategia del mercado, perfiles del consumidor, resumen estratégico, medidas de éxito y estrategia funcional.
2. **Planificación táctica.** Características del portal, mapa del portal, resumen creativo, plan del contenido, documentos técnicos y el plan del proyecto.

En la **fase de diseño y desarrollo del contenido**, los diseñadores usan el resumen creativo para explorar el espacio del diseño y la tecnología necesaria para

Licencia Creative Commons:

realizarla. El cliente aprueba el diseño y los desarrolladores completan las áreas del mismo. Las actividades que se ejecutan son:

1. **Expansión.** Generación de ideas, lenguaje del diseño, cuadrícula del diseño.
2. **Contracción.** Elección de la opción definitiva, diseño interactivo, diseño modular, desarrollo del contenido y diseño de la ingeniería de sistema
3. **Preproducción.** Construcción del prototipo, plan de lanzamiento.

En la **fase de producción y desarrollo técnico**, se tienen a tres equipos trabajando simultáneamente: desarrollo, ingeniería y cliente. La responsabilidad principal del productor es facilitar la comunicación entre los líderes de estos tres equipos y mantener el proyecto en seguimiento con una política de anticipación y detección temprana de errores. Las actividades a realizar en esta etapa son:

1. **Producción.** Proceso de producción, hospedaje, seguimiento del trabajo, control de recursos, control de cambios y versiones y automatización de recursos.
2. **Validación.** Ingeniería, prueba de bases de datos y pruebas del portal.

La **fase de lanzamiento y mantenimiento**, comienza con una lista de actividades y una fecha oficial de lanzamiento. Las bases de datos y los elementos visuales del portal están juntos y los dueños del contenido están agregando el contenido del día. Los desarrolladores prueban y navegan el portal antes de llevarlo al servidor *web*. El equipo de calidad libera el portal. Esta fase tiene dos actividades: el lanzamiento del portal y el mantenimiento en marcha durante la vida del mismo. La figura 30 ilustra el modelo de procesos asociado al método.

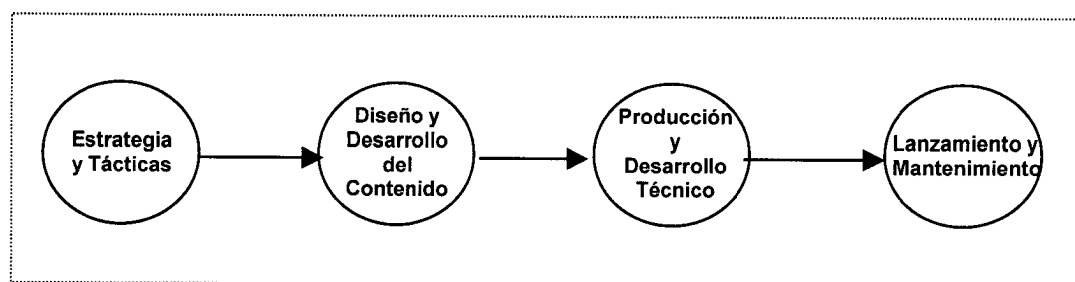


Figura 30. Modelo de Procesos asociado al Método para el Desarrollo de un Portal. (Fuente: Método para Web Sites y Portales: Una Guía para el Desarrollo Exitoso (Oliveira, 2000))

1.2 MÉTODO PARA EL DESARROLLO DE APLICACIONES DE COMERCIO ELECTRÓNICO (*e-Commerce*)

Este enfoque es presentado por Peter Fingar (Fingar, 2000), el mismo es denominado **Marco de Referencia basado en componentes para E-Commerce**. Está dividido en cuatro fases: estrategia de negocio para *e-Commerce*, procesos de negocio Inter.-empresariales, aplicaciones basadas en componentes e infraestructura tecnológica. La figura 31 ilustra el modelo de procesos asociado a este método.

La estrategia de negocio involucra a los expertos del dominio, quienes definen las iniciativas que se lograrán basados en el análisis de las fortalezas, debilidades, oportunidades y amenazas de la compañía. La estrategia de negocio determina cual es el problema a ser solucionado, generando requerimientos, metas y restricciones.

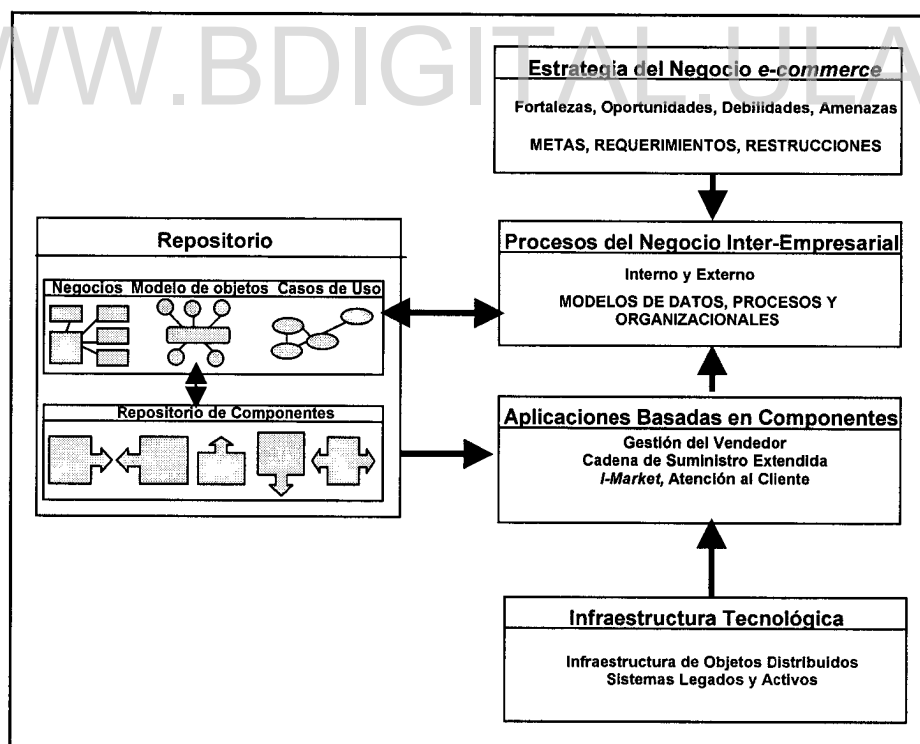


Figura 31. Modelo de Procesos asociado al Método para el Desarrollo de *e-Commerce*.
(fuente: *Component-Based Framework for e-commerce* (Fingar, 2000). Adaptación: Mendoza, 2004)

La ingeniería de procesos Inter.-empresarial define como los requerimientos son satisfechos a través de organizaciones nuevas o modificadas,

procesos y datos compartidos entre los socios del negocio de la compañía, proveedores y clientes.

Las aplicaciones basadas en componentes implementan el proceso de negocio diseñado nuevamente a través de **la infraestructura tecnológica**. Todo el proceso es guiado por un enfoque arquitectónico que hace posible el desarrollo rápido de soluciones de negocio a través del reuso de todos los elementos en un repositorio en crecimiento de modelos y componentes de *software*.

1.3. MÉTODO PARA EL DESARROLLO DE APLICACIONES *e-Business*

Este método es definido para proyectos de desarrollo de aplicaciones de *software e-Business* (Boban y Sertic, 2003). El mismo está dividido en seis (6) fases: Análisis del ambiente del negocio, adaptación del proceso de desarrollo del *software*, preparación del proceso de desarrollo de *software*, definición de requerimientos del negocio, proceso de desarrollo de la aplicación, evaluación del status del proyecto (ver figura 32).

El modelo de procesos comienza con el **análisis del ambiente del negocio**. Esta fase es fundamental para la implementación y adopción del proceso de desarrollo del *software*, ya que el mismo debe ser **adaptado al ambiente del negocio** en el cual las aplicaciones *e-Business* serán usadas. Su principal objetivo es generar una descripción precisa de los procesos del negocio que las aplicaciones *e-business* soportarán.

Luego del análisis del ambiente del negocio, se comienzan a ejecutar las actividades enfocadas a la organización del proyecto y a la **definición de requerimientos del negocio**. La definición de requerimientos del negocio es muy importante, ya que una vez definidos los mismos, éstos serán la guía de todo el proceso de desarrollo de la aplicación.

Por otro lado, las actividades orientadas a la organización del proyecto, tienen como objetivo el análisis y **preparación del proceso de desarrollo de software**, de acuerdo a las necesidades globales del proyecto. **El proceso de desarrollo de la aplicación** comienza después de que el mismo es preparado y los requerimientos

han sido claramente definidos. Durante esta actividad el equipo de proyecto es organizado de acuerdo al proceso de desarrollo de *software* preparado y los requerimientos del *e-business*. Los roles son distribuidos entre los miembros del proyecto, tomando en cuenta sus competencias, actividades definidas y ejecutadas en la fase de preparación del proceso de desarrollo y las nuevas herramientas que serán usadas durante el ciclo de desarrollo.

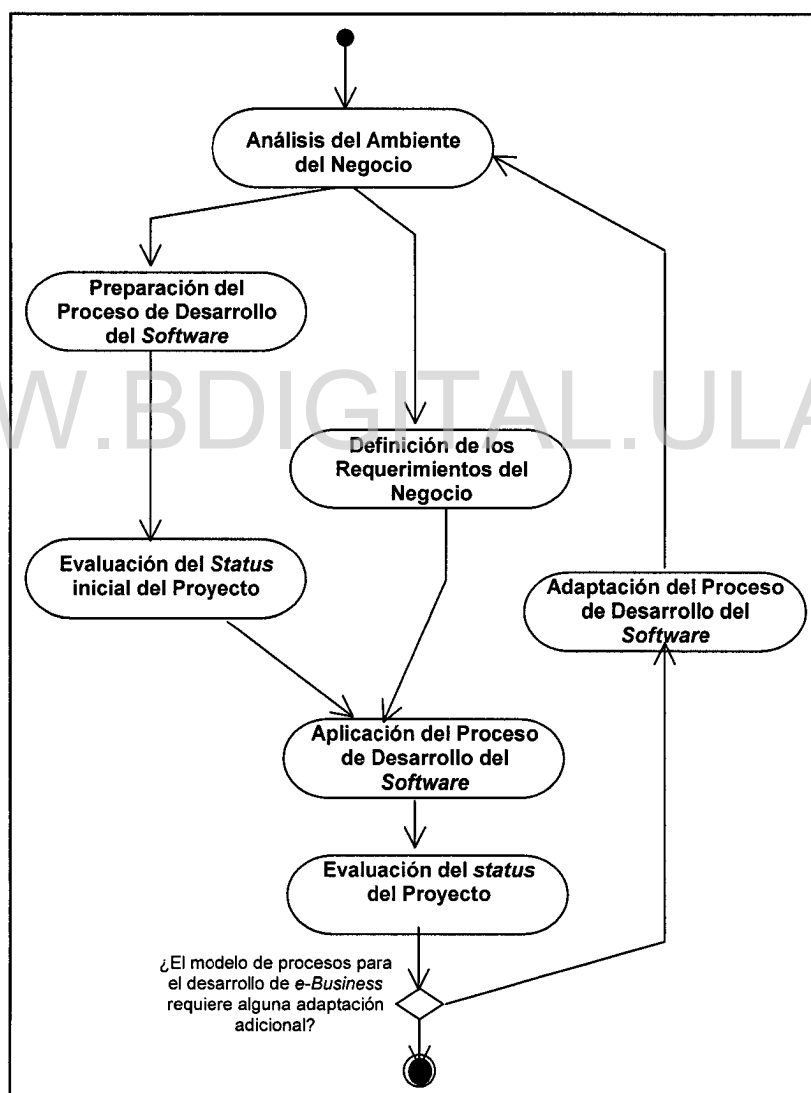


Figura 32. Modelo de Procesos asociado al Método para el Desarrollo de *e-Business*. (Adaptación de: *Development of Software for e-business* (Boban y Sertic, 2003))

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

En definitiva, el proceso de desarrollo de la aplicación podría ser implementado incrementalmente, en pequeños pasos de acuerdo a las capacidades del equipo de proyecto.

Luego de aplicado el proceso de desarrollo de *software* **el status del proyecto es evaluado**. Esta evaluación se realiza en función a la completitud del proceso de desarrollo del *software*, con el fin de identificar aquellas áreas donde el proceso no ha sido aplicado y las áreas donde el proceso de desarrollo del *software* debe ser mejorado.

Los resultados de esta evaluación determinan si los requerimientos del negocio han sido satisfechos y si el modelo de proceso para el desarrollo del *e-business* requiere alguna adaptación adicional.

El método presentado persigue como objetivo fundamental la creación de un ambiente adecuado Para el desarrollo de aplicaciones *e-business*, el misma busca el desarrollo de aplicaciones alineadas con los procesos del negocio y definiendo un proceso de desarrollo de acuerdo a las características particulares de cada proyecto.

1.4. MÉTODO PARA EL DESARROLLO DE SISTEMAS DE INFORMACIÓN WEB (WIS): Modelo de Ciclo de Vida W y sus Metodologías Asociadas para el Desarrollo de Sitios Web Corporativos (Serle y Chen, 2001).

El propósito de este método es guiar al grupo de desarrollo en la construcción y mantenimiento de *intranets*, sitios *web*, sitios transaccionales y sistemas de información *web* (WIS), considerados todos como sitios *web* corporativos. El objetivo del método es reducir los obstáculos para la construcción de sitios *web* corporativos e incrementar la completitud, compatibilidad y calidad de los mismos.

El modelo W y sus metodologías asociadas para desarrollar sitios *web* corporativos (CWSD) integra características de cuatro modelos de procesos populares: el modelo en cascada (Presman, 2001), el modelo V (Germán Ministro de Defensa, 1992), el *Rapid Application Development* (RAD) (Maner, 1997) y el *Joint Application Development* (JAD) (IBM, 1977) Esto permite a los desarrolladores de sitios *web* corporativos ajustar ambos, el modelo y los pasos asociados de las metodologías para satisfacer sus necesidades específicas.

El modelo *W* es descrito como una variación del modelo *V*. Ambos modelos enfatizan en la correlación entre las fases de desarrollo (análisis de requerimientos, análisis, diseño del sistema y programación del código) y las fases de prueba (pruebas de aceptación, pruebas del sistema y pruebas de unidad).

El modelo *W* incluye un paso de prototipo, el cual es ejecutado inmediatamente después del diseño arquitectónico de todo el sistema, y la fase implementación es realizada como una serie construcciones individuales.

La metodología para el desarrollo de sitios *web* corporativos (CWSD) cubre las fases de planificación, análisis de requerimientos, diseño del sistema, implementación incremental, pruebas del sistema, pruebas de aceptación y mantenimiento.

En **la fase de planificación** se identifican los requerimientos preliminares del proyecto, indicando las necesidades de información externas e internas de la organización. Una vez identificados estos requerimientos se realiza el estudio de factibilidad y se conforma el equipo del proyecto.

Basados en la especificación de requerimientos preliminares, se **definen los requerimientos**, a través de una descripción de alto nivel, tomando en cuenta las necesidades de los clientes, así como también, la especificación de los requerimientos planteados por el grupo de desarrollo. En la misma, también, se plantea la estructura que tendrá el sitio *web*, en función a la navegación de cada una de sus páginas. Esta fase se realiza en conjunto con los clientes y usuarios de la aplicación.

Luego se **refinan los requerimientos técnicos**, en función de los requerimientos de información y la estructura diseñada. En esta fase se definen las bases de datos, componentes, aplicaciones y programas que deben conformar al sitio *web*. Asimismo, se elaboran los modelos de datos y de procesos. Finalmente se realiza una comparación entre los estándares que existen para desarrollar sitios *web*. Un ejemplo de ello es el *IEEE Standard 2001*, el cual plantea un conjunto de requerimientos generales y procedimientos a seguir por el grupo de desarrollo *web*.

La principal actividad de la **fase de diseño** la constituye el desarrollo del prototipo. Durante la misma, también se recomienda la participación activa de los

usuarios, ya que de esta manera se incrementa el entendimiento de las necesidades de información del cliente y facilita la comunicación entre éste y el grupo de desarrollo. Luego de que el prototipo es desarrollado y evaluado con el conjunto de requerimientos especificados, este constituye la base fundamental del diseño e implementación del sitio *web* corporativo.

La siguiente fase consiste en la **implementación incremental**. Dentro de la misma se ejecutan 4 actividades: Digitalización del material y diseño del arte, codificación del programa, pruebas de unidad y de integración y validación con el cliente. Seguidamente se ejecutan las **fases de prueba del sistema y pruebas de aceptación**; y finalmente, la **fase de mantenimiento**, la cual se ejecuta permanentemente, tomando en cuenta los nuevos requerimientos de la organización. La figura 33 ilustra el modelo de procesos asociado al método.

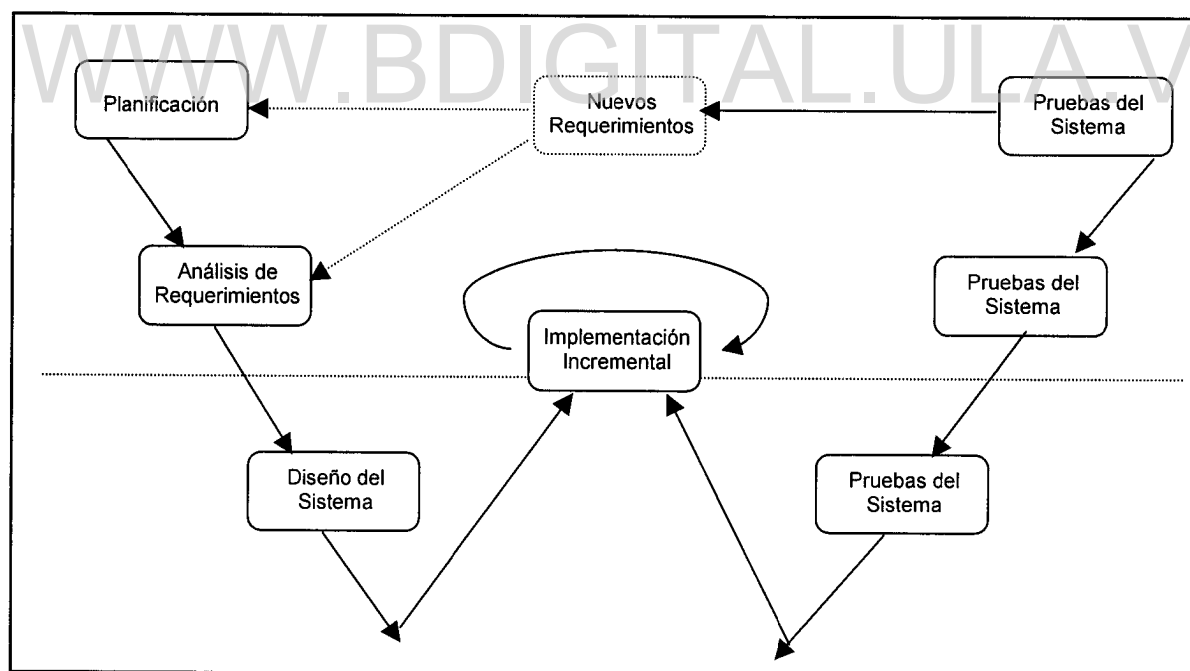


Figura 33. Modelo de Procesos asociado al Método para el Desarrollo de Sitios Web Corporativos. (Fuente: *The W Life Cycle Model and Associated Methodology for Corporate Web Site Development* (Sherrill y Chen, 2001))

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

1.5. MÉTODO PARA LA IMPLANTACIÓN DE GOBIERNO ELECTRÓNICO (e-government)

Este método es presentado por Horacio Capanegra (Capanegra, 2003). El método no pretende constituirse en una herramienta exhaustiva de gestión o de administración, sino por el contrario su alcance es orientador y general. Este método puede ser complementado y enriquecido con otras visiones, fundamentalmente en lo referido a la reingeniería de procesos y construcción y mantenimiento de sistemas de información.

El método cubre las fases: métricas para un gobierno digital, relevamiento y descripción de los procesos vigentes por servicio, diagnóstico y evaluación de los procesos relevados por servicio, definición de prioridades para la implantación de los servicios, identificación de los segmentos y sectores estratégicos por servicio, definición de la visión de un nuevo proceso por cada segmento o sector, formulación de una propuesta de rediseño por cada proceso, elaboración de nuevos diagramas de procesos por servicios a digitalizar, definición de formas de medición y construcción e implementación del sistema de información para el servicio electrónico y del proceso correspondiente (ver figura 34).

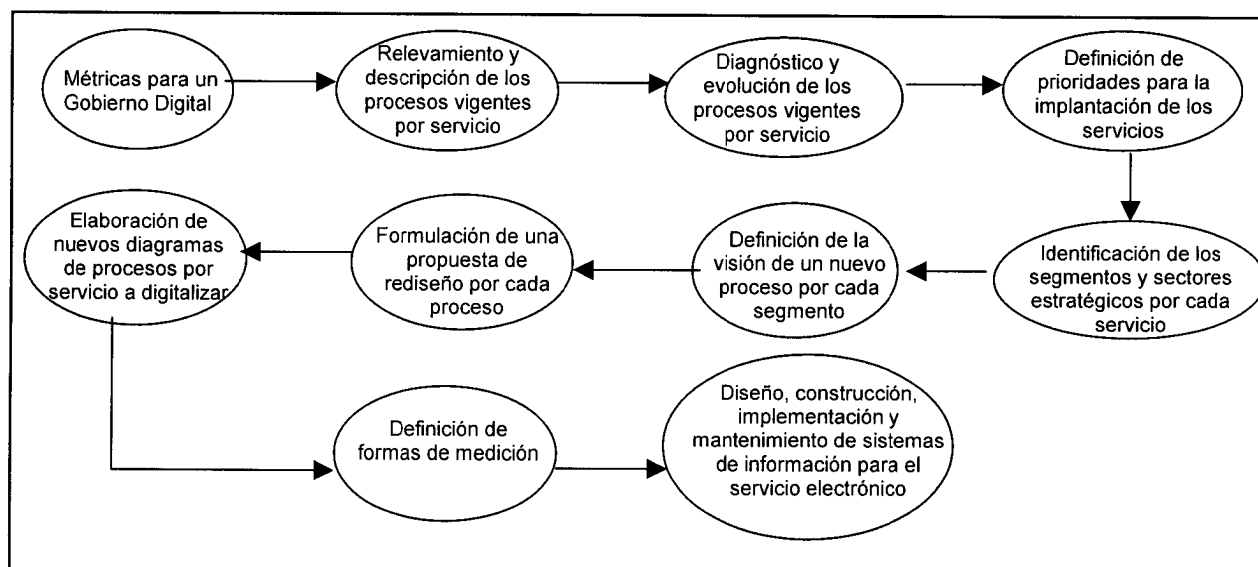


Figura 34. Modelo de Procesos asociado al Método para la Implantación de Gobierno Electrónico (e-Government). Fuente: Metodología para la Implantación del Gobierno Electrónico (Capanegra, 2003). Adaptación: (Mendoza, 2004).

La **fase de métricas para un gobierno digital** consiste en la construcción de indicadores en función de la demanda de los ciudadanos y de las empresas existentes en el país. Para ello se recomienda utilizar la metodología empleada por la Unión Europea para establecer indicadores cuantitativos y cualitativos de *benchmarking* (Grupo de trabajo de *e-Government*, 2001). Esta metodología presupone el consenso y la decisión de los servicios a digitalizar, hecho que fue discutido en el seno de la Comisión Europea. Para operar estos indicadores, el nivel de los servicios en línea debe ser medido.

En la **fase de relevamiento y descripción de los procesos vigentes por servicio** se esquematiza el proceso actual a través de un diagrama de procesos. El mismo debe describir en forma clara y concreta las entidades receptoras o transmisoras de información (personas, áreas, etc.), los soportes de información donde la misma queda registrada o es consultada (archivos, formularios, etc) y las actividades o tareas que son desarrolladas para realizar el proceso (de control, de decisión, de registro, etc.).

El **diagnostico y evaluación de los procesos relevados por servicio** implica la identificación de los problemas que se observan en los procesos actuales y análisis de sus causas, desde el punto de vista operativo, técnico, institucional, administrativo, económico, etc. Además, permitirá detectar las actividades dentro del proceso que agregan valor y las que no. Su estudio debe abordarse con equipos interdisciplinarios por la riqueza y perspectivas de personas con diferentes enfoques y perfiles.

En la fase **de definición de prioridades para la implantación de los servicios** se seleccionan aquellos servicios que ofrezcan los mayores beneficios políticos, financieros los que ofrezcan los mejores coeficientes de beneficio/costo y los que sean técnica y operativamente factibles de implementarse.

Luego, se **identifican los segmentos y sectores estratégicos por cada servicio**, es decir, se analiza todos los servicio o productos finales o productos intermedios asociados para alcanzarlo. Asimismo, por cada servicio se identifica la población objetivo que la demanda.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

La **creación de la visión de un nuevo proceso** se traduce en la necesidad de definir los objetivos y los atributos del mismo en relación a cada segmento o sector estratégico identificado por servicio a digitalizar. Después de definida la visión de cada proceso dentro del servicio, se **formula la propuesta de rediseño por cada proceso**. La misma consiste en definir el propósito de las innovaciones, las metas a alcanzar, su alcance y una descripción del proceso rediseñado. Esta propuesta debe estar acompañada de un plan de acción.

Seguidamente, se **elaboran los nuevos diagramas de procesos por servicio a digitalizar**. Estos deben diseñarse con mayor nivel de detalle. Cada uno de ellos debe reflejar claramente los puntos de intervención de entidades tanto receptoras como emisoras de información, los procesos a que son sometidos los datos y los soportes físicos donde se almacena la información. También, deben reflejar los cambios y ajustes realizados conforme a las recomendaciones que se efectuaron en la fase anterior, así como también las metas que deben alcanzarse en un determinado tiempo a partir de la puesta en vigencia del proceso rediseñado

La fase **de definición de formas de medición** se ejecuta con el objetivo de evaluar los resultados del proceso de rediseño o reingeniería. Para ello se elaboran indicadores de gestión que permitan analizar estos nuevos procedimientos administrativos.

Finalmente, a partir de los procesos rediseñados y del relevamiento de la información realizada se **construye un nuevo sistema de información** que satisfaga los requerimientos funcionales del servicio a digitalizar y del nuevo circuito administrativo propuesto. Para ello, el equipo de trabajo debe definir un estándar para diseñar, construir, implementar y mantener las aplicaciones informáticas.

Además, se debe contemplar la integración de la información corporativa del Estado ente todos sus organismos, para lo cual debe existir un marco de trabajo para establecer e implantar metadatos a través del sector público. Este marco lo constituye el estándar de metadatos para un gobierno digital. Este modelo garantiza la interoperabilidad, ya que facilita el intercambio de información entre Administraciones Públicas. Ejemplo de estos estándares es el e-GIF (*Government Interoperability Framework*) (*e-Government Interoperability Framework*, 2003).

2. EVALUACION DE LOS METODOS SELECCIONADOS

Generalmente, la evaluación de métodos va acompañada de marcos o cuadros de referencia que permiten calificar y posicionar un método con respecto a otro, considerando un conjunto de características o atributos que éstos deben cumplir: una estructura del proceso bien definida para facilitar el desarrollo de la aplicación *web*, un modelo claro y explícito que indique las características del producto que se desea desarrollar, la calidad de la aplicación *web* producida, que los recursos requeridos –humanos, económicos, tecnológicos, etc.– se utilicen de manera óptima en el contexto del desarrollo y una orientación contextual de manera tal que el método sea flexible y adaptable para ser utilizado fácilmente en dominios de aplicación diferentes.

Tal como se explicó en la evaluación de métodos realizada previamente en el capítulo 1, el marco de referencia diseñado para este fin está basado en los trabajos de (Jarve, 1992) y (Barrios, 2001). Este marco usa los principios de la Ingeniería de Métodos, los cuales establecen que un método debe tener un propósito específico *el dominio*, el tipo de problema a ser solucionado, y las condiciones o restricciones de *uso*. Otro principio establece que el método está conformado por un modelo de producto y un modelo de proceso.

El marco de referencia considera cuatro aspectos fundamentales para evaluar un método: la vista del dominio, la vista de uso, la vista del producto y la vista del proceso. Dentro de estas vistas se consideran a su vez las facetas y los atributos que las describen. Cada faceta representa un aspecto a evaluar dentro de una vista. La misma está compuesta por un conjunto de atributos y sus valores correspondientes, los cuales fueron adaptados con el fin de caracterizar los métodos para el desarrollo de aplicaciones *web*.

Se seleccionaron los cinco (5) métodos descritos en la sección anterior: **1)** Método para el Desarrollo de Portales (Oliveira, 2000), **2)** Marco de Referencia basado en componentes para *E-Commerce* (Peter Fingar, 2000) ACM, **3)** Método para el Desarrollo de Aplicaciones *e-Business* (Boban y Sertic, 2003), **4)** Modelo de Ciclo de Vida *W* y sus Metodologías Asociadas para el Desarrollo de Sitios Web Corporativos (Serle y Chen, 2001) y **5)** Método para la Implantación de Gobierno

Electrónico (Capanegra, 2003). A continuación se detallan los resultados de la evaluación:

2.1. LA VISTA DE DOMINIO

La vista del dominio está descrita a través de la faceta *alcance*. Esta faceta contempla todos los atributos relacionados con el área de aplicación, cubre factores técnicos, gerenciales o aquellos que están relacionados específicamente al dominio de aplicación.

La evaluación de los métodos, desde este punto de vista se puede observar en la tabla No. 6. En el atributo *área de aplicación* se puede observar que el método 4) fue diseñado para desarrollar sitios *web* corporativos, es decir, aplicaciones hipermedia, portales, *e-Commerce*, *e-Business* y sistemas de información *web* dependiendo siempre de la experiencia del grupo de desarrollo. El método 1) fue diseñado para portales y aplicaciones hipermedia. El resto de los métodos han sido diseñados para un dominio de aplicación en particular: *e-commerce* *e-Business* y *e-Government* respectivamente.

Todos los métodos cubren los aspectos relacionados con el dominio de aplicación para el cual se desarrolla la aplicación *web*. El método 1) y 4) cubren los aspectos técnicos, y gerenciales, lo cual permite guiar de una forma más clara al grupo de desarrollo. A diferencia del método 2) que únicamente cubre aspectos técnicos, lo que no garantiza que los recursos requeridos para el desarrollo de la aplicación *web* se administren adecuadamente. Asimismo, los métodos 3) y 5) sólo cubren aspectos gerenciales, lo que dificulta el desarrollo del proyecto, y la calidad del mismo.

Tabla No. 6. Vista del dominio (¿Para qué sirve? ¿para qué es concebido?)

Faceta	Atributos	Valores					
		1)	2)	3)	4)	5)	
Alcance	Área de aplicación	Hipermedia	✓				
		Educativo				✓	
		e-commerce					
		e-business		✓			
		Portales			✓	✓	
	Aspectos	Sistemas de Información web	✓			✓	
		e-government				✓	
		Técnicos				✓	
		Gerenciales	✓	✓		✓	✓
		Inherentes al área de Aplicación.	✓	✓	✓	✓	✓

Licencia Creative Commons

2.2. LA VISTA DE USO

Esta vista está descrita a través de las facetas de *aplicabilidad*, *uso* y *participación del usuario* en el desarrollo de la aplicación *web*. La faceta de *aplicabilidad* evalúa las fases que cubre 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. La Faceta *participación del usuario* especifica los tipos de usuario esperado y su grado de participación en el desarrollo de aplicaciones *web*. El resultado de evaluar estos aspectos se observa en la Tabla No. 7.

Tabla No. 7. Vista de Uso (Objetivos del que los usa)

Faceta	Atributos	Valores	1)	2)	3)	4)	5)
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 web	✓				
		Publicistas	✓				
		Personal directivo de grandes organizaciones			✓		✓
	Grado de Participación	Total	✓		✓	✓	✓
		Parcial		✓			
Ninguno							

Los cinco métodos cubren las fases de análisis y evaluación del ciclo de vida de una aplicación *web*. Sólo los métodos 1) y 4) cubren todas las fases del ciclo de vida (desde el análisis hasta el mantenimiento). Sin embargo, los métodos 3) y 5) no cubren las fases de diseño, desarrollo y mantenimiento, dejando la ejecución de las mismas a la experiencia del grupo de desarrollo.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Todos los métodos están orientados al servicio. Otro aspecto importante a considerar es la faceta de *uso*. Todos los métodos muestran qué hacer pero sólo el 5) enseña cómo se tiene que hacer. Esto está relacionado con la vista de proceso, en la que se observa que todo los métodos presentan una estructura de modelo de proceso bien clara y definida a través de un conjunto de fases, pasos y tareas, pero no indican en qué situaciones particulares deben ser ejecutadas, ni cómo deben ser aplicadas, lo cual depende de la experiencia del grupo de desarrollo que esté involucrado en cada proyecto particular. En cuanto a la *eficiencia de uso*, todos los métodos son adaptables y flexibles, sin embargo, los métodos 2) 3) y 5) no se consideran fáciles de usar porque dependen totalmente de la experiencia del grupo de desarrollo y no están bien documentados.

2.3. LA VISTA DEL PRODUCTO

Esta vista evalúa el modelo de producto usado por el método. Dos facetas son seleccionadas para describir el modelo del producto. *La descripción del producto*, que caracteriza al mismo por su notación, orientación y perspectiva; y la faceta de *descripción conceptual*, la cual describe los conceptos que son usados por el modelo. La evaluación de esta vista se puede observar en la tabla No. 8.

Tabla No. 8. Vista del Producto (Resultado de aplicar el método)

Faceta	Atributos	Valores	1)	2)	3)	4)	5)
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					
	Perspectiva del producto	Indefinido	✓	✓			
		Perspectiva funcional	✓		✓	✓	✓
		Perspectiva tecnológica		✓		✓	✓
Descripción Conceptual	Conceptos usados por el modelo	Perspectiva estructural	✓			✓	
		Conceptos Técnicos	✓	✓		✓	
		Conceptos estéticos	✓				

Es indispensable que un método disponga de modelo de producto con el fin de garantizar el entendimiento del resultado que se obtendrá por el uso del método. Sin

embargo, ninguno de los métodos evaluados (a excepción del 2)) presentan explícitamente el producto. Además, utilizan la notación informal (textual) para describirlo lo cual dificulta la caracterización de cada uno de los productos que se generarán al utilizar dicho método.

2.4. LA VISTA DEL PROCESO

Esta contiene un conjunto de facetas y atributos que describen el proceso de desarrollo de la aplicación que se desea producir. Esta vista evalúa el modelo de proceso usado por el método, a través de cinco facetas que consideran los atributos *orientación* y *enfoque*, las *características* y los *procesos cubiertos por el modelo*. La tabla No. 9 muestra los resultados de esta evaluación.

Todos los métodos evaluados siguen un enfoque evolutivo (adaptativo), lo que se considera una gran fortaleza, debido a que permiten hacer cambios a medida que se va evaluando la aplicación con el usuario y/o cliente. Todos los métodos presentan una estructura del proceso bien definida, sin embargo, el 2) no divide sus fases en pasos y tareas, lo que hace que las actividades no se indiquen de manera clara y explícita al grupo de desarrollo.

El método 2) no considera la gerencia del proyecto, lo que quiere decir que no incluye dentro del modelo de proceso las actividades de planificación y control, organización del equipo de trabajo y documentación.

En cuanto a la faceta de *cobertura del ciclo de desarrollo* los métodos 3) y 5) sólo cubren, la fase de análisis de requerimientos, pero no contemplan el resto de las fases de desarrollo, ya que el objetivo fundamental de los mismos es el de adaptar las aplicaciones *web (e-business, e-government)* al dominio de aplicación.

El método 2) cubre la etapa de análisis, definición de requerimientos; dándole especial importancia al diseño de las arquitecturas y a la comunicación de los objetos distribuidos. Además, cubre la etapa de pruebas de cada uno de los componentes, las de integración, las cuales son las de mayor atención, y las de la aplicación.

Por último, se puede observar que los métodos 1) y 4) son los que cubren la mayor cantidad de fases del ciclo de desarrollo. Asimismo, ambos cubren la etapa de mantenimiento del proceso de post-desarrollo.

Tabla No. 9. Vista del Proceso (Desarrollo del producto) (a)

Faceta	Atributos	Valores	Valores				
			1)	2)	3)	4)	5)
Modelo de proceso de desarrollo	Orientación	Orientada al producto		✓			
		Orientada a la actividad	✓				
		Orientada a la decisión			✓	✓	✓
	Enfoque	Ad-hoc					
		Predictivo (cascada)					
Características del proceso	Claridad	Evolutivo (adaptativo)	✓	✓	✓	✓	✓
		Estructura del proceso bien definido	✓	✓	✓	✓	✓
		Fases/pasos/tareas	✓		✓	✓	✓
	Estructura del proceso (qué hacer)	Iterativo		✓			
		Secuencial estricto	✓		✓	✓	✓
		Incremental		✓			
	Modo de ejecución del proceso	Definición clara del flujo de trabajo	✓		✓	✓	✓
		Progreso visible externamente	✓	✓			
	Visibilidad	Independiente de las herramientas	✓	✓	✓	✓	✓
		Modularidad	✓	✓		✓	
	Dependencia tecnológica	Procesos modificables			✓		✓
		explicita					
	Adaptabilidad	Completa	✓				
Implicita			✓				
Guía	Definición del plan del proyecto	✓		✓	✓	✓	
	Estimación de recursos	✓		✓	✓	✓	
Gerencia de proyecto	Planificación y control	Monitoreo y evaluación del progreso	✓		✓	✓	✓
		Estructura del equipo					
		Individual					
	Equipo de trabajo	Equipos pequeños (<= 5 personas)					
		Equipos Grandes (> 5 personas)					
		Calidad en la planificación y control					
	Aseguramiento de calidad	Calidad del equipo					
		Revisión de técnicas de validación y verificación	✓	✓	✓	✓	
	Gestión de la configuración	Planificación y ejecución de la gestión de configuración del software					
		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	✓			✓	
		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).							
Comunicación de objetos distribuidos.			✓				

Licencia Creative Commons:

Tabla No. 9. Vista del Proceso (Desarrollo del producto) (b)

Faceta	Atributos	Valores					
		1)	2)	3)	4)	5)	
Cobertura del ciclo de desarrollo	Pruebas	Prueba de cada uno de los componentes	✓	✓		✓	
		Pruebas de integración	✓	✓		✓	
		Prueba de la aplicación web.	✓	✓		✓	
		Validación y verificación de la aplicación web.	✓			✓	
Proceso de post-desarrollo	Instalación	Implantación de la aplicación web	✓			✓	
		Mantenimiento				✓	
		Soporte de mantenimiento	✓				✓

3. ASPECTOS COMUNES Y DIFERENTES ENTRE LOS METODOS PARA DESARROLLAR LAS APLICACIONES WEB SELECCIONADAS.

Una vez descritos los métodos para el desarrollo de las aplicaciones *web* seleccionadas (portales, *e-commerce*, *e-business*, sistemas de información *web* y *e-government*) se comparan con el fin de extraer los aspectos comunes entre ellos, los cuales se mencionan a continuación:

- Todos los métodos contemplan la fase de análisis de requerimientos (con diferentes denominaciones). En la misma se definen y especifican los requerimientos del cliente, usuario, grupo de desarrollo y del negocio.
- Todos los métodos (a excepción del método para desarrollar portales) contemplan un estudio detallado de la organización donde se desarrollará la aplicación *web*.
- El método para el desarrollo de *e-commerce* no contempla la construcción de un prototipo, que luego sea validado con el cliente, para mejorar la comunicación entre éste y el grupo de desarrollo y corregir posibles fallas. Asimismo, el método para implantar *e-government* tampoco lo contempla de manera explícita. El resto de los métodos si cubren esta actividad.
- Los métodos para el desarrollo de *e-business* y para la implantación de *e-government* no contemplan explícitamente actividades de integración. El resto de los métodos descritos plantean actividades para integrar los componentes de la aplicación *web*.

Licencia Creative Commons:

- Todos los métodos plantean actividades orientadas a cumplir con los atributos de calidad del producto desarrollado. El método para el desarrollo de sitios *web* corporativos lo hace a través de las actividades de pruebas. El resto de los métodos plantean la ejecución de actividades de validación y verificación, al finalizar las fases de requerimientos, diseño e implementación.
- Los métodos para el desarrollo de *e-commerce* y *e-business* no presentan actividades orientadas al mantenimiento de la aplicación *web*. El resto de los métodos si las describen como la última etapa a ejecutar y de manera permanente.
- El método para desarrollo de *e-commerce* contempla la Reutilización de componentes y de modelos de objetos de negocio y casos de uso en las fases de diseño e implementación.
- Los métodos para el desarrollo de *e-business* y de *e-government* están orientados a la adaptación de las aplicaciones *web* dentro de la organización. Los mismos dejan de lado los detalles técnicos relacionados con el desarrollo del producto, ya que dejan al grupo de desarrollo la responsabilidad de seleccionar el modelo de proceso más adecuado para desarrollar dicho producto.
- Todos los métodos son un conjunto de pasos secuenciales que indican explícitamente las actividades a ejecutar, y algunas veces de manera implícita cómo ejecutarlas, más no indican en que situaciones particulares se debe hacer.
- Ninguno de los modelos de procesos asociados a los métodos relacionan las actividades a ejecutar en el ciclo de desarrollo con los modelos de producto asociados a cada una de las aplicaciones *web*.

4. FASES COMUNES DE DESARROLLO DE UNA APLICACIÓN WEB.

Tomando en cuenta los aspectos descritos en la sección anterior, la figura 35 representa el modelo de procesos asociado a los elementos comunes encontrados en los métodos para el desarrollo de las aplicaciones *web* seleccionadas.

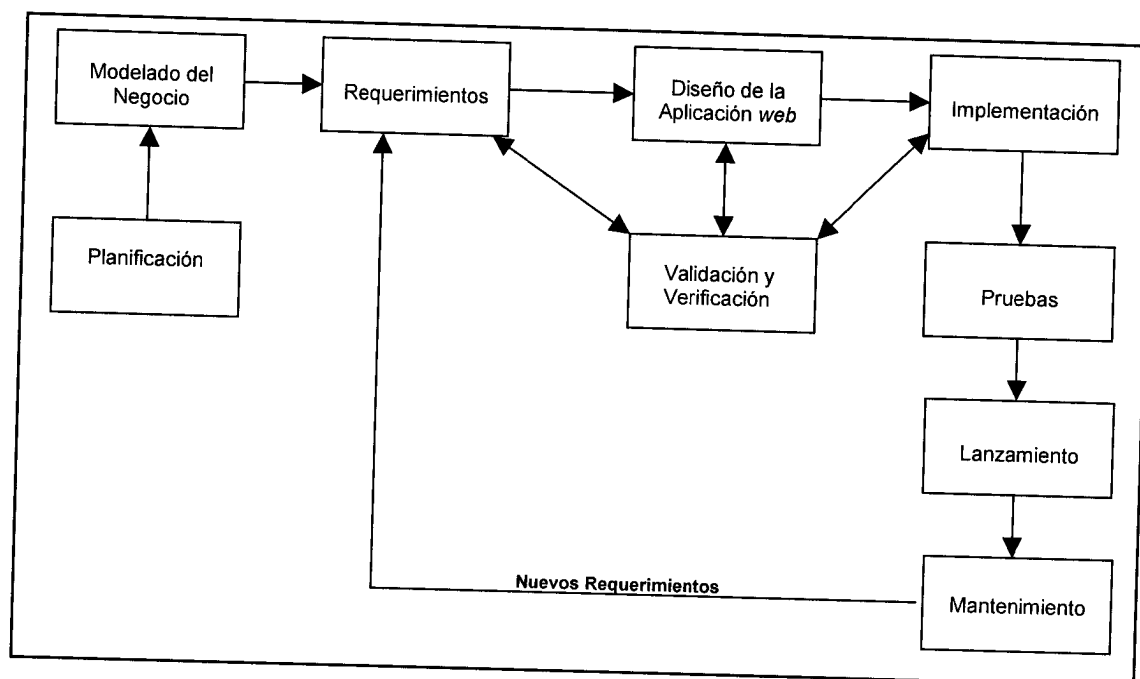


Figura 35. Fases comunes encontradas en los métodos para el desarrollo de las aplicaciones web seleccionadas.

La fase de planificación es contemplada por la mayoría de los métodos, a excepción del método para desarrollo de aplicaciones *e-commerce*. Esta fase consiste en la formulación del plan del proyecto, el cual incluye plan de contenido, organización del equipo de trabajo, definición de requerimientos preliminares del proyecto, estudio de factibilidad y la definición de métricas de calidad, del recurso humano, proceso y del producto.

La fase modelado del negocio es cubierta por la mayoría de los métodos, a excepción del método para el desarrollo de sitios web corporativos. Esta fase incluye las estrategias de mercado, perfiles del consumidor, estrategia funcional, análisis de fortalezas, oportunidades, debilidades y amenazas de las organizaciones, análisis del ambiente del negocio y relevamiento y descripción de los procesos que se llevan a cabo en la organización.

La fase de requerimientos es cubierta por todos los métodos. La misma incluye las características del producto a ser desarrollado, la definición y

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

especificación de los requerimientos que debe satisfacer la aplicación *web*, y las restricciones que la misma debe tener.

La fase de diseño no es cubierta por los métodos para el desarrollo de *e-Business* y *e-Government*. La misma incluye el diseño de la arquitectura, de la Interfaz Gráfica de Usuario (GUI), la construcción de un prototipo y el diseño de las bases de datos. En el caso particular del método para desarrollar aplicaciones *e-Commerce* contempla el reuso de los modelos de negocio, objetos y casos de uso.

La fase de implementación no es cubierta por los métodos para el desarrollo de *e-Business* y *e-Government*. La misma contempla la construcción del producto, la digitalización del material y diseño del arte y la codificación de programas. En el caso del método para el desarrollo de *e-Commerce* contempla la Reutilización de componentes y la implementación de los modelos de negocio en la infraestructura tecnológica.

La fase de pruebas es cubierta únicamente por el método para el desarrollo de sitios *web* corporativos. La misma contempla las pruebas del sistema y las pruebas de aceptación.

La fase de lanzamiento es cubierta, explícitamente, por el método para el desarrollo de portales e incluida implícitamente en los métodos para el desarrollo de sitios *web* corporativos y *e-commerce* dentro de las actividades de la fase de implementación. La misma incluye la puesta en marcha, publicación y hospedaje de la aplicación *web* en los servidores respectivos.

La fase de mantenimiento es cubierta por los métodos para el desarrollo de portales y desarrollo de sitios *web* corporativos. La misma incluye el mantenimiento perfectivo, adaptativo, correctivo o preventivo de la aplicación *web* y la construcción de nuevos productos y servicios en el caso de que surjan nuevos requerimientos.

La fase de validación y verificación es contemplada por todos los métodos. La misma se ejecuta luego de culminada una fase (especialmente las de requerimientos, diseño e implementación) con el fin de corregir los posibles errores que se pudiesen presentar y regresar a la fase donde el mismo ocurrió.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

5. CONCLUSION.

El modelo de procesos asociado a un método define las actividades que deben ser ejecutadas para desarrollar una solución de *software*. Por lo tanto, los métodos deben conectar a los actores con las actividades y productos que se producen parcial o totalmente como parte de la solución.

En este capítulo se describieron los modelos de procesos asociados a los métodos para el desarrollo de aplicaciones *web* (portales, *e-commerce*, *e-business*, sistemas de información *web* y *e-government*), se analizaron sus aspectos comunes, con el fin de definir un modelo genérico que contenga las fases comunes de todos los modelos de procesos descritos.

Dado que ninguno de los modelos de procesos asociados a los métodos relacionan las actividades a ejecutar en el ciclo de desarrollo con los modelos de producto asociados a cada una de las aplicaciones *web* y, además, todos los métodos son un conjunto de pasos secuenciales que indican explícitamente las actividades a ejecutar, y algunas veces de manera implícita cómo ejecutarlas, sin indicar en que situaciones particulares se debe hacer; se considera necesario proponer un nuevo método situacional para el desarrollo de aplicaciones *web*.

El nuevo método debe asociar las actividades con los productos, además de permitir tomar decisiones según la situación particular enfrentada en cada proyecto; y garantizar una guía completa al equipo de desarrollo, tomando en cuenta las necesidades particulares del dominio de aplicación seleccionado (comercio electrónico, gobierno electrónico, etc).

El modelo genérico de fases de desarrollo definido en este capítulo, será utilizado como punto de partida para definir las intenciones globales en el método situacional para el desarrollo de aplicaciones *web* presentado en el próximo capítulo.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

CAPITULO 5

METODO SITUACIONAL PARA EL DESARROLLO DE APLICACIONES *WEB*: MODELO DE PROCESO

En el presente capítulo se presenta el modelo de proceso de nuestro método situacional para desarrollar aplicaciones *web*. El mismo está asociado a un modelo de producto el cual fue objeto del capítulo 3.

Asimismo, se presenta un estudio de un caso que permite demostrar la aplicabilidad del método descrito.

1. PROCESO DE CONSTRUCCION DEL METODO SITUACIONAL.

El método situacional, ofrece la posibilidad, al grupo de desarrollo, de elegir entre múltiples alternativas cuál es la más conveniente, en función del dominio de aplicación y de la situación particular de modelado enfrentada por tal grupo de desarrollo como por ejemplo, los elementos del contexto de desarrollo, el tipo de aplicación, los usuarios, las herramientas y tecnología, la experiencia de desarrollo, etc.

Según el enfoque situacional de la Ingeniería de Métodos, las primeras actividades a realizar para construir un método consiste en caracterizar y describir el ambiente de utilización del método. Para el caso de la presente investigación, esa actividad fue ejecutada en el capítulo 3 a través de la descripción de los dominios de aplicación de las aplicaciones *web* seleccionadas (portales, *e-commerce*, *e-business*, *SIW*, *e-Government*).

Seguidamente, en el capítulo 4 se explicaron los modelos de procesos asociados a diferentes métodos para desarrollar este tipo de aplicaciones, extrayéndose de ellos, las fases comunes con el fin de generar un modelo de procesos estándar.

Partiendo de este modelo de procesos, del modelo de producto de una aplicación *web* presentado en el capítulo 3 y de cada uno de los dominios de aplicación, a continuación, se presenta el modelo de procesos para el desarrollo de

Licencia Creative Commons:

una aplicación *web*. El mismo está expresado en función de mapas de proceso y directivas de recorrido de estos mapas.

Este método es construido a través del uso de las tres (3) técnicas para la construcción de métodos situacionales. La primera es la Ingeniería de Métodos basada en el ensamblaje, donde se describen los fragmentos de métodos para ser ensamblados conformando el modelo de procesos, generado por el uso de la estrategia de abstracción de cada uno de los modelos de procesos asociados a cada uno de los métodos para desarrollar las aplicaciones *web* seleccionadas.

La segunda técnica es la de la Ingeniería de Métodos basada en la extensión, la cual permitió utilizar y adaptar patrones que ayudaron a identificar situaciones típicas y a dar orientación para ejecutar la extensión de método requerida.

La tercera técnica utilizada es la de la Ingeniería de Métodos basada en el paradigma, ya que los modelos de proceso generados se obtuvieron de la abstracción realizada a los modelos de proceso asociados con el desarrollo de cada una de las aplicaciones *web* seleccionadas.

El modelo de proceso asociado al método es representado a través de mapas, en los cuales existen intenciones, que por su complejidad, no pueden ser representadas en un solo mapa. Es por ello, que nosotros utilizamos dos (2) tipos de mapas: el mapa global y uno o más mapas locales. Cada mapa local corresponde al detalle de una intención global. En nuestra propuesta tenemos un (1) mapa global, tres (3) mapas locales de alto nivel y cuatro (4) mapas locales de bajo nivel.

2. MODELADO DE NEGOCIO

Desarrollar una aplicación *web* consiste en generar un producto que formará parte del modelo de sistemas de información de una organización (de su modelo de negocios).

En la figura 36 se presenta la intención de “*modelar el negocio*”. Modelar el negocio es la base para entender un negocio y obtener requerimientos de información de la organización. De hecho, tal como lo señala (Barrios y Montilva, 2003), el modelado de negocios es la actividad central para diferentes áreas incluyendo, reingeniería de procesos, modelado empresarial, desarrollo de

Sistemas de Información Web (SIW), comercio electrónico, Integración de Procesos y Gestión de Procesos de Negocios.

El punto de partida está dado por la intención de “*modelar el negocio*”, la cual no se ha incluido como parte del método por considerarse en un nivel superior al proceso de desarrollo de aplicaciones *web*. Modelar el negocio implica construir el modelo de objetivos del negocio, el modelo de procesos del negocio y el modelo de sistemas de información que soportará los dos anteriores.

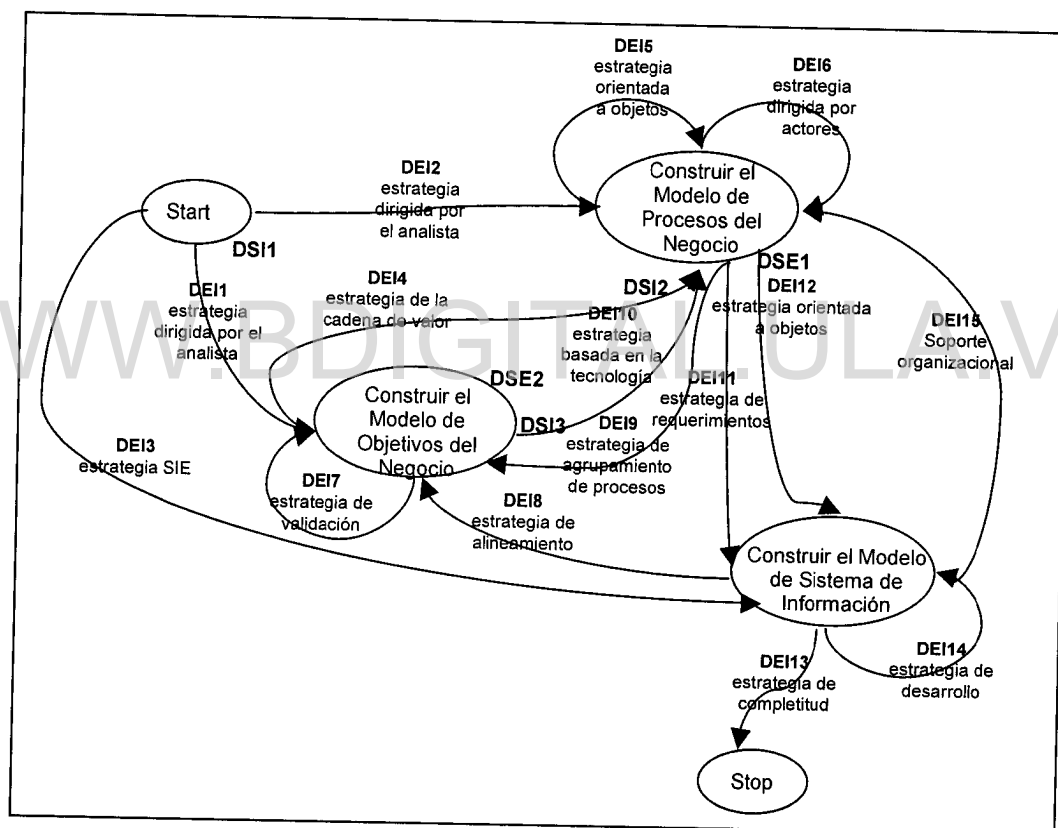


Figura 36. Mapa de la Intención de *Modelar el Negocio*. (Adaptación de: Business Modelling Through Roadmaps (Barrios y Montilva, 2003))

El mapa que representa la intención de modelar el negocio contiene los siguientes elementos:

- **5 Intenciones**, las cuales representan el objetivo que se desea satisfacer en el proceso de modelar el negocio. Estas son: comienzo, construir el modelo de objetivos del negocio, construir el modelo de

procesos del negocio, construir el modelo de sistema de información y parada

- **Un conjunto de directivas** (ver tabla 10) que permiten decidir qué modelar (si los procesos, los objetivos o los sistemas de información). Cada una de las intenciones están soportadas por diferentes tipos de directivas: Directivas de Ejecución de Intención (DEI), Directivas de Selección de Estrategia (DSE) y Directivas de Selección de Intención (DSI), explicadas en el capítulo 2 sección 2.

Tabla No. 10. Directivas de Ejecución de Intención (DEI) de la Intención de *Modelar el Negocio* (a)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumentos
DEI1. Estrategia dirigida por el analista.	Definición de la visión, misión y objetivos generales del negocio.	Falta de documentación de la organización.	(A1): No existe documentación de los procesos de la organización y la organización posee un bajo grado de madurez y/o los miembros de la organización conocen la importancia de modelar el negocio.
DEI2. Estrategia dirigida por el analista.	Definición y descomposición jerárquica de los procesos primarios del negocio.	El analista tiene necesidad de modelar el negocio.	(A2): No existe información de los procesos de la organización y la organización posee un bajo nivel de madurez y/o los miembros de la organización conocen la necesidad de modelar el negocio.
DEI3. Estrategia de SIE.	Determinación del dominio de la aplicación del SIE.	Requerimientos del negocio basados en tecnología.	(A3): Los miembros del grupo de desarrollo tienen experiencia en modelado de aplicaciones web y/o existe una documentación completa acerca de los procesos del negocio y/o la organización tiene un buen nivel de madurez.
DEI4. Estrategia de la cadena de valor.	Descomposición del alto nivel primario y que soporte los procesos del negocio en una jerarquía de procesos	Modelo de objetivos del negocio	(A4): Los objetivos del negocio son conocidos y los miembros de la organización están identificados con los procesos del negocio y existe documentación acerca de los procesos del negocio y la organización tiene un grado medio de madurez.
DEI5. Estrategia orientada a objetos.	Construcción del modelo de procesos del negocio orientado a objetos.	Modelo de procesos del negocio.	(A5): Existe documentación de los procesos del negocio y/o los miembros de la organización cuentan con la experiencia para modelar los objetos del negocio.
DEI6. Estrategia dirigida por actores.	Identificación y clasificación de los actores y las unidades del negocio.	Procesos del negocio	(A6): Los miembros de la organización conocen los procesos del negocio y hay una documentación de la estructura organizacional y/o la organización tiene cierto grado de madurez y los miembros de la organización conocen los perfiles de los actores.
DEI7. Estrategia de validación.	Revisión de la visión, misión y objetivos de la organización en conjunto con los gerentes del negocio.	Modelo de objetivos del negocio.	(A7): Están definidos los objetivos del negocio y/o los objetivos del negocio no han sido validados por los miembros de la organización y la organización tiene un bajo grado de madurez.
DEI8. Estrategia del alineamiento.	Relación de los objetivos del negocio con el SIE.	Modelo de SIE y objetivos del negocio.	(A8): La organización tiene grado medio de madurez y los miembros de la organización conocen la importancia de implantar la aplicación web y los miembros de la organización conocen el impacto que la aplicación web causa en los objetivos del negocio.

Licencia Creative Commons

Tabla No. 10. Directivas de Ejecución de Intención (DEI) de la Intención de *Modelar el Negocio* (b)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumentos
DEI9. Estrategia del agrupamiento de procesos.	Agrupamiento de los procesos del negocio que se encuentran jerárquicamente separados, con el fin de conformar los procesos primarios del negocio.	Modelo de procesos del negocio.	(A9): Se dispone del modelo de procesos del negocio y los procesos se encuentran separados jerárquicamente y/o se requiere realizar el modelo de objetivos del negocio.
DEI10. Estrategia basada en la tecnología.	Identificación de las tecnologías y modelado de reglas del negocio de alto nivel	Modelos de objetivos del negocio, procesos del negocio.	(A10): Los miembros de la organización entienden los procesos del negocio y/o hay una documentación completa acerca de la tecnología y reglas del negocio usadas para soportar los procesos del negocio y/o la organización tiene un bajo nivel de madurez y los miembros de la organización entienden la tecnología usada dentro de la organización.
DEI11. Estrategia de requerimientos.	Identificación de los requerimientos generales del sistema de información que soportará los procesos del negocio.	Modelo de procesos del negocio.	A(11): Se dispone del modelo de procesos del negocio y/o se desea conocer los requerimientos de los líderes de la organización, la organización tiene cierto grado de madurez.
DEI12. Estrategia orientada a objetos.	Creación del diagrama de clases de los objetos del negocio.	Modelo de procesos del negocio, requerimientos generales basados en tecnología.	(A12): Existe documentación de los procesos del negocio y los miembros del grupo de desarrollo tienen experiencia en el proceso de modelado orientado a objetos y/o se dispone de herramientas que faciliten el proceso de modelado.
DEI13. Estrategia de Completitud	Culminación de la intención de modelar el negocio.	Modelo de negocio.	(A13): Los miembros de la organización han concluido el modelo de negocios y el modelo de negocios ha sido validado por los gerentes de la organización y el modelo de negocios se encuentra libre de errores.
DEI14. Estrategia del desarrollo.	Desarrollo de cada una de las aplicaciones que forman parte del modelo de sistema de información	Modelo del sistema de información y modelo de procesos del negocio.	(A14): Se dispone del modelo de sistema de información, y/o se definieron claramente las aplicaciones requeridas para soportar los procesos del negocio, y/o se cuenta con los recursos necesarios para desarrollar las aplicaciones que soportarán el Sistema de Información Empresarial (SIE)
DEI15. Estrategia del soporte organizacional.	Alineación del sistema de información con los procesos del negocio.	Modelo de sistema de información	(A15): Se dispone del modelo de sistema de información y/o se requiere alinear el modelo de sistema de información con cada proceso del negocio a ser soportado y/o la organización tiene cierto grado de madurez.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

3. MODELO DE PROCESO PARA EL DESARROLLO DE APLICACIONES WEB

A continuación, se presenta el modelo de procesos correspondiente al método para desarrollar aplicaciones *web*, el cual está expresado en términos de intenciones, directivas de selección de estrategia (DSE), directivas de selección de intención (DSI) y directivas de ejecución de intención (DEI). Las DSI permiten progresar en el proceso de construir una aplicación *web* o parte de ella, al seleccionar la próxima intención a ejecutar. Las DSE permiten seleccionar una estrategia para ejecutar la intención seleccionada. Las DEI describen el conjunto de actividades y tareas a ser ejecutadas para construir un producto de acuerdo a la estrategia seleccionada.

3.1. NIVELES DE ABSTRACCION DE LOS MAPAS INTENCIONALES

En los mapas de procesos existen intenciones, que por su complejidad, no pueden ser representadas en un solo mapa. Es por ello, que nosotros utilizamos dos (2) tipos de mapas: el mapa global y uno o más mapas locales. Cada mapa local corresponde al detalle de una intención global.

En algunos casos la intención de un mapa local pudiera requerir de otros mapas "locales" más detallados.

En nuestra propuesta tenemos un (1) mapa global, tres (3) mapas locales de alto nivel y cuatro (4) mapas locales de bajo nivel.

Las Directivas de Selección de Estrategia (DSE) de cada mapa constituyen la sumatoria de los argumentos de las Directivas de Ejecución de Intención (DEI) de los mismos.

Asimismo, estas directivas (DEI) no se detallan aquí, y deberán expresarse como contextos adecuados al formalismo del proyecto *NATURE* (Barrios, 2002). Estos contextos son: plan, selección y ejecutable. La definición de los mismos queda pendiente como trabajo futuro.

A continuación se presentan los mapas globales, locales de alto nivel y locales de bajo nivel.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

3.2. MAPA GLOBAL

Es el mapa más general del modelo de proceso para desarrollar aplicaciones web. El mismo contiene tres (3) intenciones fundamentales: “gestionar el desarrollo de la aplicación web”, “desarrollar la aplicación web” e “implantar y mantener la aplicación web” (ver figura 37).

La intención de “gestionar el desarrollo de la aplicación web”, tiene como objetivo principal la definición de los procesos con los cuales se establece el proyecto, como es coordinado y el cómo hacer uso efectivo de los recursos (tiempo, personas, dinero) para lograr los objetivos del proyecto. Los principales productos generados a este nivel son: el plan del proyecto, el modelo de calidad y la lista de riesgos y los planes para mitigarlos.

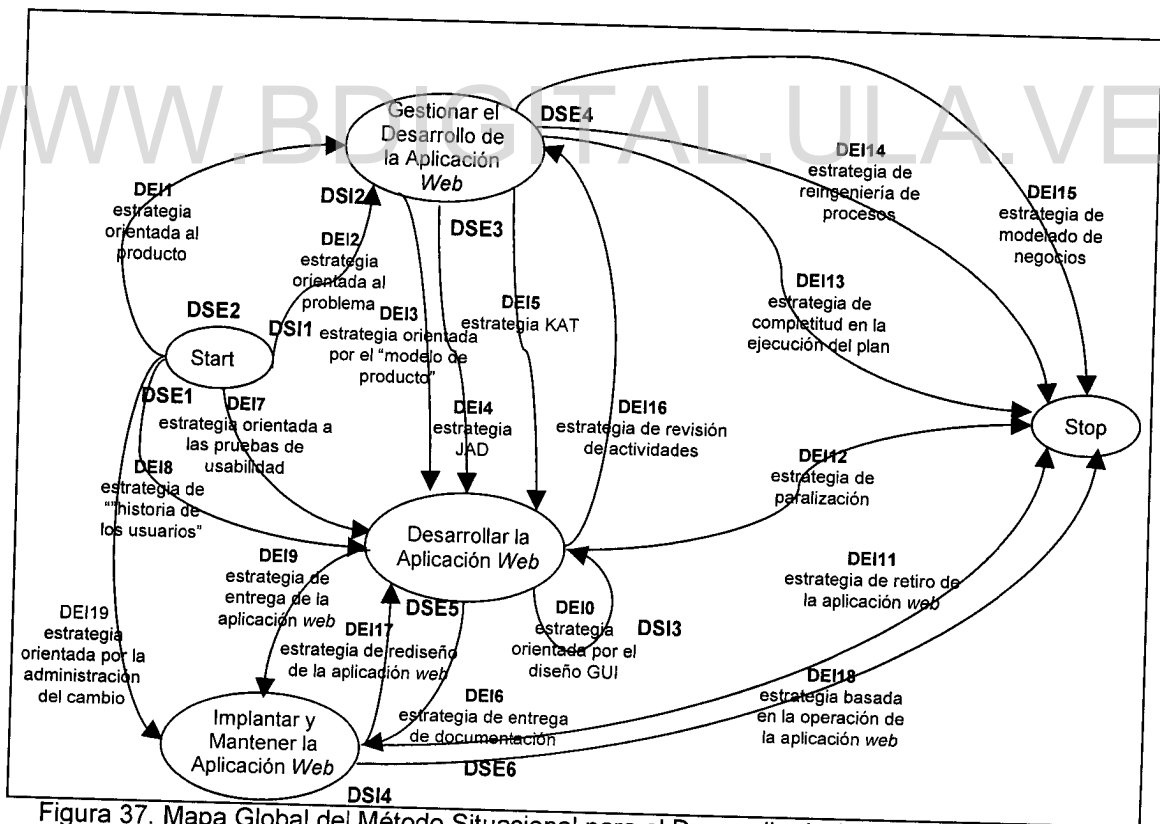


Figura 37. Mapa Global del Método Situacional para el Desarrollo de Aplicaciones Web

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

La intención de “*desarrollar la aplicación web*” consiste está orientada a la construcción de la aplicación *web*. El producto que se obtiene luego de su ejecución es la aplicación *web* operativa y probada.

La intención de “*implantar y mantener la aplicación web*” permite realizar la instalación, mantenimiento, operación y soporte y adiestramiento de los usuarios y operadores que manipularán la misma. El producto que genera es la aplicación *web* instalada y operando, así como los operadores y usuarios adiestrados y la documentación del mantenimiento.

Este mapa está conformado por los siguientes elementos:

- **5 Intenciones**, las cuales representan el objetivo que se desea satisfacer en el proceso. Estas son: *Comienzo, gestionar el desarrollo de la aplicación web, desarrollar la aplicación web, implantar y mantener la aplicación web y Parada*
- **6 Directivas de Selección de Estrategia (DSE)**, las cuales permiten seleccionar una estrategia para ejecutar una intención de modelado (final), a partir de una intención de partida. Estas son:
 - a) **DSE1:** De la cual se puede seleccionar la estrategia de “*historia de los usuarios*” o la estrategia “orientada a las pruebas de usabilidad”, con el fin de ejecutar la intención de “*desarrollar la aplicación web*”.
 - b) **DSE2:** A partir de esta directiva se puede seleccionar la estrategia “orientada al producto” o la estrategia “orientada al problema”, con el fin de ejecutar la intención de “*gestionar el desarrollo de la aplicación web*”.
 - c) **DSE3:** A partir de esta directiva se puede seleccionar la estrategia “orientada al modelo de producto”, la estrategia “JAD” o la estrategia “KAT” o una combinación de “JAD” con “modelo de producto” o “KAT” con “modelo de producto”, con el fin de partir de la intención de “*gestionar el desarrollo de la aplicación web*” a la intención de “*desarrollar la aplicación web*”.
 - d) **DSE4:** A partir de la cual se pueden seleccionar la estrategia de “modelado de negocios”, la estrategia de “reingeniería de procesos”

o la estrategia de “completitud en la ejecución del plan”, con el fin de partir e la intención de *“gestionar el desarrollo de la aplicación web”* a la intención de *“parada”*.

e) **DSE5:** A partir de esta directiva se puede seleccionar la estrategia de “entrega de la aplicación web”, la estrategia de “entrega de documentación” o ambas, con el fin de partir de la intención *“desarrollar la aplicación web”* a la intención de *“implantar y mantener la aplicación web”*.

f) **DSE6:** A partir de esta directiva se puede seleccionar la estrategia de “retiro de la aplicación web” o la estrategia “basada en la operación de la aplicación web”, con el fin de partir de la intención de *“implantar y mantener la aplicación web”* a la intención de *“parada”*.

➤ **4 Directivas de Selección de Intención (DSI)**, las cuales permiten seleccionar una intención destino entre las disponibles a partir de una intención de partida. Las mismas están identificadas como DSI1, DSI2, DSI3 y DSI4.

➤ **19 Directivas de Ejecución de Intención (DEI)**, las cuales guían la ejecución de una intención según una estrategia específica. Estas se listan en la tabla No.11 donde se indica, su descripción, situación inicial y argumentos para la ejecución de las mismas.

Tabla No. 11. Directivas de Ejecución de Intención (DEI) del Mapa Global del Método Situacional para el Desarrollo de Aplicaciones Web (a)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI1. Estrategia orientada al producto	Elaboración del plan de calidad con el fin de medir los resultados de las metas de calidad que se desean alcanzar	Requerimientos generales de la aplicación web, metas de calidad.	(A1): Existe un modelo de la aplicación web que soportará la organización y/o la organización tiene cierto grado de madurez, y/o se desea generar un producto de calidad y/o los miembros del grupo de desarrollo tienen experiencia en esta área.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Tabla No. 11. Directivas de Ejecución de Intención (DEI) del Mapa Global del Método Situacional para el Desarrollo de Aplicaciones Web (b)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI2. Estrategia orientada al problema	Establecimiento de los objetivos, el propósito del proyecto y sus requerimientos más generales.	Necesidad de desarrollo de aplicación web.	(A2): Se tiene una idea general de la aplicación web a desarrollar y/o la organización tiene cierto grado de madurez y/o existe en el grupo de desarrollo gente con experiencia en gerencia de proyectos.
DEI3. Estrategia orientada por el "modelo de producto"	Especificación de los componentes que interactúan en cada una de las capas del modelo de producto de la aplicación web y especificación de la interrelación de los mismos.	Modelo de producto de la aplicación web y requerimientos generales de la aplicación web.	(A3): Se cuenta con el modelo de producto de la aplicación web y/o no se requiere reutilizar patrones de diseño de arquitectura lógica y/o el equipo de desarrollo conoce al detalle cada uno de los componentes que integran la aplicación web.
DEI4. Estrategia JAD.	Definición de los objetivos y límites de la aplicación web,	Modelo del Negocio, usuarios y/o clientes forman parte del equipo de trabajo.	(A4): Se cuenta con la participación activa de usuarios, clientes, desarrolladores y stakeholders y/o la organización tiene cierto grado de madurez y/o el grupo de trabajo conoce los procesos del negocio.
DEI5. Estrategia KAT.	Recopilación de datos de una gran variedad de fuentes, e identificación los objetos y acciones del dominio de aplicación,	Modelo del negocio	(A5): No se cuenta con la participación activa de los usuarios, clientes y stakeholders y/o el líder del proyecto no tiene experiencia en el descubrimiento de requerimientos y/o se cuenta con muy poco tiempo para el desarrollo del producto.
DEI6. Estrategia de entrega de documentación.	Elaboración y entrega de la documentación que guía a los usuarios y operadores a cómo usar la aplicación web eficientemente.	Aplicación web probada y lista para ser operada.	(A6): Se ha concluido con el desarrollo de la aplicación web y los usuarios y operadores requieren entrenamiento para operar y soportar la aplicación web y/o se dispone de la documentación que describe el uso eficiente de la aplicación web.
DEI7. Estrategia orientada a las pruebas de usabilidad.	Desarrollo de las pruebas de objetivos y funcionalidad de la aplicación web	Prototipo, aplicación web o cualquier sistema legado desarrollados	(A7): Existe un prototipo elaborado que se desea validar con el usuario y/o existe una aplicación web desarrollada y existe insatisfacción de parte de los usuarios y/o el sistema no se encuentra instalado en una plataforma adecuada.
DEI8. Estrategia de "historia de usuarios".	Extracción y fraccionamiento de las "historias de los usuarios" (requerimientos escritos) que representen más de un producto funcional.	Plan de desarrollo de la aplicación web. (tiempo y recursos estimados).	(A8): La aplicación web se desarrolla utilizando métodos ágiles y/o se requiere introducir rápidamente la aplicación web en el mercado y el grupo de desarrollo es experto en la implementación y pruebas de aplicaciones web.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Tabla No. 11. Directivas de Ejecución de Intención (DEI) del Mapa Global del Método Situacional para el Desarrollo de Aplicaciones Web (c)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumentos
DEI9. Estrategia de entrega de la aplicación web.	Instalación, operación y soporte de la aplicación web en su ambiente de ejecución.	Aplicación web desarrollada y planes de calidad y del proyecto.	(A9): Se culminó el desarrollo de la aplicación web y los resultados de las pruebas son satisfactorios y la organización dispone de la tecnología necesaria para implantar la aplicación web y/o el grupo de desarrollo cuenta con personas expertas en la instalación de aplicaciones web.
DEI10. Estrategia orientada por el diseño de la GUI.	Definición de la estructura de navegación, el estilo visual de las páginas y la interacción de la aplicación web con el usuario.	Requerimientos generales de la aplicación web.	(A10): Se requieren seguir los principios de diseño web y/o no es necesario reutilizar patrones de diseño y/o el grupo de desarrollo no dispone de un repositorio de componentes reutilizables y/o existe tiempo suficiente para desarrollar la aplicación web y/o la organización tiene bajo grado de madurez.
DEI11. Estrategia de retiro de la aplicación web	Culminación de la intención del mapa global.	Aplicación web operativa con bajo desempeño y fallas.	(A11): La aplicación web no soporta nuevas modificaciones y/o el mantenimiento de la aplicación web es muy costoso y/o la aplicación web no satisface en más de un 40% los requerimientos de los usuarios y/o la aplicación web ha disminuido considerablemente su desempeño y/o los cambios tecnológicos obligan a desarrollar una nueva aplicación web.
DEI12. Estrategia de Paralización	Culminación del desarrollo de la aplicación web y paralización de la implantación y mantenimiento de la misma.	Aplicación web desarrollada y lista para operar.	(A12): Se desarrolló la aplicación web y/o la organización no cuenta con los recursos necesarios para adquirir la plataforma de ejecución de la misma y/o los requerimientos de la organización cambiaron por movimientos realizados en la alta gerencia.
DEI13. Estrategia de completitud en la ejecución del plan.	Culminación de la intención del mapa global.	Documentación de la ejecución de las actividades, plan de calidad de la aplicación web.	(A13): Finalizó la ejecución de una fase de desarrollo de la aplicación web y/o el grupo de desarrollo tiene experiencia en el control y monitoreo de los proyectos y/o la organización tiene cierto grado de madurez.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Tabla No. 11. Directivas de Ejecución de Intención (DEI) del Mapa Global del Método Situacional para el Desarrollo de Aplicaciones Web (d)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumentos
DEI14. Estrategia de Reingeniería de procesos	Culminación de la intención del mapa global	Elementos de la organización que no pueden ser soportados por una aplicación web: objetivos, procesos, estructura, etc.	(A14): En la organización no existe disposición para aceptar el cambio tecnológico o los objetivos de la organización no pueden ser soportados por una aplicación web o los procesos de la organización son muy largos y engorrosos y no pueden ser automatizados o la estructura organizacional no está acorde con los procesos.
DEI15. Estrategia de modelado de negocios	Culminación de la intención del mapa global.	Plan del proyecto inconcluso e impreciso.	(A15): Se pretende desarrollar la aplicación web sin conocer el negocio y/o no se dispone de un modelo de negocio y/o el grupo de desarrollo y los líderes de la organización consideraron necesario modelar el negocio antes de desarrollar la aplicación web.
DEI16. Estrategia de revisión de actividades	Supervisión de los avances del proyecto para verificar el cumplimiento de las actividades y estándares y documentación de las decisiones tomadas.	Plan del proyecto.	(A16): El líder del proyecto toma la decisión de controlar la ejecución del proyecto y/o se sospecha que se han omitido algunos estándares y/o se produjeron retrasos en la entrega de algunos productos.
DEI17. Estrategia de rediseño de la aplicación web	Implementación de cambios en la aplicación web, los cuales implican la realización de un nuevo diseño de la misma.	Lista de cambios a implementar en la aplicación web, aplicación web operando.	(A17): Se requieren implementar cambios en la aplicación web para mejorar su desempeño y/o es necesario incorporar nuevos servicios a la aplicación web y/o disminuyó el número de visitas diarias realizadas a la aplicación web.
DEI18. Estrategia basada en la operación de la aplicación web.	Culminación de la intención del mapa global.	Aplicación web operando.	(A18): Se culminó el proceso de entrega de la aplicación web y/o no se contempló el mantenimiento dentro del plan del proyecto y/o la organización no dispone de los recursos necesarios para realizar mantenimiento a la aplicación web.
DEI19. Estrategia orientada por la administración del cambio.	Adaptación de nuevos requerimientos de los usuarios a la aplicación web, que garanticen el buen desempeño de la misma.	Nuevos requerimientos operativos de los usuarios.	(A19): La aplicación web está implantada y en pleno funcionamiento y/o dentro de la organización se cuenta con personas que conocen la aplicación web y existe un nuevo requerimiento del usuario y/o existe un nuevo requerimiento para mejorar el desempeño de la aplicación web y/o se detectaron errores en la aplicación web.

Licencia Creative Commons

3.3. MAPAS LOCALES DE ALTO NIVEL

A continuación se presentan los mapas locales de alto nivel que conforman al modelo de proceso para desarrollar aplicaciones web. Estos son los mapas que detallan las intenciones globales de “*gestionar el proyecto*”, “*desarrollar la aplicación web*” e “*implantar y mantener la aplicación web*”.

3.3.1. GESTIONAR EL PROYECTO

Gestionar el desarrollo de la aplicación web, tiene como objetivo principal la definición de los procesos con los cuales se establece el proyecto, como es coordinado y el cómo hacer uso efectivo de los recursos (tiempo, personas, dinero) para lograr los objetivos del mismo. Este mapa contiene tres (3) intenciones fundamentales: “*planificar el proyecto*”, “*asegurar la calidad de la aplicación web*” y “*controlar y monitorear el proyecto*” (ver figura 38).

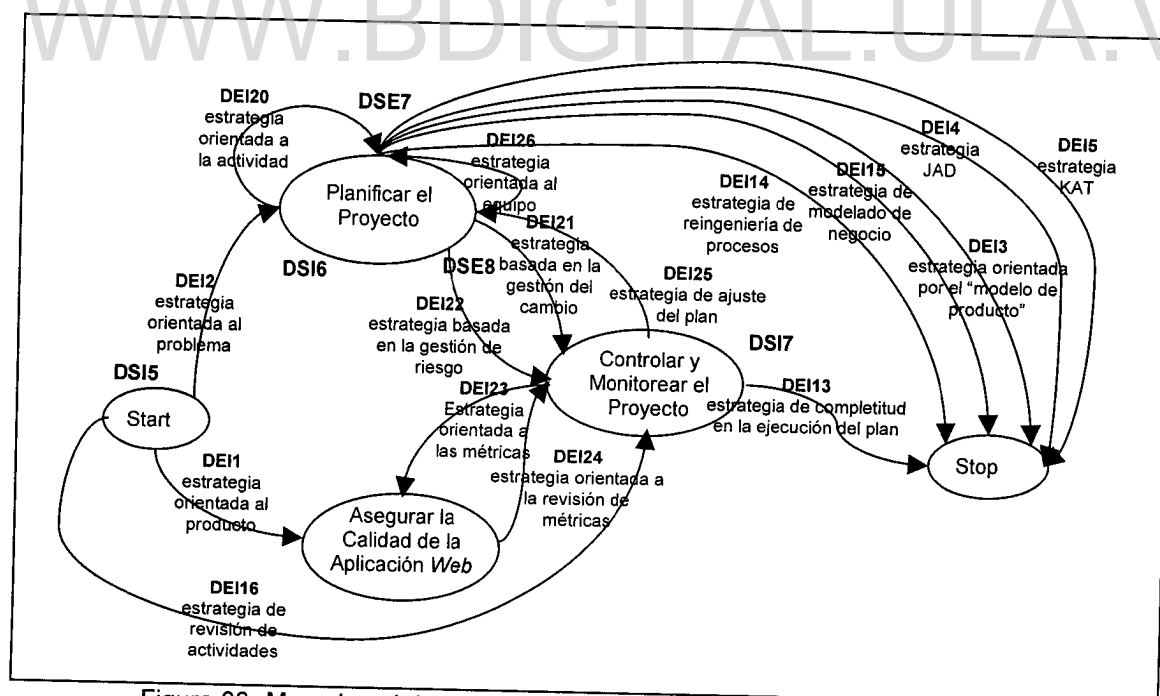


Figura 38. Mapa Local de Alto Nivel de la Intención de *Gestionar el Proyecto* del Método Situacional para el Desarrollo de Aplicaciones Web

La intención de “*planificar el proyecto*” tiene como objetivo el proporcionar un marco de trabajo que permita al gerente hacer estimaciones razonables de recursos,

costos y planificación temporal, es decir, que el gerente del proyecto tiene que estimar ¿cuánto durará?, ¿cuánto esfuerzo requerirá? Y ¿cuánta gente estará implicada?. Además, el planificador debe predecir los recursos (de hardware y software) que necesitará y el riesgo implicado.

El principal producto generado luego de ejecutada esta intención es el plan del proyecto, el cual debe: a) comunicar el ámbito y recursos a los gerentes del proyecto, personal técnico y al cliente; b) definir los riesgos y sugerir técnicas para mitigarlos; c) definir los costos y planificación temporal para la revisión de la gestión; d) proporcionar un enfoque general del desarrollo del proyecto para todo el personal relacionado con el mismo y e) describir cómo se garantizara la calidad y se gestionan los cambios.

La intención de *“asegurar la calidad de la aplicación web”* comprende procedimientos para la aplicación efectiva de métodos y herramientas, revisiones técnicas formales, técnicas y estrategias de pruebas, procedimiento de control de cambios, procedimiento de garantía de ajuste a los estándares y métricas de calidad de la aplicación definidas para la aplicación *web*.

El producto que se genera al ejecutar esta intención es el plan de calidad, el cual sirve como plantilla para las actividades de aseguramiento de la calidad instituidas en el proyecto de desarrollo de la aplicación *web*. Este plan debe identificar: evaluaciones a realizar, auditorias y revisiones a realizar, estándares que se pueden aplicar a la aplicación *web*, procedimientos para información y seguimiento de errores, documentos producidos por el grupo de calidad y realimentación de información proporcionada al equipo del proyecto.

La intención de *“controlar y monitorear el proyecto”* consiste en hacer un seguimiento al estado actual del proyecto, con el fin de identificar problemas o posibles desviaciones del plan del proyecto o el plan de calidad. En caso de existir tales desviaciones, el equipo de líderes del proyecto debe redefinir las actividades del plan en cuanto a tiempo y recursos. Asimismo, el gerente del proyecto debe renegociar con el cliente los nuevos cambios.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

El producto generado al finalizar la ejecución de esta intención es un informe donde se describe el estado actual del proyecto, los posibles riesgos, y una redefinición de las actividades y recursos a incorporar al proyecto.

Este mapa está conformado por los siguientes elementos:

- **5 Intenciones**, las cuales representan el objetivo que se desea satisfacer en el proceso. Estas son: Comienzo, planificar el proyecto, asegurar la calidad de la aplicación *web*, controlar y monitorear el proyecto y parada
- **2 Directivas de Selección de Estrategia (DSE)**, las cuales permiten seleccionar una estrategia para ejecutar una intención de modelado, a partir de una intención de partida. Estas son:
 - a) **DSE7**: De la cual se puede seleccionar la estrategia “*kAT*”, la estrategia “*JAD*”, la estrategia “orientada por el modelo de producto”, la estrategia de “modelado del negocio” o la estrategia de “reingeniería de procesos”, con el fin de partir de la intención de “*planificar el proyecto*” hasta la intención de “*parada*”.
 - b) **DSE8**: A partir de esta directiva se puede seleccionar la estrategia “basada en la gestión de riesgo”, la estrategia “basada en la gestión del cambio” o las dos anteriores, con el fin de partir de la intención de “*planificar el proyecto*” hacia la intención de “*controlar y monitorear el proyecto*”.
- **3 Directivas de Selección de Intención (DSI)**, las cuales permiten seleccionar una intención destino entre las disponibles a partir de una intención de partida. Las mismas están identificadas como DSI5, DSI6 y DSI7.
- **7 Directivas de Ejecución de Intención (DEI)**, las cuales guían la ejecución de una intención según una estrategia específica. Estas se listan en la tabla No.12 donde se indica, su descripción, situación inicial y argumentos para la ejecución de las mismas.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Tabla No. 12. Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto Nivel de la Intención de *Gestionar el Proyecto* del Método Situacional para el Desarrollo de Aplicaciones Web

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI20. Estrategia orientada a la actividad.	Determinación de las actividades necesarias relevantes en cada una de las fases de desarrollo.	Tiempo, recursos y personal necesario para la ejecución del proyecto.	(A20): Existe un modelo de negocio y se conocen los requerimientos generales con que debe cumplir la aplicación web y/o el equipo tiene experiencia en el desarrollo de aplicaciones web.
DEI21. Estrategia basada en la gestión del cambio.	Identificación, control y garantía de que los cambios se implementan adecuadamente.	Plan del proyecto	(A21): Se tiene elaborado el plan del proyecto y/o la organización tiene cierto grado de madurez y/o el equipo de desarrollo conoce la importancia de controlar el cambio y/o una de las metas del grupo de desarrollo es maximizar la productividad a través de la minimización de errores.
DEI22. Estrategia basada en la gestión del riesgo.	Identificación de los costos, esfuerzo, recursos y riesgos técnicos en el desarrollo de la aplicación web, determinación de la probabilidad de ocurrencia, impacto y tiempo en que puedan ocurrir los riesgos y determinación de las estrategias de mitigación de esos riesgos.	Plan del proyecto	(A22): Se tiene elaborado el plan del proyecto y/o la organización tiene cierto grado de madurez y/o el equipo de desarrollo tiene experiencia en la gestión de riesgo.
DEI23. Estrategia orientada a las métricas.	Establecimiento de los estándares de calidad en cada fase del proyecto, definición de las metas de calidad que se desean alcanzar y documentación del plan de calidad.	Plan del proyecto, características generales del producto.	(A23): La organización tiene cierto grado de madurez y/o no se han definido las métricas de calidad a ser controladas y/o el equipo de desarrollo cuenta con la experiencia necesaria para alcanzar las metas de calidad planteadas.
DEI24. Estrategia orientada a la revisión de métricas.	Revisión técnica y auditoria de los productos elaborados y toma de acciones correctivas, en caso de existir desviaciones.	Plan de calidad de la aplicación web y productos elaborados.	(A24): Se ha generado un producto al finalizar una fase de desarrollo de la aplicación web y/o la organización tiene cierto grado de madurez y/o el grupo de desarrollo tiene experiencia en el control de calidad de las aplicaciones web.
DEI25. Estrategia de ajuste del plan.	Revisión del plan del proyecto, con el fin de verificar si está acorde con la ejecución real de las actividades para el desarrollo de la aplicación web y corrección de las desviaciones que pudiesen encontrarse	Informe de decisiones tomadas en el control del proyecto (ajuste de recursos, tiempo)	(A25): Existen retrasos en la entrega de los productos y/o los clientes se quejan de la calidad de los productos entregados y/o el grupo de desarrollo se ocupa de llevar un control, lo más cercano a la realidad, de los aspectos gerenciales del proyecto.
DEI26. Estrategia orientada al equipo	Definición del equipo de trabajo de acuerdo a los roles, reglas, conocimientos y habilidades requeridas y en asegurar el entendimiento del equipo de trabajo	Plan del proyecto, plan de calidad.	(A26): El líder del proyecto tiene experiencia en la organización de equipos de trabajo y se cuenta con el personal que posee los conocimientos y habilidades requeridas para desarrollar la aplicación web y están definidas las métricas de calidad del producto.

Licencia Creative Commons

3.3.2. DESARROLLAR LA APLICACIÓN WEB

Desarrollar la aplicación *web* consiste del proceso orientado a la construcción de la misma. Este mapa contiene cuatro (4) intenciones fundamentales: “definir y especificar requerimientos”, “diseñar la aplicación *web*”, “implementar la aplicación *web*” y “probar la aplicación *web*” (ver figura 39).

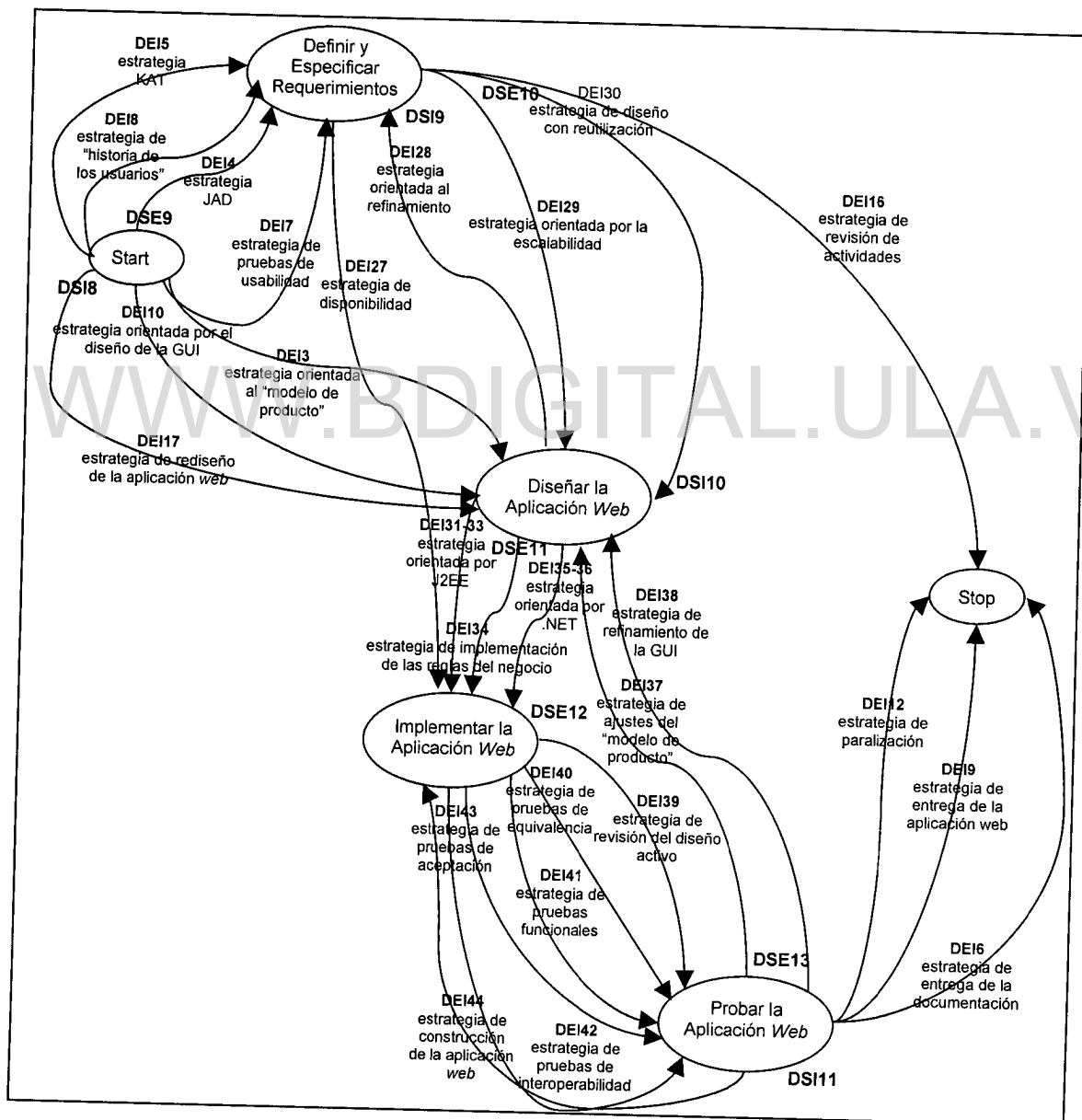


Figura 39. Mapa Local de Alto Nivel de la Intención de *Desarrollar la Aplicación Web* del Método Situacional para el Desarrollo de Aplicaciones Web

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

La intención de “*definir y especificar requerimientos*” tiene como propósito el capturar, definir y especificar tanto los requerimientos funcionales como los no funcionales. Los productos generados, luego de ejecutada esta intención son: el documento de definición de requerimientos y el documento de especificación de requerimientos.

El documento de definición de requerimientos resume en lenguaje natural los requerimientos funcionales (servicios que debe proveer la aplicación *web*) y no funcionales (velocidad, seguridad, capacidad de almacenamiento, etc.). El documento de especificación de requerimientos contiene almacenados los requerimientos de manera formal, como por ejemplo, los modelos que forman parte del Lenguaje de Modelado Unificado (UML).

La intención de “*diseñar la aplicación web*” consiste en describir todos los aspectos de la aplicación *web* que se van a implementar. Los productos que se generan al ejecutar esta intención son: documento de diseño de la arquitectura, documento de diseño de la interfaz de usuario, documento de diseño de las bases de datos, el documento de los objetos del negocio y documento de los componentes.

El documento de diseño de la arquitectura describe los diseños de la arquitectura lógica y de ejecución. El documento de diseño de la interfaz de usuario define la estructura de la Interfaz Gráfica de Usuario (GUI), el flujo de navegación, el estilo de las páginas *web* correspondientes a la capa de presentación de la aplicación *web*.

El documento de diseño de las bases de datos contiene el diseño conceptual de las bases de datos correspondientes a la capa de datos de la aplicación *web*. El documento de los objetos del negocio representa el diseño de la lógica del negocio, a través de la descripción de los objetos del negocio, flujo de procesos y reglas del negocio que serán implementadas en la aplicación *web*. El documento de componentes representa el diseño de cada componente de cada una de las capas de la aplicación *web* (presentación, lógica del negocio y datos) en forma de notación gráfica en un nivel de abstracción muy cercano al código fuente.

La intención de “*implementar la aplicación web*” tiene como finalidad la implementación del diseño producido anteriormente, así como también, el enlace de

los componentes de la capa de presentación con los componentes de la capa de lógica del negocio y el de los componentes de la capa de lógica del negocio con los componentes de la capa de datos, utilizando el “*middleware*” definido. Además, permite integrar la aplicación *web* generada con cualquier otra aplicación existente.

Los principales productos de esta fase son: los componentes de la capa de presentación, los componentes de la capa de lógica del negocio y los componentes de la capa de datos implementados, la aplicación *web* ensamblada y los documentos de la aplicación *web* indicados en el plan del proyecto.

La intención de “*probar la aplicación web*” consiste en medir los logros alcanzados en cuanto a la calidad de la aplicación *web* realizada y la satisfacción que esta ofrece a los clientes. El principal producto que se genera es la aplicación *web* probada, así como los documentos obtenidos luego de realizadas las pruebas, los cuales deben contener los resultados esperados y los resultados obtenidos.

El mapa está conformado por los siguientes elementos:

- **6 Intenciones**, las cuales representan el objetivo que se desea satisfacer en el proceso. Estas son: *Comienzo, definir y especificar requerimientos, diseñar la aplicación web, implementar la aplicación web, probar la aplicación web y parada.*
- **7 Directivas de Selección de Estrategia (DSE)**, las cuales permiten seleccionar una estrategia para ejecutar una intención de modelado, a partir de una intención de partida. Estas son:
 - a) **DSE9:** De la cual se puede seleccionar la estrategia “*KAT*”, la estrategia “*JAD*”, la estrategia de “*historia de los usuarios*” o la estrategia de “*pruebas de usabilidad*”, con el fin de ejecutar la intención de “*definir y especificar requerimientos*”.
 - b) **DSE10:** A partir de esta directiva se puede seleccionar la estrategia “*orientada por la escalabilidad*”, la estrategia de “*diseño con Reutilización*”, o las dos anteriores, con el fin de partir de la intención de “*definir y especificar requerimientos*” hacia la intención de “*diseñar la aplicación web*”.

Licencia Creative Commons:

- c) **DSE11:** A partir de esta directiva se puede seleccionar la estrategia “orientada por *J2EE*”, (que a su vez comprende tres posibles estrategias: estrategia “orientada por los *servlets*”, estrategia de “*applets*” o estrategia “orientada por *JSP*”), la estrategia de “implementación de las reglas del negocio” o la estrategia “orientada por *.NET*”, (que a su vez comprende dos posibles estrategias: estrategia “orientada por el *ASP.NET*” o estrategia “orientada por el *MMIT*”), con el fin de partir de la intención de “*diseñar la aplicación web*” hacia la intención de “*implementar la aplicación web*”.
- d) **DSE12:** A partir de esta directiva se puede seleccionar la estrategia de “revisión del diseño activo”, la estrategia de “pruebas de equivalencia”, la estrategia de “pruebas funcionales”, o todas anteriores, con el fin de partir de la intención de “*implementar la aplicación web*” hacia la intención de “*probar la aplicación web*”.
- e) **DSE13:** A partir de esta directiva se puede seleccionar la estrategia de “ajustes del modelo de producto”, la estrategia de “refinamiento de la *GUI*”, o las dos anteriores, con el fin de partir de la intención de “*probar la aplicación web*” hacia la intención de “*diseñar la aplicación web*”.
- f) **DSE14:** A partir de esta directiva se puede seleccionar la estrategia de “entrega de la documentación”, la estrategia de “entrega de la aplicación *web*”, o las dos anteriores, con el fin de partir de la intención de “*probar la aplicación web*” hacia la intención de “*parada*”.
- g) **DSE15:** A partir de esta directiva se puede seleccionar la estrategia “orientada al modelo de producto”, la estrategia “orientada por el diseño de la *GUI*”, las dos anteriores o la estrategia de “rediseño de la aplicación *web*”, con el fin de “*diseñar la aplicación web*”.
- **4 Directivas de Selección de Intención (DSI)**, las cuales permiten seleccionar una intención destino entre las disponibles a partir de una

Licencia Creative Commons:

intención de partida. Las mismas están identificadas como DSI8, DSI9, DSI10 y DSI11.

- **18 Directivas de Ejecución de Intención (DEI)**, las cuales guían la ejecución de una intención según una estrategia específica. Estas se listan en la tabla No.13 donde se indica, su descripción, situación inicial y argumentos para la ejecución de las mismas.

Tabla No. 13. Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto Nivel de la Intención de *Desarrollar la Aplicación Web* del Método Situacional para el Desarrollo de *Aplicaciones Web* (a)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI27. Estrategia de disponibilidad	Construcción del código fuente y preparación de las pruebas.	"Historia de los usuarios"(requerimientos)	(A26):Se requiere una rápida implementación de la "historia de los usuarios" y/o el grupo de desarrollo tiene experiencia en desarrollar aplicaciones <i>web</i> bajo el enfoque ágil y/o se dispone de componentes reutilizables de fácil adaptación en la aplicación <i>web</i> .
DEI28. Estrategia orientada al refinamiento	Refinamiento de la definición y especificación de requerimientos, luego que ha habido un acuerdo entre el cliente y el grupo de desarrollo.	Prototipo de la aplicación <i>web</i>	(A27).Se diseñó un prototipo de la aplicación <i>web</i> y/o el cliente y el grupo de desarrollo definieron claramente las expectativas que debe satisfacer la aplicación <i>web</i> y/o los requerimientos no estaban completamente definidos y especificados.
DEI29. Estrategia orientada por la escalabilidad	Medición de las posibilidades de crecimiento de la aplicación <i>web</i> y selección de la plataforma de <i>hardware</i> y <i>software</i> más adecuada.	Documento de definición de requerimientos	(A29): Se cuenta con el documento de requerimientos de la aplicación <i>web</i> y/o se dispone de la tecnología adecuada para implantar la aplicación <i>web</i> .
DEI30. Estrategia de diseño con Reutilización	Búsqueda y Reutilización de un patrón para diseñar la aplicación <i>web</i> .	Documento de especificación de requerimientos de la aplicación <i>web</i> .	(A30): El equipo de desarrollo posee experiencia en el reuso de patrones de diseño y/o se requiere mejorar la productividad en el desarrollo de la aplicación <i>web</i> y/o se requiere colocar rápidamente la aplicación <i>web</i> en el mercado y se dispone de una documentación de patrones de diseño adecuada.
DEI31. Estrategia orientada por <i>SERVLET</i> (Estrategia orientada por <i>J2EE</i>)	Construcción de <i>SERVLET</i> para implementar la capa de presentación de la aplicación <i>web</i> .	Plataforma <i>J2EE</i> , documentación del diseño de la aplicación <i>web</i> .	(A31):Se seleccionó <i>J2EE</i> como plataforma que soportará la aplicación <i>web</i> y/o se reutilizó el patrón del cliente <i>web</i> ligero y/o se requiere que las páginas <i>web</i> sean procesadas en el servidor y/o se seleccionó <i>JAVA</i> como lenguaje de programación para procesar las peticiones del cliente.

Licencia Creative Commons.

Tabla No. 13. Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto Nivel de la Intención de *Desarrollar la Aplicación Web* del Método Situacional para el Desarrollo de Aplicaciones Web (b)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI32. Estrategia de <i>applets</i> (Estrategia orientada por <i>J2EE</i>)	Construcción de <i>applet</i> para implementar la capa de presentación de la aplicación <i>web</i> .	Plataforma <i>J2EE</i> , documentación del diseño de la aplicación <i>web</i> .	(A32): Se seleccionó <i>J2EE</i> como plataforma que soportará la aplicación <i>web</i> y/o se reutilizó el patrón del cliente <i>web</i> pesado y/o se requiere que las páginas <i>web</i> sean procesadas en el cliente y/o se seleccionó <i>JAVA</i> como lenguaje de programación para procesar las peticiones del cliente.
DEI33. Estrategia orientada por <i>JSP</i> (Estrategia orientada por <i>J2EE</i>)	Construcción de <i>JSP</i> para implementar la capa de presentación de la aplicación <i>web</i> .	Plataforma <i>J2EE</i> , documentación del diseño de la aplicación <i>web</i> .	(A33): Se seleccionó <i>J2EE</i> como plataforma que soportará la aplicación <i>web</i> y/o se reutilizó el patrón del cliente <i>web</i> ligero y/o se requiere que las páginas <i>web</i> sean procesadas en el servidor y/o se seleccionaron los lenguajes <i>HTML</i> y <i>XML</i> entremezclados con <i>JAVA</i> para procesar las peticiones del cliente.
DEI34. Estrategia de implementación de las reglas del negocio	Construcción del código fuente de las reglas del negocio que serán implementadas en la aplicación <i>web</i> .	Documentación del diseño de la aplicación <i>web</i> .	(A34): Se conocen claramente las reglas del negocio y/o Se diseñó y aprobó la <i>GUI</i> y/o se requieren implementar y validar las reglas del negocio para continuar con el desarrollo de la aplicación <i>web</i> .
DEI35. Estrategia orientada por el <i>ASP.NET</i> (Estrategia orientada por <i>.NET</i>)	Construcción de un <i>ASP.NET</i> para implementar la capa de presentación de la aplicación <i>web</i> .	Plataforma <i>.NET</i> , documentación del diseño de la aplicación <i>web</i> .	(A35): Se seleccionó <i>.NET</i> como plataforma que soportará la aplicación <i>web</i> y/o se reutilizó el patrón del cliente <i>web</i> ligero y/o se requiere que las páginas <i>web</i> sean procesadas en el servidor y/o el grupo de desarrollo tiene experiencia en el uso del lenguaje <i>VISUAL BASIC</i> .
DEI36. Estrategia orientada por el <i>MMIT</i> (Estrategia orientada por <i>.NET</i>)	Uso de la herramienta <i>Microsoft Mobile Internet Toolkit (MMIT)</i> para implementar la capa de presentación de la aplicación <i>web</i> .	Plataforma <i>.NET</i> , documentación del diseño de la aplicación <i>web</i> .	(A36): Se seleccionó <i>.NET</i> como plataforma que soportará la aplicación <i>web</i> y se requiere acceder a la aplicación <i>web</i> desde un teléfono móvil o un dispositivo <i>PDA</i> y/o se reutilizó el patrón del cliente <i>web</i> ligero y se requiere que las páginas <i>web</i> sean procesadas en el servidor.
DEI37. Estrategia de ajustes del "modelo de producto"	Realización de cambios en el modelo de producto que garanticen la integración de todos los componentes que constituyen la aplicación <i>web</i> .	Modelo de producto, documentación de las pruebas de integración de la aplicación <i>web</i> .	(A37): Se realizaron las pruebas de integración de la aplicación <i>web</i> y/o se detectaron fallas en la interacción de los componentes que conforman la aplicación <i>web</i> y/o no existen fallas a nivel de implementación y/o se sospechan de fallas a nivel de diseño y/o se presume que se haya omitido algún componente fundamental en el modelo de producto de la aplicación <i>web</i> .
DEI38. Estrategia de refinamiento de la <i>GUI</i>	Adaptación de la <i>GUI</i> a las necesidades planteadas por los clientes y usuarios de la aplicación <i>web</i> .	<i>GUI</i> de la aplicación <i>web</i> , lista de cambios solicitados por los clientes y usuarios.	(A38): Se realizaron las pruebas de la aplicación <i>web</i> y/o la aplicación <i>web</i> no cumple con todas las funcionalidades exigidas por los clientes y usuarios y/o los clientes solicitaron nuevos cambios antes de implantar la aplicación <i>web</i> .

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Tabla No. 13. Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto Nivel de la Intención de *Desarrollar la Aplicación Web* del Método Situacional para el Desarrollo de Aplicaciones Web (c)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI39. Estrategia de revisión del diseño activo	Inspección de un componente de la aplicación <i>web</i> por todos los miembros del equipo de desarrollo.	Componente implementado.	(A39): Se está desarrollando una aplicación <i>web</i> compleja con requerimientos de seguridad críticos y/o se requieren encontrar distintos tipos de fallas en un componente y/o se requiere de la inspección realizada por cada uno de los miembros del grupo de desarrollo.
DEI40. Estrategia de pruebas de equivalencia	Prueba de una unidad de la aplicación <i>web</i> a través de la técnica de la "caja negra".	Unidad de la aplicación <i>web</i> implementada.	(A40): Se implementó una unidad de la aplicación <i>web</i> y/o se seleccionó una unidad de la aplicación <i>web</i> para disminuir la complejidad de las pruebas y/o se requiere disminuir el número de casos de prueba y/o las personas responsables de realizar las pruebas conocen los requerimientos de entrada y salida de la aplicación <i>web</i> y/o las personas responsables de realizar las pruebas no conocen el contenido de las unidades de la aplicación <i>web</i> .
DEI41. Estrategia de pruebas funcionales	Determinación de las diferencias entre los requerimientos funcionales y la aplicación <i>web</i> .	Documento de especificación de requerimientos y aplicación <i>web</i> operativa.	(A41): Se dispone de la aplicación <i>web</i> operativa y se dispone del documento de especificación de requerimientos y/o se requieren probar aquellos servicios que son relevantes para los usuarios y clientes.
DEI42. Estrategia de pruebas de interoperabilidad	Prueba de la integración de la aplicación <i>web</i> con otras aplicaciones existentes.	Aplicación <i>web</i> operativa, aplicación(es) existente(s) funcionando.	(A42): Se requiere integrar la aplicación <i>web</i> con otras aplicaciones existentes y/o se cuenta con la plataforma necesaria para integrar ambas aplicaciones y/o el grupo de desarrollo dispone de la documentación detallada de las aplicaciones existentes y/o se cuenta con el modelo de sistema de información del modelo de negocio.
DEI43. Estrategia de pruebas de aceptación	Prueba de la aplicación <i>web</i> realizada por el grupo de desarrollo, los clientes y usuarios.	Aplicación <i>web</i> operativa, documento de definición de requerimientos de la aplicación <i>web</i>	(A43): Se requiere que los clientes preparen las pruebas de la aplicación <i>web</i> y/o los clientes y usuarios conocen los requerimientos de la aplicación <i>web</i> y/o la aplicación <i>web</i> reemplazara a una que está actualmente operativa, o que se retiró recientemente.
DEI44. Estrategia de construcción de la aplicación <i>web</i>	Construcción progresiva de la aplicación <i>web</i> , luego de implementar y probar cada unidad por separado.	Unidad de la aplicación <i>web</i> implementada y probada, modelo de producto de la aplicación <i>web</i> .	(A44): Se requiere implementar y probar cada unidad de la aplicación <i>web</i> y/o es necesario disponer de un servicio particular de la aplicación <i>web</i> y/o la aplicación <i>web</i> a desarrollar es muy compleja y/o se requiere garantizar la calidad de cada uno de los servicios que son implementados en la aplicación <i>web</i> .

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

3.3.3. IMPLANTAR Y MANTENER LA APLICACIÓN WEB

Implantar y mantener la aplicación *web* consiste en la instalación, mantenimiento, operación y soporte y adiestramiento de los usuarios y operadores que manipularán la misma. Este mapa está conformado por cuatro (4) intenciones fundamentales: “instalar la aplicación *web*”, “adiestrar a los usuarios y operadores”, “operar la aplicación *web*” y “mantener la aplicación *web*” (ver figura 40).

La intención de “instalar la aplicación *web*” consiste en implantar la aplicación *web* en su ambiente de ejecución, y verificar que la misma trabaja adecuadamente. El producto generado luego de ejecutar esta intención es la aplicación *web* instalada y lista para ser operada.

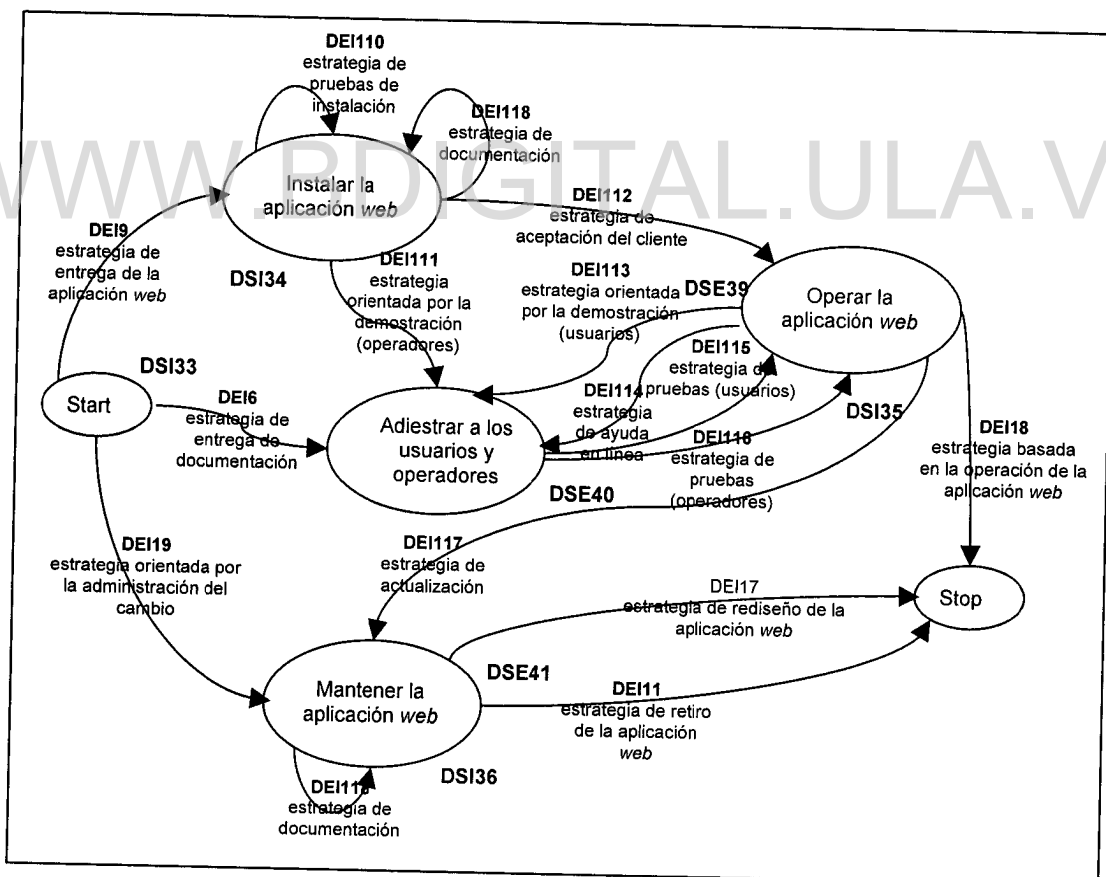


Figura 45. Mapa Local de Alto Nivel de la Intención de *Implantar y Mantener la Aplicación Web* del Método Situacional para el Desarrollo de Aplicaciones Web

La intención de “adiestrar a los usuarios y operadores” tiene por objetivo el que los usuarios y operadores reciban el entrenamiento planificado en el plan del

proyecto. El producto que se obtiene son los operadores y usuarios adiestrados listos para operar y soportar la aplicación *web*.

La intención de “operar la aplicación *web*” consiste en que los usuarios y operadores operen y le den soporte a la aplicación *web* con datos reales. Esta intención, generalmente, se lleva a cabo en presencia de los instructores que dictaron el entrenamiento, con el fin de garantizar que las primeras operaciones se ejecutan correctamente. El producto generado es la aplicación *web* operada y soportada.

La intención de “mantener la aplicación *web*” consiste en realizar cualquier tipo de cambios a la misma, luego que ésta es operada y soportada por los usuarios y operadores. Estos cambios pueden ser de simple actualización, los cuales deben realizarse constantemente, o para corregir fallas (correctivos), o para agregar un nuevo servicio a la aplicación *web* (adaptativos) o para mejorar el desempeño de la aplicación *web* (perfectivo). El producto generado luego de la ejecución de esta intención es la aplicación *web* mantenida y toda la documentación que esto implica.

Este está conformado por los siguientes elementos:

- **6 Intenciones**, las cuales representan el objetivo que se desea satisfacer en el proceso. Estas son: *Comienzo, instalar la aplicación web, adiestrar a los usuarios y operadores, operar la aplicación web, mantener la aplicación web y parada.*
- **3 Directivas de Selección de Estrategia (DSE)**, las cuales permiten seleccionar una estrategia para ejecutar una intención de modelado, a partir de una intención de partida. Estas son:
 - a) **DSE39:** De la cual se puede seleccionar la estrategia “orientada por la demostración (usuarios)”, la estrategia de “ayuda en línea”, o todas las anteriores, con el fin de partir de la intención de “operar la aplicación *web*” hacia la “intención de adiestrar a los usuarios y operadores”.
 - b) **DSE40:** A partir de esta directiva se puede seleccionar la estrategia de “pruebas (usuarios)”, la estrategia de “pruebas (operadores)”, o las dos anteriores, con el fin de partir de la intención de “adiestrar a

los usuarios y operadores” hacia la intención de “operar la aplicación web”.

- c) **DSE41:** A partir de esta directiva se puede seleccionar la estrategia de “rediseño de la aplicación web”, o la estrategia de “retiro de la aplicación web”.
- **4 Directivas de Selección de Intención (DSI)**, las cuales permiten seleccionar una intención destino entre las disponibles a partir de una intención de partida. Las mismas están identificadas como DSI33, DSI34, DSI35 y DSI36.
- **9 Directivas de Ejecución de Intención (DEI)**, las cuales guían la ejecución de una intención según una estrategia específica. Estas se listan en la tabla No.14 donde se indica, su descripción, situación inicial y argumentos para la ejecución de las mismas.

Tabla No. 14 Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto nivel de la Intención de *Implantar y Mantener la Aplicación Web* del Método Situacional para el Desarrollo de Aplicaciones Web (a)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI110. Estrategia de pruebas de instalacion	Prueba de la instalación de la aplicación web en el ambiente de ejecución.	Arquitectura de ejecución y aplicación web operativa.	(A110):Las pruebas de sistema no fueron realizadas en el ambiente de ejecución y/o se requieren realizar las pruebas de la aplicación web con el cliente y/o se requiere verificar la completitud de la aplicación web instalada y/o se requiere verificar los requerimientos funcionales y no funcionales que se pueden ver afectados con la instalación en el ambiente de ejecución.
DEI111. Estrategia orientada por la demostración (operadores)	Entrenamiento basado en la demostración y clases dirigidas a los operadores.	Aplicación web operando y operadores.	(A111):Se requiere realizar un entrenamiento dinámico, interactivo y flexible y/o se cuenta con las herramientas necesarias para realizar un entrenamiento basado en computadora y/o se requiere verificar el aprendizaje obtenido por el operador a través de un <i>feedback</i> .

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Tabla No. 14 Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto Nivel de la Intención de *Implantar y Mantener la Aplicación Web* del Método Situacional para el Desarrollo de Aplicaciones Web (b)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI112. Estrategia de aceptación del cliente	Inicio de la operación y soporte de la aplicación web una vez que el cliente y/o los usuarios la han probado.	Aplicación web.	(A112): El cliente está conforme con la implementación de los requerimientos funcionales y no funcionales y/o es necesario que los clientes y el grupo de desarrollo dialoguen para verificar los cambios que hayan sufrido los requerimientos y/o se requiere entregar la aplicación web al cliente.
DEI113. Estrategia orientada por la demostración (usuarios)	Entrenamiento basado en la demostración y clases dirigidas a los usuarios de la aplicación web.	Aplicación web operando y usuarios.	(A113): Se requiere realizar un entrenamiento dinámico, interactivo y flexible y/o se cuenta con las herramientas necesarias para realizar un entrenamiento basado en computadora y/o se requiere verificar el aprendizaje obtenido por los usuarios a través de un <i>feedback</i> .
DEI114. Estrategia de pruebas (usuarios)	Ejercitación de las principales funciones que satisfacen los requerimientos funcionales y no funcionales de la aplicación web.	Aplicación web operando, documento de definición de requerimientos y usuarios.	(A114): El entrenamiento del usuario es guiado por las funciones principales de la aplicación web, y/o se dispone de la documentación detallada de los requerimientos de la aplicación web y/o el usuario debe navegar y acceder a todas las páginas que conforman a la aplicación web.
DEI115. Estrategia de ayuda en línea	Adiestramiento basado en la asistencia automatizada a través de menú o etiquetas de ayuda en la aplicación web.	Aplicación web operando.	(A115): La aplicación web dispone de ayuda en línea y/o la aplicación web es muy fácil de manejar y/o los usuarios de la aplicación web son expertos y no requieren de entrenamiento basado en demostración.
DEI116. Estrategia de pruebas (operadores)	Ejercitación de las principales funciones de soporte de la aplicación web.	Aplicación web operando, funciones de soporte..	(A116): El entrenamiento del usuario es guiado por las funciones principales de soporte de la aplicación web, y/o el operador debe configurar la aplicación web y/o el operador debe conocer como conceder o denegar acceso a los usuarios y/o el operador debe garantizar el almacenamiento adecuado de los datos.
DEI117. Estrategia de actualización	Actualización periódica de la aplicación web.	Aplicación web operando.	(A117): La aplicación web está operando en su ambiente de ejecución y/o es necesario actualizar su información periódicamente y/o se dispone de un equipo encargado de realizar las actualizaciones y/o la empresa tiene bien definidas las políticas de mantenimiento de una aplicación web.

Licencia Creative Commons

Tabla No. 14 Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto Nivel de la Intención de *Implantar y Mantener la Aplicación Web* del Método Situacional para el Desarrollo de Aplicaciones *Web* (c)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI118. Estrategia de documentación.	Elaboración del documento donde se especifican las pruebas de instalación, las características de la aplicación <i>web</i> , su forma de uso, soporte, operación y mantenimiento de la misma.	Aplicación <i>web</i> operando y en mantenimiento.	(A118): La aplicación <i>web</i> se encuentra operando en su ambiente de ejecución y/o se dispone de un plan de adiestramiento a los usuarios y operadores de la aplicación <i>web</i> y/o la organización tiene cierto grado de madurez.

3.4. MAPAS LOCALES DE BAJO NIVEL: INTENCIÓN “DESARROLLAR LA APLICACIÓN WEB”

El objetivo fundamental de esta investigación, es proponer el modelo de producto y el modelo de proceso de un método situacional para desarrollar aplicaciones *web*. Por tal razón, es necesario detallar la intención de desarrollar la aplicación *web*, dejando para trabajos futuros el detalle de las intenciones de gestionar el proyecto e implantar y mantener la aplicación *web*.

A continuación se presentan los mapas locales de bajo nivel que detallan la intención de desarrollar la aplicación *web*. Estos son: definir y especificar requerimientos, diseñar la aplicación *web*, implementar la aplicación *web* y probar la aplicación *web*.

3.4.1. DEFINIR Y ESPECIFICAR REQUERIMIENTOS

Definir y especificar requerimientos tiene como propósito el capturar, definir y especificar tanto los requerimientos funcionales como los no funcionales. Este mapa contiene tres (3) intenciones fundamentales: “*Capturar requerimientos*”, “*definir requerimientos*” y “*especificar requerimientos*” (ver figura 41).

La intención de “*capturar requerimientos*” consiste en preguntar al cliente, a los usuarios y a los que están involucrados en los objetivos de la aplicación *web*

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

sobre sus necesidades, investigar cómo la aplicación *web* se ajustará a las necesidades del negocio, y cómo la misma será utilizada en el día a día.

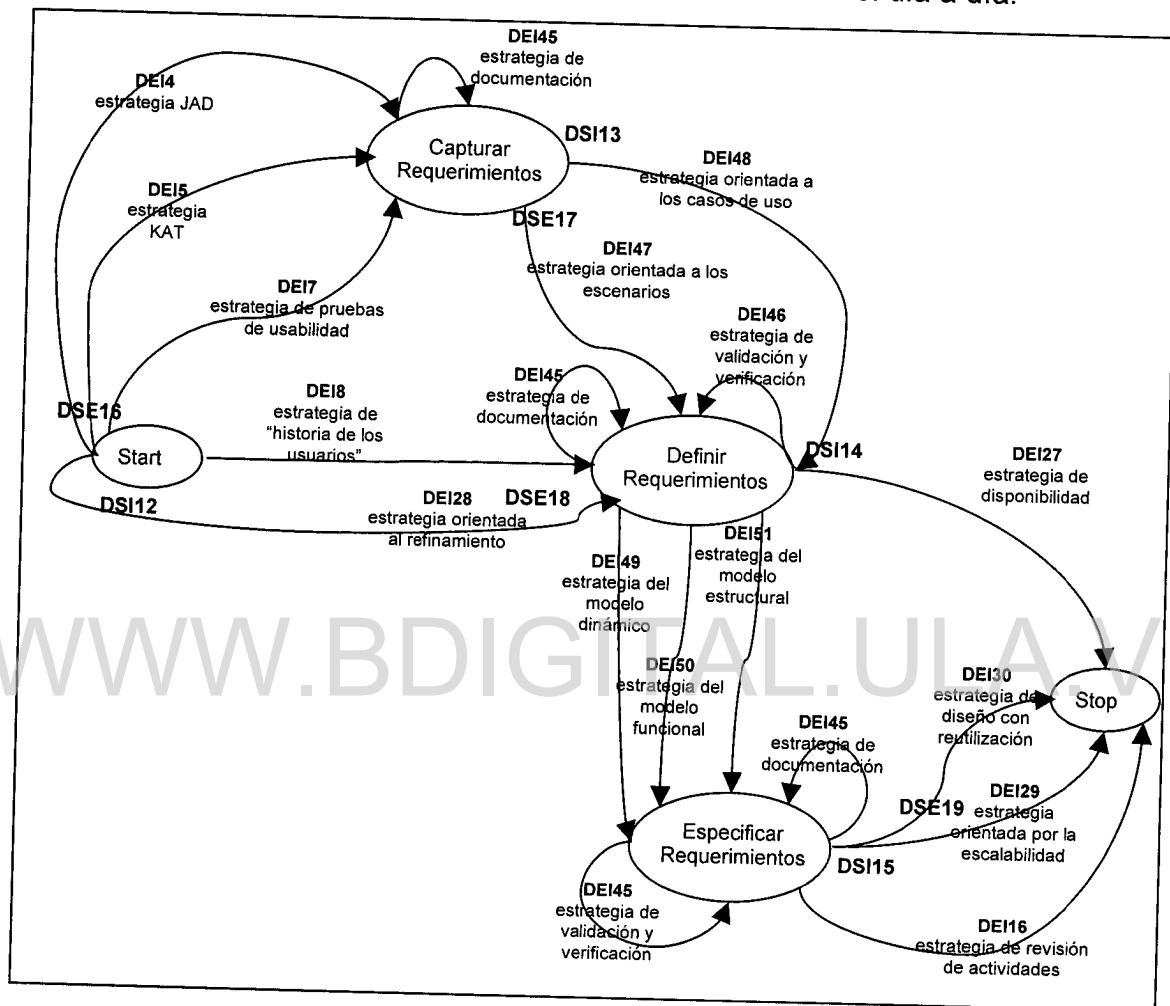


Figura 41. Mapa Local de Bajo Nivel de la Intención de *Definir* y *Especificar Requerimientos* del Método Situacional para el Desarrollo de Aplicaciones Web

El resultado alcanzado como consecuencia de la ejecución de esta intención variará dependiendo de la estrategia utilizada para capturar los requerimientos. Los aspectos principales que debe contener el producto obtenido deben incluir: una relación de necesidades y características, un informe conciso del alcance de la aplicación *web* y una lista de clientes, usuarios y otros actores que deben de participar en la captura de requerimientos.

La intención de "definir requerimientos" tiene por objetivo la construcción de una lista completa de todas las necesidades que el cliente y los usuarios esperan

satisfacer de la aplicación *web* a desarrollar. La misma representa el principal medio de comunicación entre el cliente, los usuarios y el grupo de desarrollo, y constituye el elemento principal del documento de definición de requerimientos.

El documento de definición de requerimientos resume en lenguaje natural los requerimientos funcionales (servicios que debe proveer la aplicación *web*) y no funcionales (velocidad, seguridad, capacidad de almacenamiento, etc.).

La intención de “*especificar requerimientos*” consiste en describir las funciones y características de la aplicación *web*, así como las restricciones que dirigen su desarrollo y la información (datos y flujos de control) que entra y sale del sistema. Esta descripción es escrita a través de modelos gráficos, modelos matemáticos formales, una colección de escenarios de uso, un prototipo o una combinación de todas las anteriores.

Esta descripciones constituyen el principal elemento del documento de especificación de requerimientos, el cual contiene almacenados los requerimientos de manera formal, como por ejemplo, los modelos que forman parte del Lenguaje de Modelado Unificado (UML).

El mapa está conformado por los siguientes elementos:

- **5 Intenciones**, las cuales representan el objetivo que se desea satisfacer en el proceso. Estas son: *Comienzo*, *capturar requerimientos*, *definir requerimientos*, *especificar requerimientos* y *parada*.
- **4 Directivas de Selección de Estrategia (DSE)**, las cuales permiten seleccionar una estrategia para ejecutar una intención de modelado, a partir de una intención de partida. Estas son:
 - a) **DSE16:** De la cual se puede seleccionar la estrategia “*kAT*”, la estrategia “*JAD*”, la o la estrategia de “pruebas de usabilidad”, con el fin de “*capturar requerimientos*”.
 - b) **DSE17:** A partir de esta directiva se puede seleccionar la estrategia “orientada a los casos de uso”, la estrategia “orientada a los escenarios”, o las dos anteriores, con el fin de partir de la intención de “*capturar requerimientos*” hacia la intención de “*definir requerimientos*”.

- c) **DSE18:** A partir de esta directiva se puede seleccionar la estrategia del “modelo dinámico”, la estrategia del “modelo funcional”, la estrategia del “modelo estructural”, o todas las anteriores, con el fin de partir de la intención de “*definir requerimientos*” hacia la intención de “*especificar requerimientos*”.
- d) **DSE19:** A partir de esta directiva se puede seleccionar la estrategia “orientada por la escalabilidad”, la estrategia de “diseño con Reutilización”, o las dos anteriores, con el fin de partir de la intención de “*especificar requerimientos*” hacia la intención de “*parada*”.
- **4 Directivas de Selección de Intención (DSI)**, las cuales permiten seleccionar una intención destino entre las disponibles a partir de una intención de partida. Las mismas están identificadas como DSI12, DSI13, DSI14 y DSI15.
- **7 Directivas de Ejecución de Intención (DEI)**, las cuales guían la ejecución de una intención según una estrategia específica. Estas se listan en la tabla No.15 donde se indica, su descripción, situación inicial y argumentos para la ejecución de las mismas.

Tabla No. 15. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de *Definir y Especificar Requerimientos* del Método Situacional para el Desarrollo de Aplicaciones Web (a)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI45. Estrategia de documentación	Elaboración de los documentos de definición y especificación de requerimientos	Requerimientos de la aplicación web capturados, definidos y especificados.	(A45): La organización tiene cierto grado de madurez y/o no se está desarrollando la aplicación web bajo un enfoque ágil y/o se dispone de una herramienta que facilite la especificación de requerimientos.
DEI46. Estrategia de validación y verificación	Verificación de la captura, definición y especificación de requerimientos se encuentran libres de errores u omisiones.	Requerimientos de la aplicación web capturados definidos y especificados.	(A46): Se capturaron definieron y especificaron los requerimientos de la aplicación web y/o el cliente exige validar la definición de requerimientos y/o el grupo de desarrollo requiere garantizar la calidad de la aplicación web.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Tabla No. 15. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de Definir y Especificar Requerimientos del Método Situacional para el Desarrollo de Aplicaciones Web (b)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI47. Estrategia orientada a los escenarios	Descripción informal de las características de la aplicación web desde el punto de vista del actor.	Características de la aplicación web.	(A47): La definición de requerimientos se centra en las funcionalidades que deberán implementarse en la aplicación web y/o se desea buscar una forma de complementar el proceso de definición de requerimientos y/o se requiere lograr un adecuado entendimiento entre los usuarios y clientes.
DEI48. Estrategia orientada a los casos de uso	Definición de los límites de la aplicación web y las relaciones entre la aplicación web y el entorno.	Características de la aplicación web.	(A48): Se representarán los requerimientos desde el punto de vista del usuario y/o el grupo de desarrollo tiene experiencia en la definición de casos de uso y/o se requiere definir claramente los roles de cada uno de los actores.
DEI49. Estrategia del modelo dinámico	Especificación del comportamiento de la aplicación web y de los pases de mensajes e información entre objetos, interacciones con los usuarios y de los distintos estados por los que pasan los objetos dinámicos.	Documento de definición de requerimientos	(A49): Se requiere representar el comportamiento de la aplicación web en términos de estados y eventos y/o se requiere especificar el intercambio de mensajes entre objetos y/o se requiere representar el comportamiento interno de un método y/o el grupo de desarrollo tiene experiencia en UML y/o se dispone de herramientas que faciliten el modelado de la aplicación web.
DEI50. Estrategia del modelo funcional	Especificación del comportamiento externo de la aplicación web.	Documento de definición de requerimientos	(A50): Se requiere representar gráficamente las necesidades funcionales de cada actor y/o se requiere describir la interacción entre los actores y la aplicación web bajo la forma de un flujo de eventos y/o el grupo de desarrollo tiene experiencia en UML y/o se dispone de herramientas que faciliten el proceso de modelado.
DEI51. Estrategia del modelo estructural	Especificación de la estructura estática de la aplicación web, en términos de clases y de relaciones entre las mismas.	Documento de definición de requerimientos.	(A51): El grupo de desarrollo implementará la aplicación web a través del uso de lenguajes orientados a objetos y/o el grupo de desarrollo tiene experiencia en UML y/o se disponen de herramientas que faciliten el proceso de modelado y la generación de código fuente.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

3.4.2. DISEÑAR LA APLICACIÓN WEB

Diseñar la aplicación *web* consiste en describir todos los aspectos de la aplicación *web* que se van a implementar. Este mapa contiene seis (6) intenciones fundamentales: “*diseñar la arquitectura de ejecución*”, “*reutilizar los patrones de diseño*”, “*diseñar la arquitectura lógica*”, “*diseñar la capa de presentación*”, “*diseñar la capa de lógica del negocio*” y “*diseñar la capa de datos*” (ver figura 42).

La intención de “*diseñar la arquitectura de ejecución*” consiste en localizar en los recursos físicos los componentes en tiempo de ejecución. El producto que se genera luego de ejecutada esta intención forma parte del documento de la arquitectura del sistema, ya que el mismo contiene el diseño de la arquitectura de ejecución y el diseño de la arquitectura lógica.

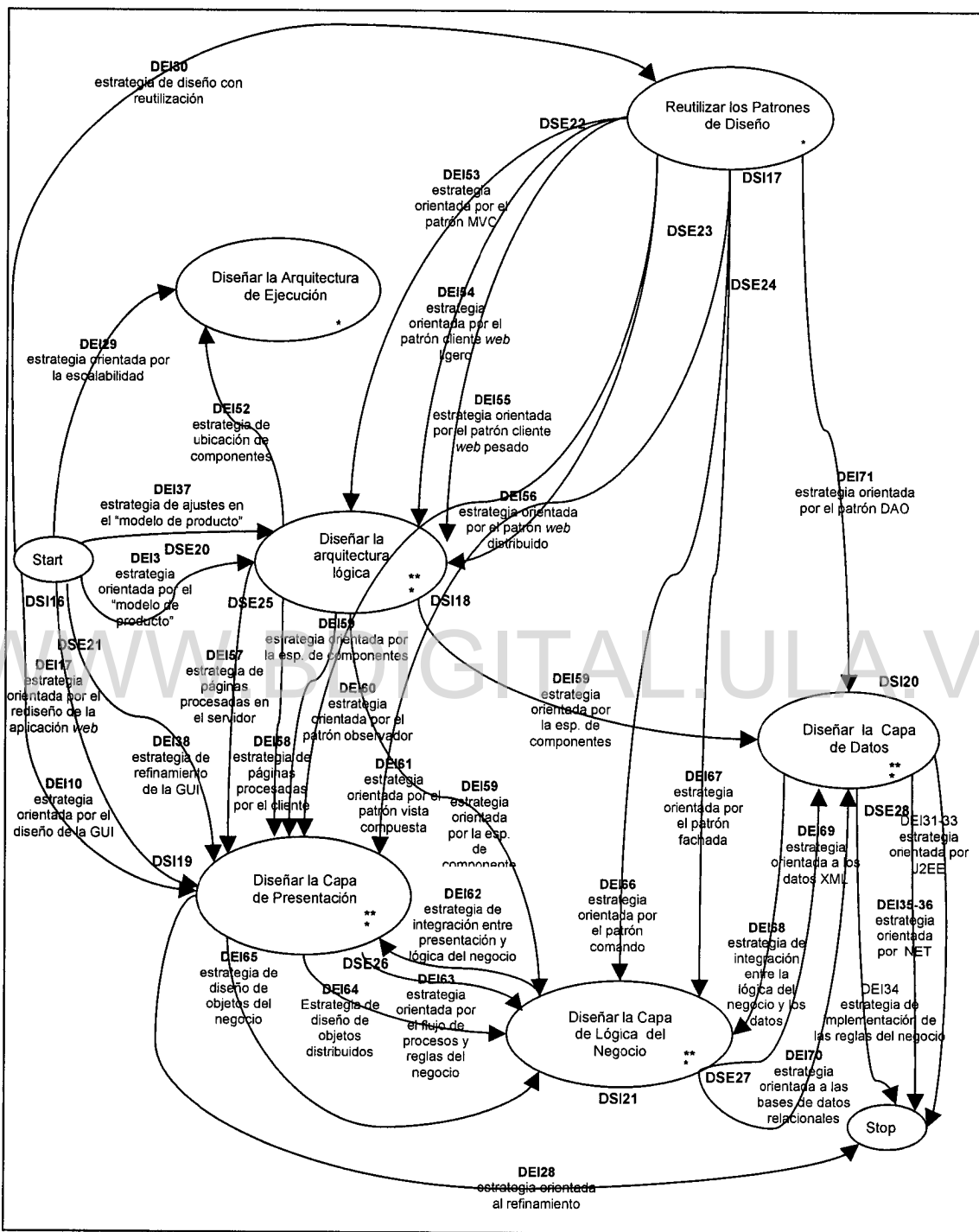
La intención de “*reutilizar los patrones de diseño*” permite crear la arquitectura de diseño integrando componentes reutilizables, con el fin de reutilizar diseños exitosos basando los nuevos diseños en experiencias previas. El producto que se genera luego de la ejecución de esta intención es el patrón implementado en cualquier otra intención de diseño.

La intención de “*diseñar la arquitectura lógica*” consiste en definir la arquitectura a través de un conjunto de componentes de software, conectores y restricciones que están organizados en capas. El resultado de ejecutar esta intención forma parte del documento de diseño de la arquitectura, ya que, como se dijo anteriormente, el mismo está conformado por el diseño de la arquitectura de ejecución y el diseño de la arquitectura lógica.

La intención de “*diseñar la capa de presentación*” consiste en definir los aspectos relacionados con el diseño *web*, navegación, presentación y estilo. Así como también, en especificar los componentes que conformarán dicha capa. Los productos que se generan luego de ejecutada esta intención son: el documento de diseño de la interfaz de usuario y el documento de los componentes de la capa de presentación, el cual es parte integrante del documento de componentes.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)



* DEI72 Documentación
 ** DEI73 Validación y Verificación
 DEI131-33 Todas las estrategias utilizadas en el Start de J2EE
 DEI135-36 Todas las estrategias utilizadas en el Start de .NET

Figura 42. Mapa Local de Bajo Nivel de la Intención de Diseñar la Aplicación Web del Método Situacional para el Desarrollo de Aplicaciones Web

El documento de diseño de la interfaz de usuario define la estructura de la Interfaz Gráfica de Usuario (GUI), el flujo de navegación, el estilo de las páginas *web* correspondientes a la capa de presentación de la aplicación *web*; y el documento de componentes representa el diseño de cada componente de cada una de las capas de la aplicación *web* (presentación, lógica del negocio y datos) en forma de notación gráfica en un nivel de abstracción muy cercano al código fuente.

La intención de “*diseñar la capa de lógica del negocio*” consiste en definir los aspectos relacionados con el diseño de los objetos del negocio, el flujo de procesos y reglas del negocio. Así como también, en especificar los componentes que conformarán dicha capa y la forma como se integrarán los componentes de la capa de presentación con los componentes de la capa del negocio. Los productos que se generan son el documento de diseño de los objetos del negocio y el documento de componentes de la capa de lógica del negocio, el cual es parte integrante del documento de diseño de componentes de la aplicación *web*.

El documento de los objetos del negocio representa el diseño de la lógica del negocio, a través de la descripción de los objetos del negocio, flujo de procesos y reglas del negocio que serán implementadas en la aplicación *web*.

La intención de “*diseñar la capa de datos*” consiste en definir la estructura y el contenido de las bases de datos asociadas a los componentes. Así como también se realiza la especificación de los componentes de la capa de datos y un mapeo entre éstos y las entidades en las bases de datos. Además se definen los mecanismos de integración entre los componentes de la capa de lógica del negocio y los componentes de la capa de datos.

Los productos que se generan al ejecutar esta intención son: el documento de diseño de la base de datos y el documento de componentes de la capa de datos, el cual es parte integrante del documento de diseño de componentes de la aplicación *web*. El documento de diseño de la base de datos contiene el diseño conceptual de la base de datos y la relación entre dicho diseño con los componentes especificados en esta capa.

Este está conformado por los siguientes elementos:

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

- **8 Intenciones**, las cuales representan el objetivo que se desea satisfacer en el proceso. Estas son: *Comienzo, diseñar la arquitectura de ejecución, reutilizar los patrones de diseño, diseñar la arquitectura lógica, diseñar la capa de presentación, diseñar la capa de lógica del negocio, diseñar la capa de datos y parada.*
- **9 Directivas de Selección de Estrategia (DSE)**, las cuales permiten seleccionar una estrategia para ejecutar una intención de modelado, a partir de una intención de partida. Estas son:
 - a) **DSE20:** De la cual se puede seleccionar la estrategia “orientada por el modelo de producto”, o la estrategia de “ajustes en el modelo de producto”, con el fin de diseñar la “*arquitectura lógica*”.
 - b) **DSE21:** A partir de esta directiva se puede seleccionar la estrategia “orientada por el diseño de la GUI”, la estrategia “orientada por el rediseño de la aplicación web” o la estrategia de “refinamiento de la GUI”, con el fin de “*diseñar la capa de presentación*”.
 - c) **DSE22:** A partir de esta directiva se puede seleccionar la estrategia “orientada por el patrón MVC”, la estrategia “orientada por el patrón cliente web ligero”, la estrategia “orientada por el patrón cliente web pesado”, o la estrategia “orientada por el patrón web distribuido”, con el fin de partir de la intención de “*reutilizar patrones de diseño*” hacia la intención de “*diseñar la arquitectura lógica*”.
 - d) **DSE23:** A partir de esta directiva se puede seleccionar la estrategia “orientada por el patrón observador”, o la estrategia “orientada por el patrón vista compuesta”, con el fin de partir de la intención de “*reutilizar los patrones de diseño*” hacia la intención de “*diseñar la capa de presentación*”.
 - e) **DSE24:** A partir de esta directiva se puede seleccionar la estrategia “orientada por el patrón comando”, o la estrategia “orientada por el patrón fachada”, con el fin de partir de la intención de “*reutilizar patrones de diseño*” hacia la intención de “*diseñar la capa de lógica del negocio*”.

Licencia Creative Commons:

- f) **DSE25:** A partir de esta directiva se puede seleccionar la estrategia “páginas procesadas en el servidor”, la estrategia de “páginas procesadas por el cliente”, o la estrategia “orientada por la especificación de componentes”, con el fin de partir de la intención de “diseñar la arquitectura lógica” hacia la intención de “*diseñar la capa de presentación*”.
- g) **DSE26:** A partir de esta directiva se puede seleccionar la estrategia de “diseño de objetos del negocio”, la estrategia de “diseño de objetos distribuidos”, la estrategia “orientada por el flujo de procesos y reglas del negocio”, o la combinación de todas las anteriores, con el fin de partir de la intención de “*diseñar la capa de presentación*” hacia la intención de “*diseñar la capa de lógica del negocio*”.
- h) **DSE27:** A partir de esta directiva se puede seleccionar la estrategia “orientada a los datos XML”, o la estrategia “orientada a las bases de datos relacionales”, con el fin de partir de la intención de “*diseñar la capa de lógica del negocio*” hacia la intención de “*diseñar la capa de datos*”.
- i) **DSE28:** A partir de esta directiva se puede seleccionar la estrategia “orientada por *J2EE*”, (que a su vez comprende tres posibles estrategias: estrategia “orientada por los *servlets*”, estrategia de “*applets*” o estrategia “orientada por *JSP*”), la estrategia de “implementación de las reglas del negocio” o la estrategia “orientada por *.NET*”, (que a su vez comprende dos posibles estrategias: estrategia “orientada por el *ASP.NET*” o estrategia “orientada por el *MMIT*”), con el fin de partir de la intención de “*diseñar la capa de datos*” hacia la intención de “*parada*”.
- **6 Directivas de Selección de Intención (DSI)**, las cuales permiten seleccionar una intención destino entre las disponibles a partir de una intención de partida. Las mismas están identificadas como DSI16, DSI17, DSI18, DSI19, DS20 y DS21.

Licencia Creative Commons:

- **22 Directivas de Ejecución de Intención (DEI)**, las cuales guían la ejecución de una intención según una estrategia específica. Estas se listan en la tabla No.16 donde se indica, su descripción, situación inicial y argumentos para la ejecución de las mas.

Tabla No. 16. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de *Diseñar la Aplicación Web* del Método Situacional para el Desarrollo de Aplicaciones *Web* (a)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI52. Estrategia de ubicación de componentes	Selección del <i>hardware</i> y <i>software</i> necesario donde se implantará la aplicación <i>web</i> , ubicación de los componentes de cada capa de la aplicación <i>web</i> en el <i>hardware</i> donde será implantado.	Documento de la arquitectura lógica de la aplicación <i>web</i> y documento de definición y especificación de requerimientos.	(A52): Se dispone del documento de la arquitectura lógica y/o se cuenta con la tecnología necesaria para implantar la aplicación <i>web</i> y/o el grupo de desarrollo tiene experiencia en el mapeo de componentes/ <i>hardware</i> .
DEI53. Estrategia orientada por el patrón MVC.	Especificación de los componentes que interactúan en cada una de las capas de la aplicación <i>web</i> tomando como base el patrón Modelo-Vista-Controlador y especificación de las interfaces de dichos componentes.	Patrón Modelo-Vista-Controlador y documento de especificación de requerimientos de la aplicación <i>web</i> .	(A53): La aplicación <i>web</i> tiene un alto grado de interactividad con el usuario y múltiples interfaces gráficas son requeridas (vistas) para cada requerimiento particular de los usuarios y/o el grupo de desarrollo tiene experiencia en la Reutilización de patrones de diseño.
DEI54. Estrategia orientada por el patrón cliente <i>web</i> ligero.	Especificación de los componentes que interactúan en cada una de las capas de la aplicación <i>web</i> tomando como base el patrón cliente <i>web</i> ligero y especificación de las interfaces de dichos componentes.	Patrón cliente <i>web</i> ligero y documento de especificación de requerimientos de la aplicación <i>web</i> .	(A54): La aplicación <i>web</i> se implantará en Internet y se requiere un mínimo control sobre la configuración del cliente y la versión del navegador y la validación de los datos se efectúa en el servidor <i>web</i> y/o se desea desarrollar una aplicación <i>web</i> de comercio electrónico.
DEI55. Estrategia orientada por el patrón cliente <i>web</i> pesado.	Especificación de los componentes que interactúan en cada una de las capas de la aplicación <i>web</i> tomando como base el patrón cliente <i>web</i> pesado y especificación de las interfaces de dichos componentes.	Patrón cliente <i>web</i> pesado y documento de especificación de requerimientos de la aplicación <i>web</i>	(A55): Se requiere una interfaz de usuario sofisticada y/o los datos pueden ser validados en el cliente y/o se debe asumir cierto control sobre la configuración del cliente y la versión del navegador y/o el grupo de desarrollo tiene experiencia en la Reutilización de patrones de diseño.
DEI56. Estrategia orientada por el patrón <i>web</i> distribuido.	Especificación de los componentes que interactúan en cada una de las capas de la aplicación <i>web</i> tomando como base el patrón <i>web</i> distribuido, y especificación de las interfaces de dichos componentes.	Patrón <i>web</i> distribuido y documento de especificación de requerimientos de la aplicación <i>web</i> .	(A56): La aplicación <i>web</i> se implantará en una Intranet y debe existir un adecuado control sobre la configuración del cliente y de la red y/o se desean reutilizar varios patrones de diseño de arquitectura lógica y/o el grupo de desarrollo tiene experiencia en la Reutilización de patrones de diseño.

Tabla No. 16. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de *Diseñar la Aplicación Web* del Método Situacional para el Desarrollo de Aplicaciones Web (b)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI57. Estrategia orientada por las páginas procesadas en el servidor.	Diseño de la interfaz gráfica de usuario y especificación de las reglas de validación de los datos que serán introducidos por el usuario.	Documento de especificación de requerimientos o patrón del cliente web ligero.	(A57): Se reutilizó el patrón del cliente web ligero y/o se está desarrollando una aplicación de comercio electrónico y/o el cliente solo requiere un navegador para realizar peticiones a la página web.
DEI58. Estrategia de páginas procesadas en el cliente.	Diseño de la interfaz gráfica de usuario y especificación de las reglas de validación de los datos que serán introducidos por el usuario.	Documento de especificación de requerimientos o patrón del cliente web pesado o patrón del cliente web distribuido.	(A58): Se reutilizó el patrón del cliente web pesado y/o se requiere asumir cierto control sobre la configuración del cliente y la versión del navegador y/o se requiere una interfaz de usuario sofisticada que incluya la validación de datos del lado del cliente.
DEI59. Estrategia de especificación de componentes.	Elaboración del modelo de componentes de cada una de las capas de la aplicación web, especificación del contrato de uso indicando lista de operaciones de las interfaces, parámetros de entrada y salida, restricciones aplicables y especificación de pre y post condiciones	Modelo de producto de la aplicación web y patrones reutilizados en el diseño de la arquitectura lógica.	(A59): Se cuenta con el diseño de la arquitectura lógica de la aplicación web y/o el grupo de desarrollo tiene experiencia en el diseño basado en componentes y/o se dispone de herramientas que soporten el diseño basado en componentes.
DEI60 Estrategia orientada por el patrón observador.	Especificación de la función de cada componente de la capa de presentación y la integración entre ellos.	Patrón observador y especificación de componentes de la capa de presentación.	(A60): Se reutilizó el patrón MVC en el diseño de la arquitectura lógica y se requiere disminuir el acoplamiento entre las vistas y el controlador y la aplicación web tiene un alto grado de interacción y/o es necesario facilitar la Reutilización de componentes de la capa de presentación.
DEI61 estrategia del patrón vista compuesta.	Diseño del estilo visual de las páginas basadas en la creación de vistas compuestas, y diseño de cada vista atómica que será incluida en la vista compuesta.	Documento de definición de requerimientos de la aplicación web y patrón de vista compuesta.	(A61): Se requiere manipular la distribución de la página independientemente del contenido y se requiere reutilizar las vistas atómicas y se requiere generar páginas web que muestren componentes que puedan combinarse de distintas maneras en una sola página y/o se requiere hacer un prototipo de la distribución de la aplicación web
DEI62. Estrategia de integración entre las capas de presentación y lógica del negocio.	Especificación de las interfaces de los componentes y la forma como van a interactuar la capa de presentación y lógica del negocio.	Componentes de la capa de presentación y componentes de la capa de lógica del negocio.	(A62): Se tienen especificados los componentes de la capa de presentación y lógica del negocio y se conocen muy bien las reglas del negocio y se tiene bien claro el grado de dependencia que se debe mantener entre las capas.

Licencia Creative Commons

Tabla No. 16. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de *Diseñar la Aplicación Web* del Método Situacional para el Desarrollo de Aplicaciones Web (c)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI63. Estrategia orientada por los flujos de procesos y reglas del negocio.	Elaboración de los diagramas de secuencia y colaboración que muestren el detalle de los flujos de procesos en la aplicación web y Detalles de las reglas del negocio.	Modelo de procesos del negocio y/o documento de especificación de requerimientos de la aplicación web.	(A63): Se cuenta con la documentación completa del modelo de negocios y/o se dispone del documento de especificación de requerimientos y/o se dispone de herramientas que soporten el diseño orientado por flujos de procesos y reglas del negocio.
DEI64. Estrategia del diseño de objetos distribuidos.	Identificación de los objetos distribuidos y sus protocolos de comunicación y separación de los objetos del negocio y los objetos distribuidos.	Objetos distribuidos, componentes de la capa de lógica del negocio.	(A64): Se reutilizó el patrón web distribuido en el diseño de la arquitectura lógica de la aplicación web y/o la aplicación web se implantará en una Intranet y/o el grupo de desarrollo tiene conocimientos sólidos en las características y comportamiento de los objetos distribuidos en una aplicación web y/o se dispone de herramientas que soporten esta estrategia de diseño.
DEI65. Estrategia del diseño de objetos del negocio.	Identificación de los objetos que conformarán la capa de lógica del negocio y separación de los objetos del negocio y los objetos de la aplicación web.	Modelo de procesos del negocio orientado a objetos y documento de especificación de requerimientos de la aplicación web.	(A65): Se cuenta con la documentación del modelo de negocio y/o se dispone de herramientas que soportan el diseño de objetos del negocio y/o la aplicación web se implementará con el apoyo de un lenguaje orientado a objetos.
DEI66. Estrategia del patrón comando.	Agrupamiento los flujos de procesos que se implementarán en la aplicación web como un requerimiento funcional.	Patrón comando, modelo del negocio y documentos de especificación de requerimientos de la aplicación web.	(A66): Se reutilizó el patrón MVC en el diseño de la arquitectura lógica de la aplicación web y/o se requiere una aplicación web con alto grado de interactividad y/o es requerido ejecutar el comando "Deshacer" en la aplicación web y/o es necesario almacenar los requerimientos del usuario sin conocer su contenido.
DEI67. Estrategia del patrón fachada.	Especificación de cada uno de los componentes de la capa de lógica del negocio y creación de interfaces sencillas para aquellos componentes complejos o difíciles de acceder.	Patrón fachada, documento de especificación de requerimientos de la aplicación web.	(A67): Se reutilizo el patrón MVC y/o se reutilizó el patrón del cliente web pesado y/o se reutilizó el patrón web distribuido en el diseño de la arquitectura lógica de la aplicación web y/o se requiere separar los objetos del negocio de los objetos de la aplicación y/o se requieren diseñar componentes reutilizables.
DEI68. Estrategia de integración entre las capas de lógica del negocio y datos.	Especificación de las interfaces de los componentes de las capas de lógica del negocio y datos y la forma como van a interactuar ambas capas.	Componentes de la capa de lógica del negocio y los componentes de la capa de datos.	(A68): Se tienen especificados los componentes de las capas de lógica del negocio y datos y se conocen muy bien los controles de acceso a la capa de datos y se tiene definido el grado de dependencia que debe mantenerse entre ambas capas.

Tabla No. 16. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de *Diseñar la Aplicación Web* del Método Situacional para el Desarrollo de Aplicaciones Web (d)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI69. Estrategia orientada a los datos XML.	Definición de la estructura y el contenido de los componentes de datos, definición de la categoría de almacenamiento de datos XML (documento o datos) y de la estrategia de almacenamiento de los datos (BLOB, archivos, mapeo de la estructura de datos en repositorio de bases de datos)	Componentes de la capa de datos, diseño de la capa de lógica del negocio.	(A69): Se requiere transmitir documentos en la aplicación web y/o la aplicación web es del tipo B2B y/o se requiere construir formatos de mensajes para integrar la aplicación web con sistemas legados y/o la aplicación web manipulará varios catálogos de productos y/o el grupo de desarrollo tiene experiencia en la gestión de datos XML y se cuenta con herramientas que soporten datos XML.
DEI70. Estrategia orientada por las bases de datos relacionales.	Definición de la estructura y el contenido de los componentes de datos, definición de cada una de las tablas y sus relaciones y de la estrategia de manipulación y acceso a los datos.	Componentes de la capa de datos, diseño de la capa de lógica del negocio.	(A70): Se requiere administrar grandes volúmenes de datos y/o no se dispone de herramientas que ofrezcan capacidades de almacenamiento de datos XML y/o se requiere utilizar todas las potencialidades proporcionadas por SQL y se requiere un buen rendimiento en la transmisión y procesamiento de los datos y/o el grupo de desarrollo no cuenta con experiencia para gestionar datos XML y/o no es necesario integrar la aplicación web con sistemas legados.
DEI71. Estrategia del patrón DAO.	Identificación de las fuentes de datos a utilizar en la aplicación web y sus métodos de acceso.	Diseño de la capa de lógica del negocio, patrón DAO.	(A71): Se requiere el uso de distintas fuentes de datos y/o se implementarán bases de datos relacionales y/o se requiere integrar varias aplicaciones web y/o el grupo de desarrollo tiene experiencia en encapsular los accesos a las fuentes de datos a través de la Reutilización de patrones de diseño.
DEI72. Estrategia de documentación.	Elaboración del documento donde se especifica la arquitectura de ejecución, la arquitectura lógica, el diseño de la interfaz de usuario, el diseño de los componentes de cada capa de la aplicación web, el diseño de los objetos del negocio y el diseño de las bases de datos.	Diseño de la arquitectura de ejecución, diseño de la arquitectura lógica, diseño de cada una de las capas de la aplicación web.	(A72): Se realizó el diseño de la arquitectura de ejecución y/o se realizó el diseño de la arquitectura lógica y/o se realizó del diseño de cada una de las capas de la aplicación web y/o la organización tiene cierto grado de madurez.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Tabla No. 16. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de *Diseñar la Aplicación Web* del Método Situacional para el Desarrollo de Aplicaciones *Web* (e)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI73. Estrategia de validación y verificación	Comprobación de que los patrones de diseño se adaptan a la arquitectura lógica de la aplicación <i>web</i> o a cada una de sus respectivas capas: presentación, lógica del negocio y datos y comprobación de que la arquitectura lógica y cada una de las capas satisfacen los requerimientos funcionales y no funcionales de la aplicación <i>web</i> .	Patrones de diseño, diseño de la arquitectura lógica, diseño de la capa de presentación, diseño de la capa de lógica del negocio y diseño de la capa de datos.	(A73): Se selecciona un patrón de diseño para ser reutilizado y el grupo de desarrollo no está seguro de que el patrón seleccionado resuelve el problema de diseño y/o el grupo de desarrollo tiene experiencia en el reuso de patrones de diseño y/o el grupo de desarrollo tiene experiencia en la identificación de componentes de cada una de las capas de la aplicación <i>web</i> .

3.4.3. IMPLEMENTAR LA APLICACIÓN WEB.

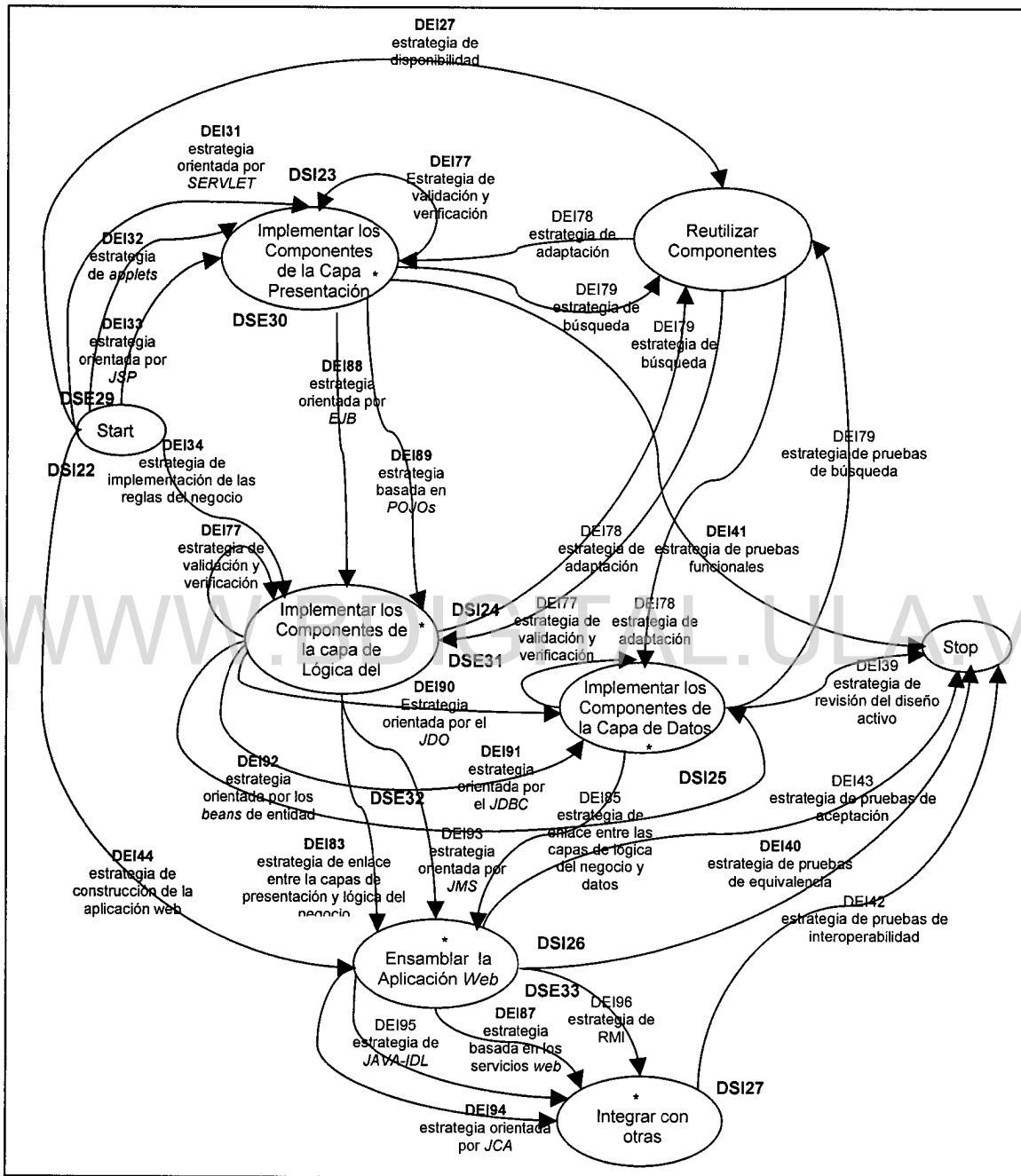
Implementar la aplicación *web* tiene como finalidad la implementación del diseño producido anteriormente, así como también, el enlace de los componentes de la capa de presentación con los componentes de la capa de lógica del negocio y el de los componentes de la capa de lógica del negocio con los componentes de la capa de datos, utilizando el “*middleware*” definido. Además, permite integrar la aplicación *web* generada con cualquier otra aplicación existente. Este mapa contiene seis (6) intenciones fundamentales: “*reutilizar componentes*”, “*implementar los componentes de la capa de presentación*”, “*implementar los componentes de la capa de lógica del negocio*”, “*implementar los componentes de la capa de datos*”, “*ensamblar la aplicación web*” e “*integrar con otras aplicaciones*”.

Es importante señalar, que por razones de claridad en la lectura este mapa se ha dividido en dos nuevos mapas de acuerdo al tipo de *middleware* a utilizar en la implementación de la aplicación *web*: “*implementar la aplicación web con .NET*” e “*implementar la aplicación web con J2EE*” (ver figuras 43 y 44).

También cabe resaltar que esta intención constituye la parte del modelo de proceso del método situacional para el desarrollo de aplicaciones *web* más vulnerable al cambio, debido a que depende de la tecnología que esté disponible en el mercado.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)



* DEI97 Documentación

Figura 44. Mapa Local de Bajo Nivel de la Intención de *Implementar la Aplicación Web (J2EE)* del Método Situacional para el Desarrollo de Aplicaciones Web

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Es importante aclarar que las intenciones en ambos mapas son exactamente las mismas, la diferencia se presenta cuando las estrategias se refieren específicamente al *middleware* seleccionado. De resto todos los demás aspectos son comunes entre ambos mapas. A continuación, se describen primero las intenciones fundamentales de implementar la aplicación *web* y luego se detalla cada mapa en particular.

La intención de “*reutilizar componentes*” consiste en buscar, adquirir, suscribir y adaptar componentes ya desarrollados y probados anteriormente, en la implementación de cada una de las capas que constituyen la aplicación *web*. El principal producto que se genera, luego de ejecutar esta intención, es la documentación del componente que ha sido adaptado en la aplicación *web*, indicando, detalladamente, las interfaces utilizadas en la misma.

La intención de “*implementar los componentes de la capa de presentación*” tiene como finalidad la implementación de los componentes ya diseñados y especificados anteriormente. El producto que se obtiene luego de la ejecución de esta intención lo constituyen los componentes de la capa de presentación ya implementados utilizando un *middleware* específico.

La intención de “*implementar los componentes de la capa de lógica del negocio*” tiene como finalidad la implementación de los componentes ya diseñados y especificados anteriormente. El producto que se obtiene luego de la ejecución de esta intención lo constituyen los componentes de la capa de lógica del negocio ya implementados utilizando un *middleware* específico.

La intención de “*implementar los componentes de la capa de datos*” tiene como finalidad la implementación de los componentes ya diseñados y especificados anteriormente. El producto que se obtiene luego de la ejecución de esta intención lo constituyen los componentes de la capa de datos ya implementados utilizando un *middleware* específico.

La intención de “*ensamblar la aplicación web*” consiste en enlazar los componentes de la capa de presentación con los componentes de la capa de lógica del negocio, y los componentes de la capa de lógica del negocio con los de la capa de datos. Esto se realiza a través de las interfaces diseñadas para ensamblar la

aplicación *web*. El producto que se genera al culminar la ejecución de esta intención está constituido por la aplicación *web* ensamblada y los documentos de la aplicación *web* indicados en el plan del proyecto.

La intención de “*integrar con otras aplicaciones*” consiste en utilizar el *middleware* adecuado para que dos aplicaciones diferentes puedan interoperar aún estando implantadas en ambientes de ejecución diferentes. Cabe destacar, que esta intención solo se ejecuta en el caso de que existan otras aplicaciones (portales, *e-commerce*, *ERP*, etc.) implementadas con las cuales se requiere cierto grado de integración para lograr satisfacer un requerimiento particular. El producto generado es la aplicación *web* integrada con las aplicaciones ya existentes y la documentación de este resultado.

El mapa de implementar la *aplicación web con .NET*, y el mapa de *implementar la aplicación con J2EE* están conformados por los siguientes elementos comunes:

- **3 Intenciones**, las cuales representan el objetivo que se desea satisfacer en el proceso. Estas son: *Comienzo, reutilizar componentes, implementar los componentes de la capa de presentación, implementar los componentes de la capa de lógica del negocio, implementar los componentes de la capa de datos, ensamblar la aplicación web, integrar con otras aplicaciones y parada.*
- **5 Directivas de Selección de Estrategia (DSE)**, las cuales permiten seleccionar una estrategia para ejecutar una intención de modelado, a partir de una intención de partida. Estas son:
 - a) **DSE29:** De la cual se puede seleccionar la estrategia “orientada por el *ASP.NET*”, la estrategia “orientada por el *MMIT (.NET)*”, la estrategia “orientada por los *SERVLETS*”, la estrategia de “*applets*”, o la estrategia “orientada por *JSP*” (*J2EE*), con el fin de “*implementar los componentes de la capa de presentación*”.
 - b) **DSE30:** De la cual se puede seleccionar la estrategia “orientada por el *.NET Enterprise Service*”, la estrategia “orientada por el *Biztalk Server (.NET)*”, la estrategia “orientada por *EJB*” o la estrategia

“basada en *POJOs*” (*J2EE*), con el fin de partir de la intención de “implementar los componentes de la capa de presentación” hacia la intención de “*implementar los componentes de la capa de lógica del negocio*”.

c) **DSE31:** A partir de esta directiva se puede seleccionar la estrategia “orientada por el *OLDB*”, la estrategia “orientada al *XML*”, la estrategia “orientada al *ADO.NET*” (*.NET*), la estrategia “orientada por el *JDO*”, la estrategia “orientada por el *JDBC*” o la estrategia “orientada por los *beans* de entidad” (*J2EE*), con el fin de partir de la intención de “implementar los componentes de la capa de lógica del negocio” hacia la intención de “*implementar los componentes de la capa de datos*”.

d) **DSE32:** A partir de esta directiva se puede seleccionar la estrategia de “enlace entre las capas de presentación y lógica del negocio” (*.NET* y *J2EE*), la estrategia “orientada por *MSMQ*” (*.NET*), o la estrategia “orientada por *JMS*” (*J2EE*), con el fin de partir de la intención de “*implementar los componentes de la capa de lógica del negocio*” hacia la intención de “*ensamblar la aplicación web*”.

e) **DSE33:** A partir de esta directiva se puede seleccionar la estrategia “orientada por el *ja.NET*” (*.NET*), la estrategia “basada en los servicios *web*” (*.NET* y *J2EE*), la estrategia “orientada por *JCA*”, la estrategia de “*JAVA-IDL*” o la estrategia “*RMI-IIOP*” (*J2EE*), con el fin de partir de la intención de “*ensamblar la aplicación web*” hacia la intención de “*integrar con otras aplicaciones*”.

➤ **6 Directivas de Selección de Intención (DSI)**, las cuales permiten seleccionar una intención destino entre las disponibles a partir de una intención de partida. Las mismas están identificadas como DSI22, DSI23, DSI24, DSI25, DSI26 y DSI27.

➤ **7 Directivas de Ejecución de Intención (DEI) (comunes en ambos mapas)**, las cuales guían la ejecución de una intención según una estrategia específica. Estas se listan en la tabla No.17 donde se indica,

su descripción, situación inicial y argumentos para la ejecución de las mismas.

Tabla No. 17. Directivas de Ejecución de Intención (DEI) Comunes de los Mapas Locales de Bajo Nivel de la Intención de *Implementar la Aplicación Web con .NET y J2EE* del Método Situacional para el Desarrollo de Aplicaciones *Web* (a)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI77. Estrategia de validación y verificación	Comprobación de que el código fuente de cada uno de los componentes de las capas de presentación, lógica del negocio y datos es consistente con el diseño de la aplicación web.	Componentes de las capas de presentación, lógica del negocio y datos implementados.	(A77):El grupo de desarrollo se preocupa por asegurar la calidad de los componentes implementados y/o la organización tiene cierto grado de madurez y/o el grupo de desarrollo requiere comprobar que los componentes implementados se corresponde con el diseño realizado anteriormente.
DEI78. Estrategia de adaptación	Creación de interfaces que permitan adaptar los componentes reutilizados en la aplicación web	Componentes reutilizados.	(A78):El grupo de desarrollo requiere aumentar la productividad en la implementación de la aplicación web y/o el grupo de desarrollo tiene experiencia en las técnicas de adaptación de componentes y/o los componentes encontrados están probados y certificados.
DEI79. Estrategia de búsqueda	Localización de componentes de las capas de presentación, lógica del negocio y datos en librerías de componentes reutilizables.	Librerías de componentes reutilizables.	(A79):El grupo de desarrollo requiere aumentar la productividad en la implementación de la aplicación web y/o el grupo de desarrollo dispone de librerías de componentes reutilizables y/o es necesario disponer rápidamente de la aplicación web.
DEI83. Estrategia de enlace entre las capas de presentación y lógica del negocio	Implementación de las interfaces de los componentes de la capa de presentación y lógica del negocio.	Componentes de la capa de presentación y componentes de la capa de lógica del negocio.	(A83):Se tienen especificados los componentes de la capa de presentación y lógica del negocio y se dispone de la especificación de las interfaces de los componentes de las capas de presentación y lógica del negocio y/o se tiene bien claro el grado de acoplamiento que se debe mantener entre las capas.
DEI85. Estrategia de enlace entre las capas de lógica del negocio y datos.	Implementación de las interfaces de los componentes de la capa de lógica del negocio y datos.	Componentes de la capa de lógica del negocio y datos.	(A85):Se tienen especificados los componentes de lógica del negocio y datos y se dispone de la especificación de las interfaces de los componentes de las capas de lógica del negocio y datos y/o se tiene bien claro el grado de acoplamiento que se debe mantener entre las capas.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Tabla No. 17. Directivas de Ejecución de Intención (DEI) Comunes de los Mapas Locales de Bajo Nivel de la Intención de *Implementar la Aplicación Web con .NET y J2EE* del Método Situacional para el Desarrollo de Aplicaciones *Web* (b)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI87. Estrategia basada en los servicios <i>web</i>	Implementación de aplicaciones <i>web</i> distribuidas que se comunican enviando mensajes SOAP sobre HTTP	Aplicación <i>web</i> ensamblada, y aplicaciones existentes.	(A87): Se requiere integrar la aplicación <i>web</i> con otras aplicaciones existentes y/o el grupo de desarrollo tiene experiencia en integración de aplicaciones haciendo uso de los servicios <i>web</i> y/o Se dispone de herramientas que soportan el mapeo de mensajes SOAP a invocaciones de métodos en objetos.
DEI97. Estrategia de documentación.	Elaboración del documento donde se especifica la implementación de los componentes de cada capa de la aplicación <i>web</i> , el ensamblaje de aplicación <i>web</i> y la integración de esta con otras aplicaciones existentes.	Aplicación <i>web</i> implementada, ensamblada e integrada con otras aplicaciones.	(A97): Se implementaron los componentes de cada una de las capas de la aplicación <i>web</i> y/o se ensambló la aplicación <i>web</i> y/o se realizó a integración de la aplicación <i>web</i> con otras aplicaciones existentes y/o la organización tiene cierto grado de madurez.

Las directivas de ejecución de intención (DEI) propias de implementar la aplicación *web* con *.NET* se muestran en la tabla 18, y las específicas de implementar la aplicación *web* con *J2EE* en la tabla 19.

Tabla No. 18. Directivas de Ejecución de Intención (DEI) propias del Mapa Local de Bajo Nivel de la Intención de *Implementar la Aplicación Web con .NET* del Método Situacional para el Desarrollo de Aplicaciones *Web* (a)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI80. Estrategia orientada por el <i>OLDB</i>	Implementación de las interfaces <i>OLDB</i> , con el fin de permitir el acceso a todo tipo de datos.	Fuentes de datos, interfaces <i>OLDB</i> .	(A80): Se implementó una aplicación <i>web</i> distribuida y/o se requiere que la aplicación <i>web</i> acceda a diferentes tipos de datos y/o se implementó la aplicación <i>web</i> en la plataforma <i>.NET</i> .
DEI81. Estrategia orientada al <i>XML</i>	Almacenamiento, recuperación y mantenimiento de datos en formato <i>XML</i> .	Diseño de las bases de datos.	(A81): Se está implementando una aplicación <i>web</i> <i>B2B</i> que requiere transmitir documentos y/o es necesario implementar de periódicos, artículos y publicidad de varios ítems dentro de la aplicación <i>web</i> y/o Se implementó la aplicación <i>web</i> en la plataforma <i>.NET</i> .

Licencia Creative Commons

Tabla No. 18. Directivas de Ejecución de Intención (DEI) propias del Mapa Local de Bajo Nivel de la Intención de *Implementar la Aplicación Web con .NET* del Método Situacional para el Desarrollo de *Aplicaciones Web (b)*

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI82. Estrategia orientada al <i>ADO.NET</i>	Implementación de acceso a bases de datos a través de los objetos <i>ADO.NET</i>	Bases de datos, componentes de la capa de datos, objetos <i>ADO.NET</i> .	(A82): Se requiere optimizar el acceso a bases de datos de la aplicación <i>web</i> y/o la aplicación <i>web</i> implementada está basada en <i>XML</i> y/o se requiere ejecutar código de acceso a bases de datos bajo un ambiente controlado.
DEI84. Estrategia orientada por el <i>MSMQ</i>	Ensamblaje de la aplicación <i>web</i> a través de <i>Message Queue (MSMQ)</i> con el fin de proveer soporte para enviar y recibir mensajes vía <i>HTTP</i> .	Componentes de cada una de las capas de la aplicación <i>web</i>	(A84): Se implementó una aplicación <i>web</i> distribuida y/o se implementó la aplicación <i>web</i> en la plataforma <i>.NET</i> y/o se requiere comunicar los componentes de la aplicación <i>web</i> a través de pases de mensajes bajo <i>HTTP</i> .
DEI86. Estrategia orientada por el <i>Ja.NET</i>	Integración de aplicaciones <i>web</i> implementadas en <i>J2EE</i> y <i>.NET</i>	Aplicaciones <i>web</i> y conector <i>Ja.NET</i>	(A86): Se implementó una aplicación <i>web</i> distribuida en la plataforma <i>.NET</i> y Se requiere integrar la aplicación <i>web</i> con otra desarrollada en la plataforma <i>J2EE</i> y/o el grupo de desarrollo tiene experiencia en integración de aplicaciones <i>web</i> .

Tabla No. 19. Directivas de Ejecución de Intención (DEI) propias del Mapa Local de Bajo Nivel de la Intención de *Implementar la Aplicación Web con J2EE* del Método Situacional para el Desarrollo de *Aplicaciones Web (a)*

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI88. Estrategia orientada por <i>EJB</i>	Implementación de la capa de lógica de negocio utilizando <i>Enterprise Java Beans (EJB)</i> .	Componentes de la capa de lógica de negocio y <i>EJB</i> .	(A88): Se diseñó una aplicación <i>web</i> distribuida y/o se implementó la aplicación <i>web</i> en la plataforma <i>J2EE</i> y/o los componentes de la capa de lógica de negocio serán implementados en lenguaje <i>JAVA</i> y/o los procesos del negocio están separados de las entidades del negocio y/o el grupo de desarrollo tiene experiencia en la implementación de <i>EJB</i> .
DEI89. Estrategia basada en <i>POJOs</i>	Implementación de la lógica del negocio utilizando <i>Plain Old Java Object (POJOs)</i> .	Componentes de la capa de lógica de negocio.	(A89): La capa de lógica de negocio de la aplicación <i>web</i> contiene los procesos y las entidades del negocio y/o el rendimiento y la facilidad de mantenimiento es un factor crítico en la aplicación <i>web</i> y/o se requiere implementar rápidamente los procesos de negocio de la organización y/o la aplicación <i>web</i> se implementó en la plataforma <i>J2EE</i> .

Tabla No. 19. Directivas de Ejecución de Intención (DEI) propias del Mapa Local de Bajo Nivel de la Intención de *Implementar la Aplicación Web con J2EE* del Método Situacional para el Desarrollo de Aplicaciones Web (b)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI90. Estrategia orientada al <i>JDO</i>	Implementación de los componentes de la capa de datos utilizando <i>Java Data Object (JDO)</i>	Componentes de la capa de lógica del negocio y componentes de la capa de datos.	(A90): La aplicación web se implementó en la plataforma <i>J2EE</i> y/o es necesario realizar un mapeo de modelo de objetos a un esquema relacional y/o se requiere definir un modelo estándar de persistencia de objetos y/o se requiere implementar bases de datos orientadas a objetos.
DEI91. Estrategia orientada por el <i>JDBC</i>	Implementación de los componentes de la capa de datos utilizando <i>Java Database Connectivity (JDBC)</i> .	Componentes de la capa de lógica del negocio y componentes de la capa de datos.	(A91): La aplicación web se implementó utilizando <i>POJOs</i> y/o se utilizaron <i>beans</i> de entidad con persistencia manejada (<i>BMP</i>) y/o se requiere un alto rendimiento en la actualización y recuperación de datos de la bases de datos y/o el grupo de desarrollo tiene experiencia en <i>SQL</i> .
DEI92. Estrategia orientada por los <i>beans</i> de entidad	Implementación de los componentes de la capa de datos utilizando los <i>beans</i> de entidad.	Componentes de la capa de lógica del negocio y componentes de la capa de datos.	(A92): Se diseñó una aplicación web distribuida y/o se implementaron los <i>EJB</i> y/o el esquema de bases de datos diseñado es el modelo relacional y/o se requiere reutilizar los <i>beans</i> de entidad y/o se requiere controlar fácilmente las transacciones, seguridad o concurrencia de los datos de la base de datos.
DEI93. Estrategia orientada por <i>JMS</i>	Ensamblaje de la aplicación web a través de la entrega de mensajes y componentes <i>J2EE</i> .	Componentes implementados de las diferentes capas de la aplicación web.	(A93): La aplicación web se implementó en la plataforma <i>J2EE</i> y/o se reutilizó el patrón del cliente web pesado y/o el cliente es independiente del servidor y/o se requieren crear nuevas aplicaciones <i>JAVA</i> para integrarlas entre si y/o se requieren comunicar aplicaciones escritas en diferentes lenguajes.
DEI94. Estrategia orientada por <i>JCA</i>	Integración de la aplicación web con otras aplicaciones o sistemas utilizando <i>J2EE Connector Architecture (JCA)</i> .	Aplicación web y sistemas y/o aplicaciones web desarrolladas en diferentes plataformas.	(A94): Se requiere integrar la aplicación web implementada en la plataforma <i>J2EE</i> con diferentes sistemas heterogéneos y/o el cliente y el servidor están fuertemente ligados y/o las aplicaciones web y sistemas se requieren comunicar tanto asíncrona como síncronamente.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Tabla No. 19. Directivas de Ejecución de Intención (DEI) propias del Mapa Local de Bajo Nivel de la Intención de *Implementar la Aplicación Web con J2EE* del Método Situacional para el Desarrollo de *Aplicaciones Web (c)*

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI95. Estrategia de JAVA-IDL	Integración de una aplicación web con aplicaciones y/o sistemas desarrollados en diferentes lenguajes de programación basada en <i>Common Object Request Broker Architecture (CORBA)</i>	Aplicación web y sistemas y/o aplicaciones web desarrolladas en diferentes lenguajes de programación.	(A95): Se requiere integrar la aplicación web implementada en la plataforma J2EE con otros componentes implementados en lenguajes de programación distintos a JAVA y/o se requiere enlazar servicios y atender peticiones de componentes de J2EE y/o se requiere desarrollar la interfaz del cliente con un objeto remoto mediante un IDL traducido a JAVA.
DEI96. Estrategia de RMI	Integración de una aplicación web con otros servicios proporcionados por otra aplicación a través de la comunicación remota entre aplicaciones y componentes JAVA.	Aplicaciones web o componentes escritos en el lenguaje JAVA.	(A96): Se requiere integrar dos aplicaciones o componentes escritos en JAVA y/o se requiere conectar una aplicación cliente con una aplicación servidor y/o se requiere invocar objetos ubicados en una JVM distinta a la de la aplicación web a través de RMI.

3.3.4. PROBAR LA APLICACIÓN WEB

Probar la aplicación web consiste en medir los logros alcanzados en cuanto a la calidad de la aplicación web implementada y la satisfacción que esta ofrece a los clientes. Este mapa contiene cuatro (4) intenciones fundamentales: “inspeccionar el componente”, “probar las unidades”, “probar la integración” y “probar el sistema” (ver figura 45).

La intención de “inspeccionar el componente” tiene por objetivo el encontrar fallas en un componente individual a través de la inspección manual de su código fuente. El producto generado luego de ejecutada esta intención es el componente probado y libre de fallas.

La intención de “probar las unidades” se centra en buscar fallas en pequeños bloques de la aplicación web. Los mismos son seleccionados del modelo de objetos y del modelo de producto de la aplicación web. Estas pruebas permiten corregir las fallas fácilmente debido a que son pocos los componentes que están involucrados en una unidad, asimismo, permite ejecutar simultáneamente las inspecciones a cada

componente, y las pruebas a la unidad completa. El producto generado al ejecutar esta intención es la unidad o bloque de aplicación *web* probado, el cual generalmente, es una unidad que se ensambla a una aplicación *web* o un servicio que puede ser reusado muchas veces e incorporado al mercado.

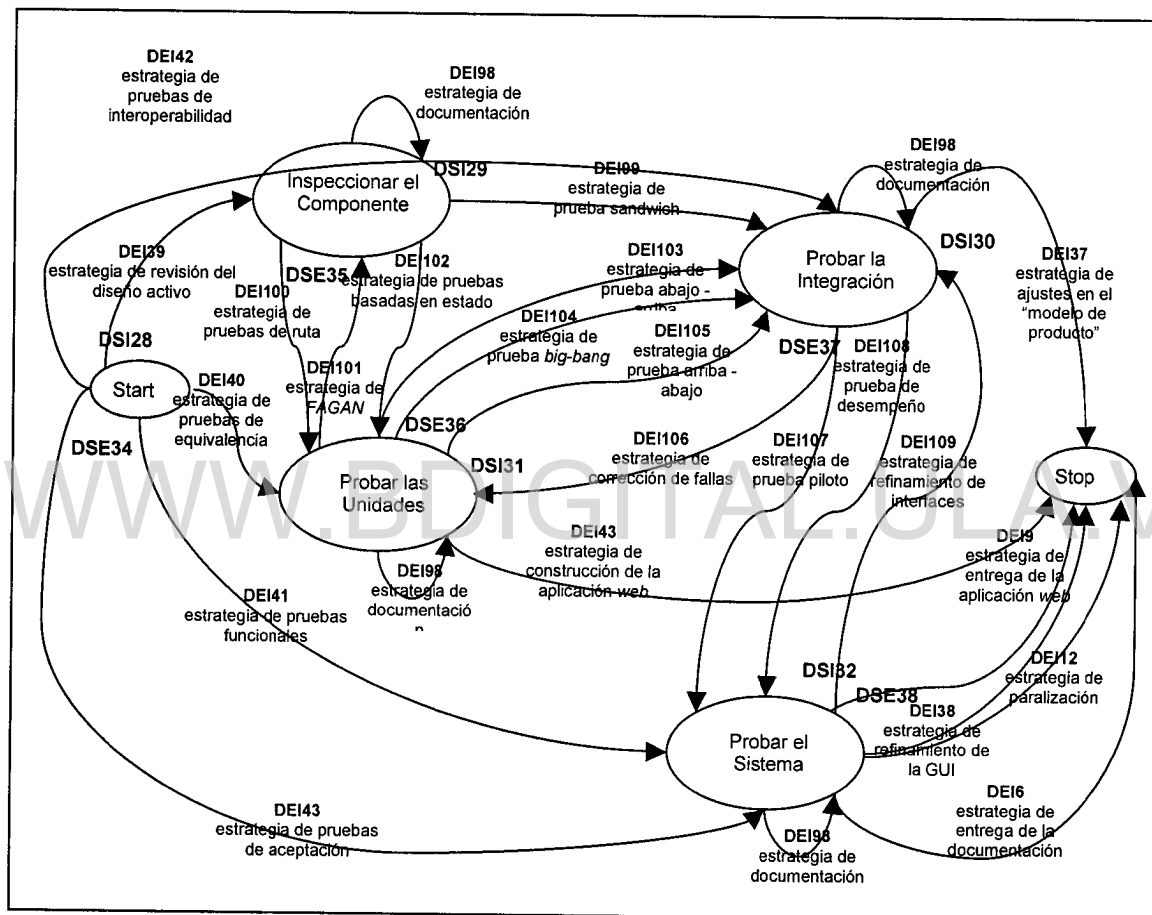


Figura 45. Mapa Local de Bajo Nivel de la Intención de *Probar la Aplicación Web* del Método Situacional para el Desarrollo de Aplicaciones *Web*

La intención de "probar la integración" consiste en probar bloques de aplicación *web* más grandes, los cuales están conformados por un grupo relativamente grande de componentes con el fin de detectar fallas que no hayan sido corregidas anteriormente. Estas pruebas se centran en verificar la interacción de los componentes a través de sus interfaces y la manera como éstos se integran para implementar un servicio en la aplicación *web*.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

La intención de “*probar el sistema*”, se refiere a las pruebas que se realizan a toda a la aplicación *web*. Estas pruebas tienen como objetivo el asegurar que la aplicación *web* ensamblada cumple con todos los requerimientos funcionales y no funcionales establecidos anteriormente. El producto generado al finalizar esta intención es la aplicación *web* probada y lista para ser instalada en su ambiente de ejecución.

Es importante aclarar, que todas las pruebas que forman parte de la intención de probar la aplicación *web* son planificadas en el plan del proyecto elaborado en la intención de gestionar el proyecto y documentadas luego de realizadas cada una de ellas.

Este mapa está conformado por los siguientes elementos:

- **6 Intenciones**, las cuales representan el objetivo que se desea satisfacer en el proceso. Estas son: Comienzo, inspeccionar el componente, probar las unidades, probar la integración, probar el sistema y parada.
- **5 Directivas de Selección de Estrategia (DSE)**, las cuales permiten seleccionar una estrategia para ejecutar una intención de modelado, a partir de una intención de partida. Estas son:
 - a) **DSE34**: De la cual se puede seleccionar la estrategia de “pruebas funcionales”, la estrategia de “pruebas de aceptación”, o ambas, con el fin de “*probar el sistema*” (la aplicación *web* ensamblada).
 - b) **DSE35**: A partir de esta directiva se puede seleccionar la estrategia de “pruebas de ruta”, la estrategia de “pruebas basadas en estado”, o ambas, con el fin de partir de la intención de “*inspeccionar el componente*” hacia la “*intención de probar las unidades*”.
 - c) **DSE36**: A partir de esta directiva se puede seleccionar la estrategia de “prueba abajo-arriba”, la estrategia de “prueba *big-bang*”, la estrategia de “prueba arriba-abajo” o cualquier combinación de las anteriores, con el fin de partir de la intención de “*probar las unidades*” hacia la intención de “*probar la integración*”.

Licencia Creative Commons:

- d) **DSE37:** A partir de esta directiva se puede seleccionar la estrategia de “prueba piloto”, la estrategia de “prueba de desempeño”, o las dos anteriores, con el fin de partir de la intención de “*probar la integración*” hacia la intención de “*probar el sistema*” (aplicación *Web*).
- e) **DSE38:** A partir de esta directiva de puede seleccionar la estrategia de “entrega de la aplicación *web*”, estrategia de “refinamiento de la GUI”, la estrategia de “paralización”, o la estrategia de “entrega de la documentación”, con el fin de partir de la intención de “*probar el sistema*” (aplicación *Web*) hacia la intención de “*parada*”.
- **5 Directivas de Selección de Intención (DSI)**, las cuales permiten seleccionar una intención destino entre las disponibles a partir de una intención de partida. Las mismas están identificadas como DSI28, DSI29, DSI30, DSI31 y DSI32.
- **12 Directivas de Ejecución de Intención (DEI)**, las cuales guían la ejecución de una intención según una estrategia específica. Estas se listan en la tabla No 20 donde se indica, su descripción, situación inicial y argumentos para la ejecución de las mismas.

Tabla No. 20. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de *Probar la Aplicación Web* del Método Situacional para el Desarrollo de Aplicaciones *Web* (a)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI98. Estrategia de documentación	Elaboración del documento donde se especifican las pruebas de unidad, pruebas de sistemas y pruebas de integración realizadas a la aplicación <i>web</i> .	Plan de pruebas y Aplicación <i>web</i> probada.	(A98): Se cuenta con el plan de pruebas y/o se realizaron todas las pruebas a la aplicación <i>web</i> y/o la organización tiene cierto grado de madurez.
DEI99. Estrategia de prueba <i>sandwich</i>	Prueba de integración de la aplicación <i>web</i> a través de la combinación de las estrategias <i>top-down</i> y <i>bottom-up</i> .	Componentes de la aplicación <i>web</i>	(A99): Los componentes de la aplicación <i>web</i> están distribuidos en tres (3) capas y/o se requiere prestar especial atención a las interfaces de los componentes y/o se requiere probar en paralelo la integración de los componentes de cada una de las capas y la integración de toda la aplicación <i>web</i> .

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Tabla No. 20. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de Probar la Aplicación Web del Método Situacional para el Desarrollo de Aplicaciones Web (b)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI100. Estrategia de pruebas de ruta	Identificación de fallas en la implementación de los componentes a través de la prueba "caja blanca".	Componentes de la aplicación web.	(A100): Se requiere probar todas las rutas posibles seguidas a través del código y su comportamiento ante un evento dado y/o los responsables de las pruebas conocen el código fuente y las estructuras de datos y/o el componente está codificado en un lenguaje imperativo y/o se requieren detectar todas las fallas relacionadas con el uso incorrecto de estructuras de datos y flujos de control.
DEI101. Estrategia de FAGAN	Inspección de un componente a través de la revisión del código fuente.	Componentes de la aplicación web.	(A101): Los responsables de las pruebas deben estar familiarizados con el código fuente y/o las revisiones del código fuente se deben realizar en reuniones de trabajo constituidas por el autor del componente, el moderador y los revisores y/o se requiere identificar las desviaciones de los estándares de codificación.
DEI102. Estrategia de pruebas basadas en estado	Prueba de bloques de la aplicación web: componentes o grupos de componentes.	Componentes de la aplicación web.	(A102): El lenguaje utilizado para implementar la aplicación web es orientado a objetos y/o se requiere comparar el estado actual de las clases que constituyen la aplicación web con el estado esperado y/o el grupo de desarrollo y los responsables de las pruebas tienen experiencia en UML.
DEI103. Estrategia de prueba abajo-arriba	Identificación de fallas al integrar varios grupos de componentes comenzando desde la capa de datos.	Componentes de la aplicación web.	(A103): Se requiere probar cada componente en forma individual y/o es necesario probar las interfaces de cada componentes cuando este es integrado con otro de su misma capa o con los componentes de la capa inmediatamente superior y/o se requiere comenzar las pruebas por la capa de datos.
DEI104. Estrategia de prueba big-bang	Prueba de integración de los componentes de la aplicación web.	Componentes de la aplicación web.	(A104): Cada uno de los componentes que constituyen la aplicación web ya fueron probados individualmente y/o las pruebas se deben realizar sobre la totalidad de la aplicación web y/o no se requiere examinar al detalle la interfaz de cada uno de los componentes.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Tabla No. 20. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de *Probar la Aplicación Web* del Método Situacional para el Desarrollo de Aplicaciones Web (c)

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI105. Estrategia de prueba arriba-abajo	Identificación de fallas al integrar varios grupos de componentes comenzando desde la capa de presentación.	Componentes de la aplicación web.	(A105): Se requiere probar cada componente en forma individual y/o es necesario probar las interfaces de cada componente cuando este es integrado con otro de su misma capa o con los componentes de la capa inmediatamente inferior y/o se requiere comenzar las pruebas por la capa de presentación.
DEI106. Estrategia de corrección de fallas	Corrección de fallas en los componentes que constituyen la aplicación web.	Componentes de la aplicación web.	(A106): Se realizaron pruebas de integración a la aplicación web y/o se identificaron fallas en algunos componentes y/o las fallas deben ser corregidas individualmente por cada componente y/o la aplicación web debe ser integrada nuevamente.
DEI107. Estrategia de prueba piloto	Pruebas de funcionalidades realizadas por grupo seleccionado de usuarios en el ambiente de ejecución.	Aplicación web operativa.	(A107): Se cuenta con un grupo de usuarios para probar las funcionalidades de la aplicación web en su ambiente de ejecución y/o no existe una documentación detallada de los requerimientos del usuario y/o el cliente y usuarios no participaron activamente en el desarrollo de la aplicación web.
DEI108. Estrategia de prueba de desempeño	Identificación de diferencias entre las metas de diseño planteadas en el diseño de la aplicación web y la aplicación web.	Aplicación web operativa.	(A108): Se cuenta con la documentación detallada de la definición y especificación de requerimientos y/o se requiere verificar si la aplicación web puede responder a múltiples solicitudes simultáneamente y/o se requiere identificar fallas en la validación de los datos y/o se requiere verificar la seguridad de la aplicación web y/o se requiere verificar los tiempos de respuesta de la aplicación web y/o se requiere verificar la habilidad que posee la aplicación web de recuperarse de errores.
DEI109. Estrategia de refinamiento de interfaces	Corrección de fallas en las interfaces de los componentes de la aplicación web.	Aplicación web operativa	(A109) Se realizaron las pruebas de sistema de la aplicación web y/o se detectaron fallas en la integración de algunos componentes y/o es necesario verificar las interfaces de los componentes de la aplicación web y/o es necesario integrar de nuevo cada uno de los componentes de la aplicación web.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

4. RECORRIDO DE LOS MAPAS DE PROCESOS: UNA APLICACIÓN REAL

La aplicación de un método situacional implica la selección y ejecución de las intenciones del mapa de procesos para construir un producto o parte de un producto determinado.

La selección y ejecución de estas intenciones es guiada por los distintos tipos de directivas: Directivas de Selección de Intención (DSI), Directivas de Selección de Estrategia (DSE) y Directivas de Ejecución de Intención.

Para un caso particular, la secuencia de ejecución de intenciones sigue determinadas estrategias, las cuales determinan lo que se llama una ruta en el mapa de procesos. La misma indica el proceso seguido para construir un producto y representa el nivel de abstracción más bajo de un fragmento de método, tal como se explicó en el capítulo 2 sección 2.

4.1. RECORRIDO DEL MAPA GLOBAL

Para efectos de validación y utilización de nuestra propuesta se construyó un prototipo de una aplicación *web*. A continuación se presenta la situación inicial de este pequeño proyecto de desarrollo.

El ejemplo que se desarrollará es un prototipo de una aplicación *web*, cuyo fin es el de implementar el método situacional para el desarrollo de aplicaciones *web*. El desarrollo de este prototipo tiene como finalidad la construcción futura, de una herramienta *CASE* que permita automatizar la aplicación de dicho método.

El proceso para desarrollar el proyecto planteado comienza con la intención "Start" del mapa global (figura 46). El conocimiento de la necesidad de desarrollar la aplicación *web* y la experiencia del grupo de desarrollo en gestionar proyectos nos ayudan a decidir que el proceso debe comenzar por la intención global de "Gestionar el desarrollo de la aplicación *web*" a través de la "estrategia orientada al problema" (1). Dicha decisión está soportada por DSI1 y la DSE2, y los argumentos que sustentan la ejecución de una DEI se encuentran descritos en la tabla No. 21. Aclarando que las DSI guían la selección de la próxima intención a ser ejecutada, partiendo de una intención; mientras que las DSE permiten tomar una decisión acerca de cual estrategia es más apropiada para ejecutar la intención seleccionada y

las DEI guían la ejecución de la intención seleccionada con la estrategia seleccionada.

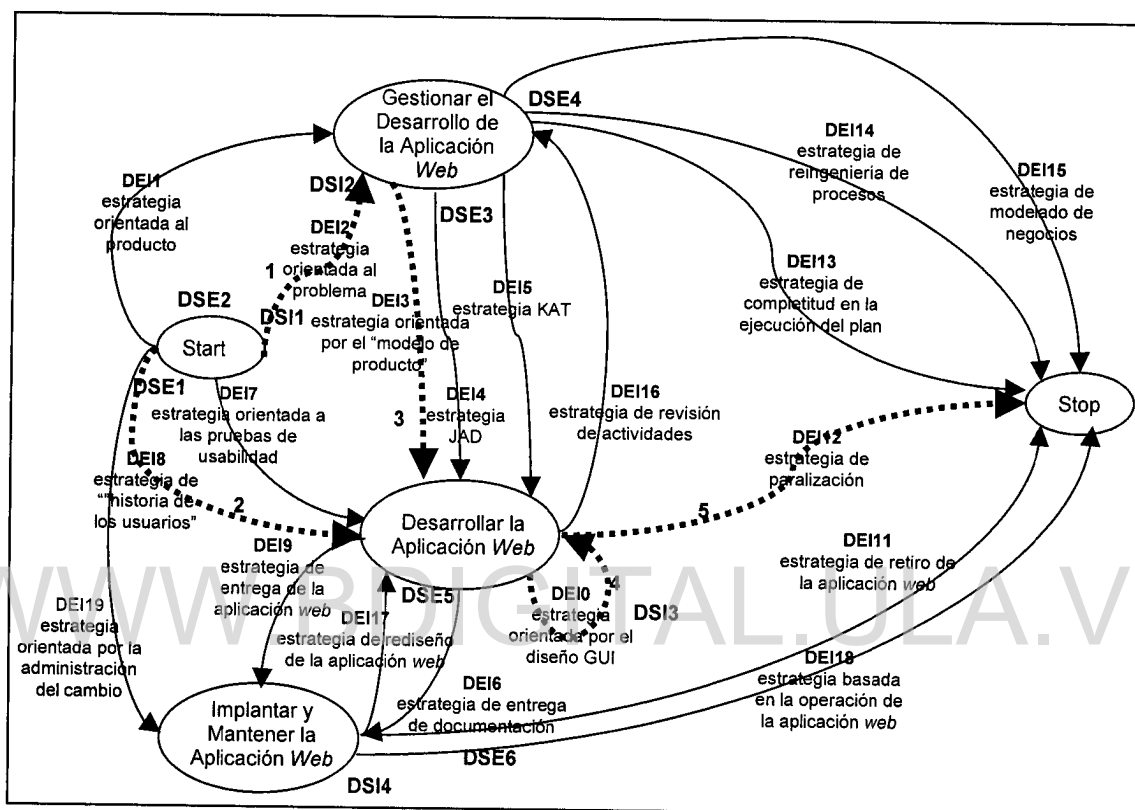


Figura 46. Recorrido del Mapa Global del Método Situacional para el Desarrollo de Aplicaciones Web

Siguiendo el mapa global y de acuerdo a nuestra situación de modelado la siguiente intención a ejecutar es la de “Desarrollar la aplicación web”, por la “estrategia de historia de los usuarios” (2). Esta decisión está soportada por DSI1 y DSE1. Asimismo, esta intención será ejecutada, posteriormente, por las “estrategia orientada por el modelo de producto” (3) y la “estrategia orientada por el diseño de la GUI” (4). Estas decisiones están soportada por DSI2, DSE3 y DSI3, respectivamente.

Finalmente decidimos ir a la intención de “Stop” por la “estrategia de paralización” (5), debido a que el objetivo es desarrollar un prototipo de lo que será la aplicación web, por tanto, no es necesario ejecutar la intención de “implantar y mantener la aplicación web”. A continuación se detallan los procesos seguidos en

cada una de las intenciones ejecutadas en sus respectivos mapas locales (figuras 47 y 48).

Tabla No. 21. Directivas de Ejecución de Intención (DEI) del Mapa Global del Método Situacional para el Desarrollo de Aplicaciones Web ejecutadas en el estudio de un caso

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI2. Estrategia orientada al problema	Establecimiento de los objetivos, el propósito del proyecto y sus requerimientos más generales.	Necesidad de desarrollo de aplicación web.	(A2): Se tiene una idea general de la aplicación web a desarrollar y/o la organización tiene cierto grado de madurez y/o existe en el grupo de desarrollo gente con experiencia en gerencia de proyectos.
DEI3. Estrategia orientada por el "modelo de producto"	Especificación de los componentes que interactúan en cada una de las capas del modelo de producto de la aplicación web y especificación de la interrelación de los mismos.	Modelo de producto de la aplicación web y requerimientos generales de la aplicación web.	(A3): Se cuenta con el modelo de producto de la aplicación web y/o no se requiere reutilizar patrones de diseño de arquitectura lógica y/o el equipo de desarrollo conoce al detalle cada uno de los componentes que integran la aplicación web.
DEI8. Estrategia de "historia de usuarios".	Extracción y fraccionamiento de las "historias de los usuarios" (requerimientos escritos) que representen más de un producto funcional.	Plan de desarrollo de la aplicación web. (tiempo y recursos estimados).	(A8): La aplicación web se desarrolla utilizando métodos ágiles y/o se requiere introducir rápidamente la aplicación web en el mercado y el grupo de desarrollo es experto en la implementación y pruebas de aplicaciones web.
DEI10. Estrategia orientada por el diseño de la GUI.	Definición de la estructura de navegación, el estilo visual de las páginas y la interacción de la aplicación web con el usuario.	Requerimientos generales de la aplicación web.	(A10): Se requieren seguir los principios de diseño web y/o no es necesario reutilizar patrones de diseño y/o el grupo de desarrollo no dispone de un repositorio de componentes reutilizables y/o existe tiempo suficiente para desarrollar la aplicación web y/o la organización tiene bajo grado de madurez.
DEI12. Estrategia de Paralización	Culminación del desarrollo de la aplicación web y paralización de la implantación y mantenimiento de la misma.	Aplicación web desarrollada y lista para operar.	(A12): Se desarrolló la aplicación web y/o la organización no cuenta con los recursos necesarios para adquirir la plataforma de ejecución de la misma y/o los requerimientos de la organización cambiaron por movimientos realizados en la alta gerencia.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

4.2. RECORRIDO DEL MAPA LOCAL DE ALTO NIVEL: GESTIONAR EL DESARROLLO DE LA APLICACION WEB

El proceso para gestionar el desarrollo de la aplicación *web* comienza con la intención "Start" (figura 47). Como se mencionó anteriormente, el conocimiento de la necesidad de desarrollar la aplicación *web* y la experiencia del grupo de desarrollo en gestionar proyectos nos ayudan a decidir que el proceso debe comenzar por la intención de "Planificar el proyecto" a través de la "estrategia orientada al problema" (1). Dicha decisión está soportada por DSI1 y la DSE2, y los argumentos que sustentan la ejecución de esta DEI se encuentran descritos en la tabla No. 21.

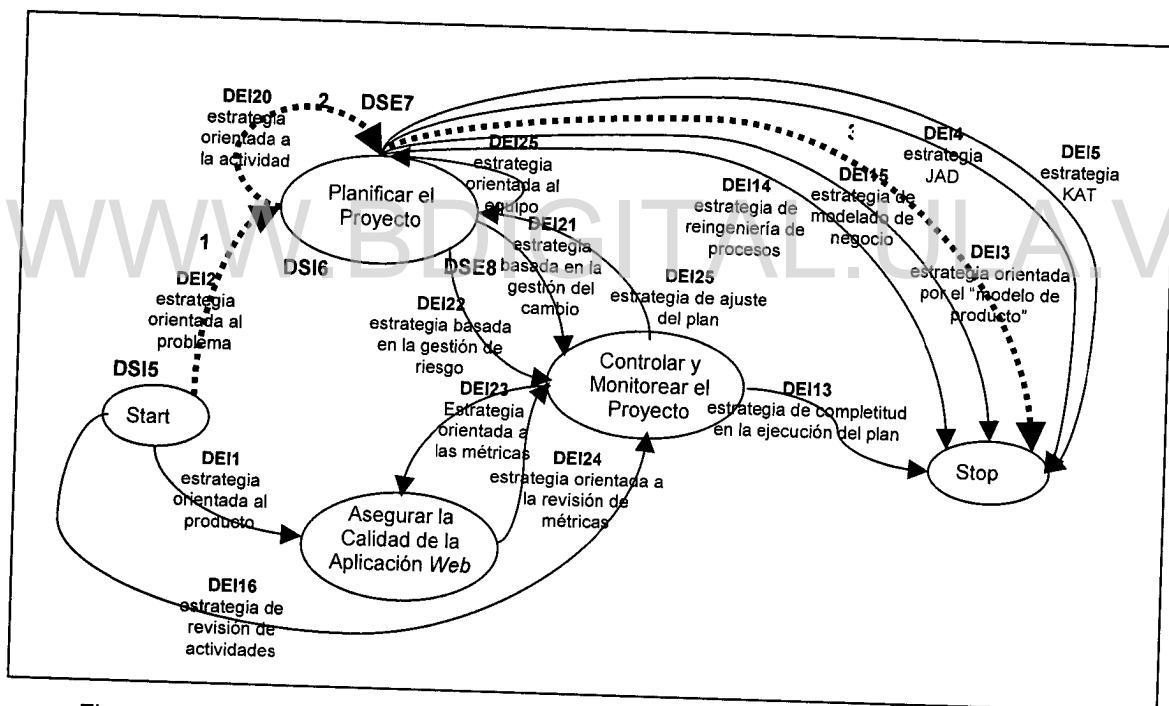


Figura 47. Recorrido del Mapa Local de Alto Nivel de la Intención de *Gestionar el Proyecto*

Siguiendo el mapa de la intención de gestionar el proyecto (figura 44) y de acuerdo a nuestra situación de modelado la siguiente intención a ejecutar es la de "Planificar el proyecto", por la "estrategia orientada a la actividad" (2). Esta decisión está soportada por DSI6.

Finalmente decidimos ir a la intención de "Stop" por la "estrategia orientada por el modelo de producto" (3) con el fin de especificar los componentes que

interactúan en cada una de las capas y ejecutar la intención de “Desarrollar la aplicación web”. Los argumentos y la situación inicial de las DEI ejecutadas en esta intención se muestran en la tabla No. 22.

Tabla No. 22. Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto Nivel de la Intención de *Gestionar el Proyecto* ejecutadas en el estudio de un caso.

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI20. Estrategia orientada a la actividad.	Determinación de las actividades necesarias relevantes en cada una de las fases de desarrollo.	Tiempo, recursos y personal necesario para la ejecución del proyecto.	(A20): Existe un modelo de negocio y se conocen los requerimientos generales con que debe cumplir la aplicación web y/o el equipo tiene experiencia en el desarrollo de aplicaciones web.

4.3. RECORRIDO DEL MAPA LOCAL DE ALTO NIVEL: DESARROLLAR LA APLICACIÓN WEB

El proceso para desarrollar la aplicación web comienza con la intención “Start” (figura 48). La necesidad de desarrollar rápidamente el prototipo y la experiencia en la implementación y pruebas de aplicaciones web del grupo de desarrollo nos ayudan a decidir que el proceso debe comenzar por la intención de “Definir y especificar requerimientos” a través de la “estrategia de historia de usuarios” (1).

Siguiendo el mapa de la intención de desarrollar la aplicación web y de acuerdo a nuestra situación de modelado, ahora se ejecuta la intención de “Diseñar la aplicación web” a través de las “estrategia orientada al modelo de producto” (2) y la “estrategia orientada por el diseño de la GUI” (3). Dichas decisiones están soportadas por DSI2, DSE3 y DSI3, respectivamente. Asimismo, los argumentos que sustentan la ejecución de estas DEI se encuentran descritos en la tabla No. 21.

Luego de haber definido el modelo de producto y diseñado la GUI, es necesario ejecutar de nuevo la intención de “Definir y especificar requerimientos” (4), pero esta vez siguiendo la “estrategia orientada al refinamiento”, con el fin de completar la definición de requerimientos. Esta decisión está sustentada por la DS10.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

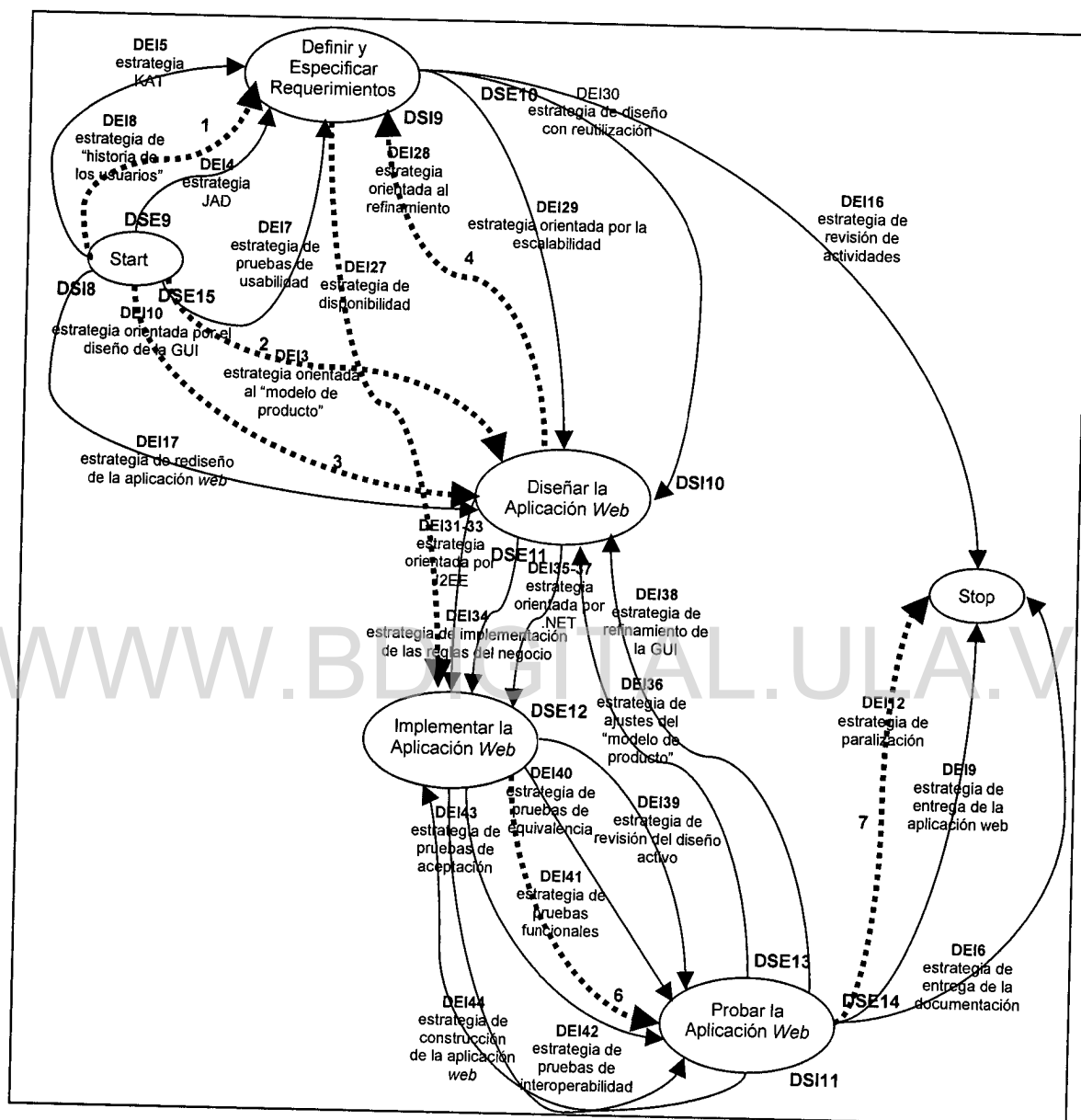


Figura 48. Recorrido del Mapa Local de Alto Nivel de la Intención de Desarrollar la Aplicación Web

La siguiente intención a ejecutar es la de "Implementar la aplicación web", siguiendo la "estrategia de disponibilidad" (5), ya que se requiere una rápida implementación del prototipo. Esta decisión está soportada por la DSI9.

Continuando con el modelo de procesos, la próxima decisión está asociada con la intención de "Probar la aplicación web" siguiendo la "estrategia de pruebas funcionales" (6), debido a que se requieren probar aquellos servicios que son

relevantes para la implementación del prototipo. Dicha decisión está soportada por la DSE12.

Finalmente decidimos ir a la intención de “*Stop*” por la “*estrategia de paralización*” (7), ya que a este nivel ya se ha cumplido con el objetivo de desarrollar la aplicación *web*. Los argumentos y la situación inicial de las DEI ejecutadas en esta intención se muestran en la tabla No. 23. A continuación se detallan los procesos seguidos en cada una de las intenciones ejecutadas en el proceso de desarrollar la aplicación *web*.

Tabla No. 23. Directivas de Ejecución de Intención (DEI) del Mapa Local de Alto Nivel de la Intención de *Desarrollar la Aplicación web* ejecutadas en el estudio de un caso

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI27. Estrategia de disponibilidad	Construcción del código fuente y preparación de las pruebas.	“Historia de los usuarios”(requerimientos)	(A27):Se requiere una rápida implementación de la “historia de los usuarios” y/o el grupo de desarrollo tiene experiencia en desarrollar aplicaciones <i>web</i> bajo el enfoque ágil y/o se dispone de componentes reutilizables de fácil adaptación en la aplicación <i>web</i> .
DEI28. Estrategia orientada al refinamiento	Refinamiento de la definición y especificación de requerimientos, luego que ha habido un acuerdo entre el cliente y el grupo de desarrollo.	Prototipo de la aplicación <i>web</i>	(A28):Se diseñó un prototipo de la aplicación <i>web</i> y/o el cliente y el grupo de desarrollo definieron claramente las expectativas que debe satisfacer la aplicación <i>web</i> y/o los requerimientos no estaban completamente definidos y especificados.
DEI41. Estrategia de pruebas funcionales	Determinación de las diferencias entre los requerimientos funcionales y la aplicación <i>web</i> .	Documento de especificación de requerimientos y aplicación <i>web</i> operativa.	(A41):Se dispone de la aplicación <i>web</i> operativa y se dispone del documento de especificación de requerimientos y/o se requieren probar aquellos servicios que son relevantes para los usuarios y clientes.

4.4. RECORRIDO DEL MAPA LOCAL DE BAJO NIVEL DE LA INTENCIÓN “DESARROLLAR LA APLICACIÓN WEB”: DEFINIR Y ESPECIFICAR REQUERIMIENTOS

El proceso para definir y especificar requerimientos comienza con la intención “*Start*” (figura 49). La necesidad de desarrollar rápidamente el prototipo y la experiencia en la implementación y pruebas de aplicaciones *web* del grupo de

Licencia Creative Commons:

desarrollo nos ayudan a decidir que el proceso debe comenzar por la intención de “Definir requerimientos” a través de la “estrategia de historia de usuarios” (1).

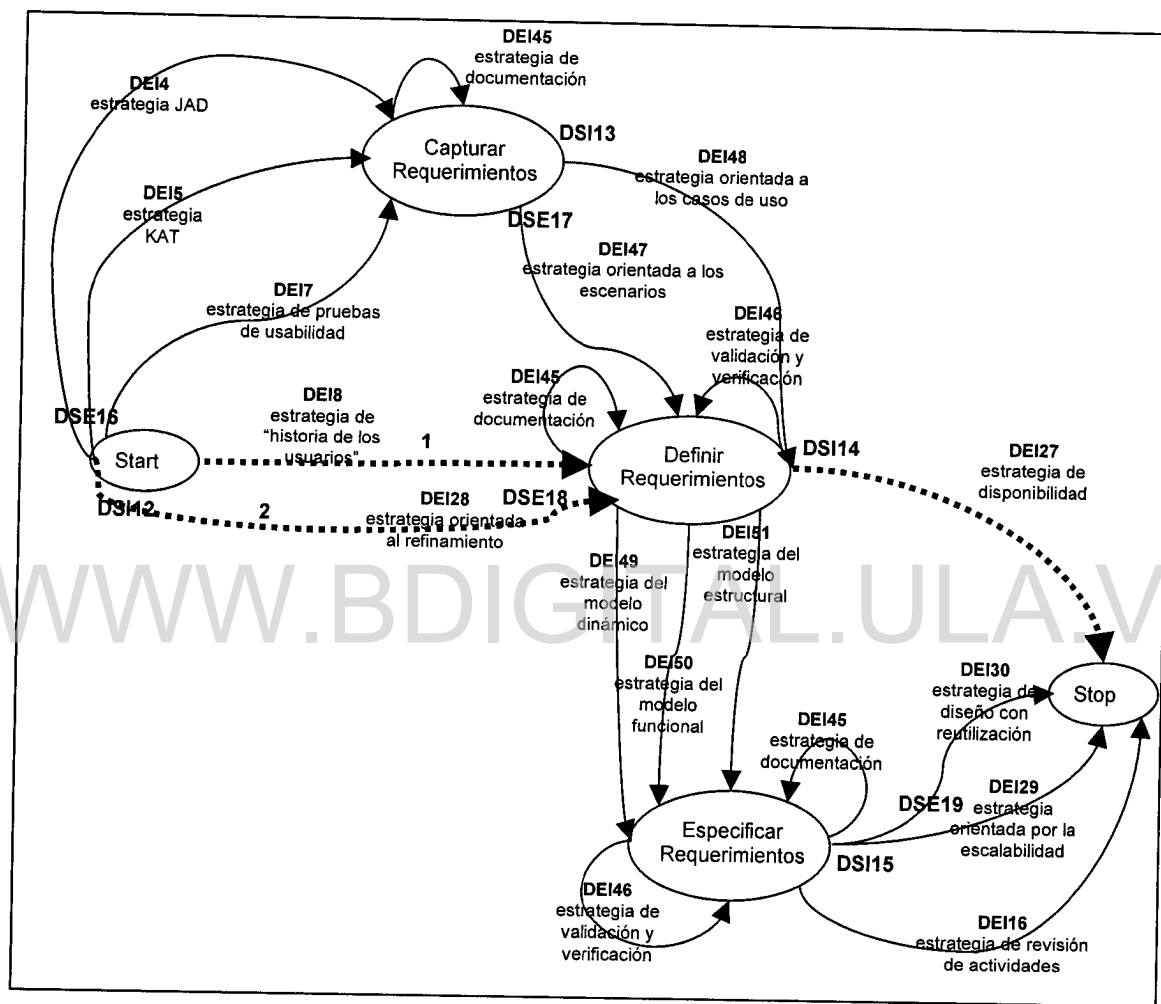


Figura 49. Recorrido del Mapa Local de Bajo Nivel de la Intención de Definir y Especificar Requerimientos

Siguiendo el mapa de la intención de definir y especificar requerimientos y de acuerdo a nuestra situación de modelado, luego se ejecuta la misma intención pero por la “estrategia orientada al refinamiento” (2). Es importante aclarar, que estas directivas de ejecución de intención (DEI) no se realizan secuencialmente. Primero se ejecuta la “estrategia de historia de los usuarios”, luego se ejecuta la intención de diseñar la aplicación web (figura 47) y por último se ejecuta la intención de “definir

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

requerimientos” por la *“estrategia orientada al refinamiento”*. Esta decisión está soportada por la DSI10.

Finalmente decidimos ir a la intención de *“Stop”* por la *“estrategia de disponibilidad”* (3), ya que se requiere una rápida implementación del prototipo. Esta decisión está soportada por la DSI14.

4.5. RECORRIDO DEL MAPA LOCAL DE BAJO NIVEL DE LA INTENCIÓN “DESARROLLAR LA APLICACIÓN WEB”: DISEÑAR LA APLICACIÓN WEB

El proceso de diseñar la aplicación *web* comienza con la intención *“Start”* (figura 50). El hecho de contar con un modelo de producto genérico, que representa las características fundamentales de una aplicación *web*, nos ayuda a decidir que el proceso debe comenzar por la intención de *“Diseñar la arquitectura lógica”* a través de la *“estrategia orientada por el modelo de producto”* (1).

Siguiendo el mapa de la intención de diseñar la aplicación *web* y de acuerdo a nuestra situación de modelado, luego se ejecuta la intención de *“Diseñar la capa de presentación”* seguida por la *“estrategia orientada por el diseño del GUI”* (2). Esta decisión está soportada por la DSI3.

Finalmente decidimos ir a la intención de *“Stop”* por la *“estrategia orientada al refinamiento”* (3), ya que es necesario completar la definición de requerimientos. Esta decisión está soportada por la DSI10.

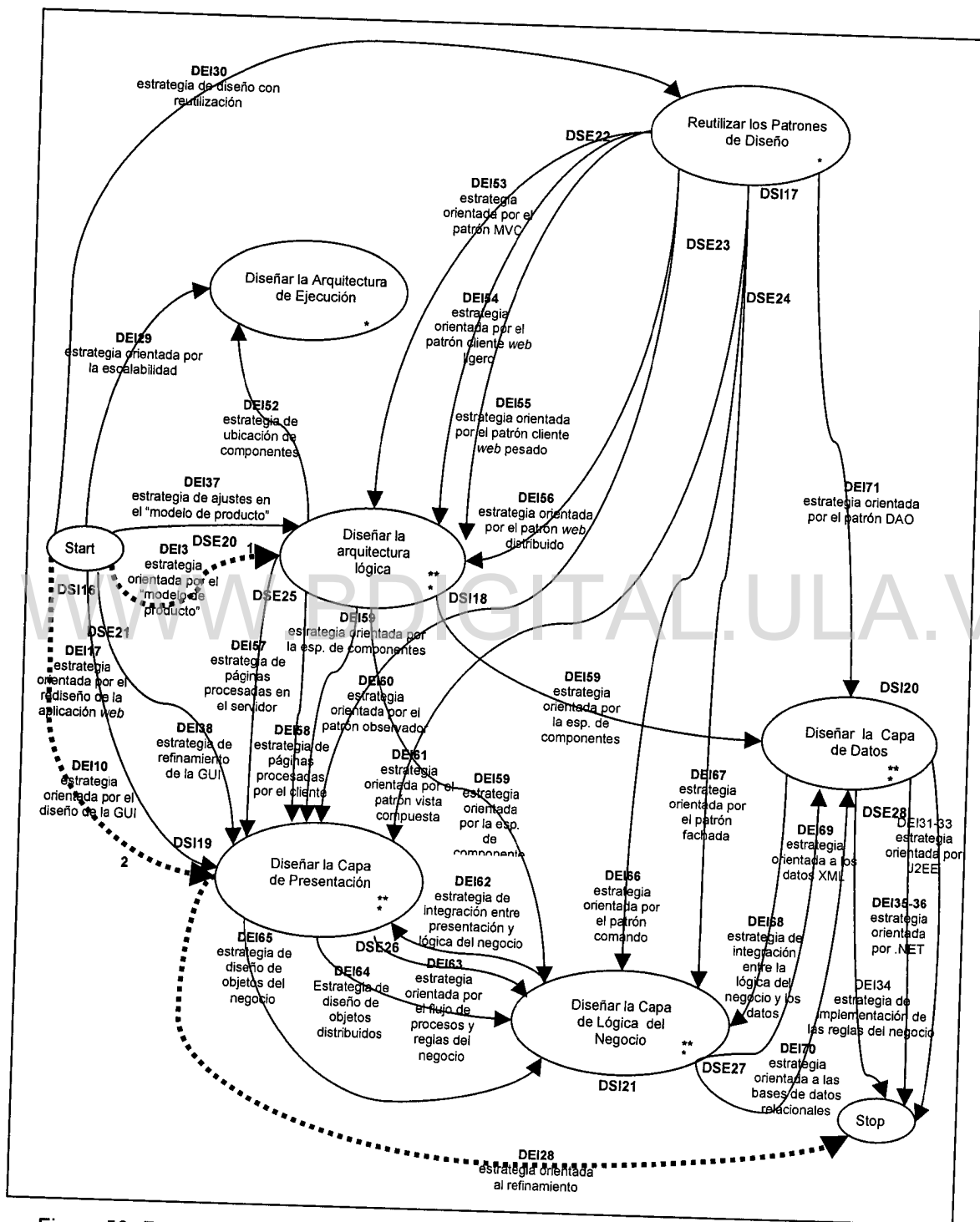


Figura 50. Recorrido del Mapa Local de Bajo Nivel de la Intención de Diseñar la Aplicación Web

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

4.6. RECORRIDO DEL MAPA LOCAL DE BAJO NIVEL DE LA INTENCIÓN “DESARROLLAR LA APLICACIÓN WEB”: IMPLEMENTAR LA APLICACIÓN WEB

El proceso para implementar la aplicación *web* comienza con la intención “*Start*” (figura 51). La necesidad de implementar rápidamente el prototipo, nos ayuda a decidir que el proceso debe comenzar por la intención de “*Reutilizar componentes*” a través de la “*estrategia de disponibilidad*” (1).

Siguiendo el mapa de la intención de implementar la aplicación *web* y de acuerdo a nuestra situación de modelado, se ejecutan, en forma iterativa, la intención de “*implementar los componentes de la capa de presentación*” por las “*estrategia de adaptación*” (2) y “*estrategia de validación y verificación*” (3) y la intención de “*reutilizar componentes*” por la “*estrategia de búsqueda*” (4), ambas intenciones de ejecutan hasta lograr implementar el prototipo requerido.

Finalmente decidimos ir a la intención de “*Stop*” por la “*estrategia de pruebas funcionales*” (5), ya que es necesario probar aquellos servicios que son relevantes para la implementación del prototipo. Esta decisión está soportada por la DSE12. Los argumentos y la situación inicial de las DEI ejecutadas en esta intención se muestran en la tabla No. 24.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

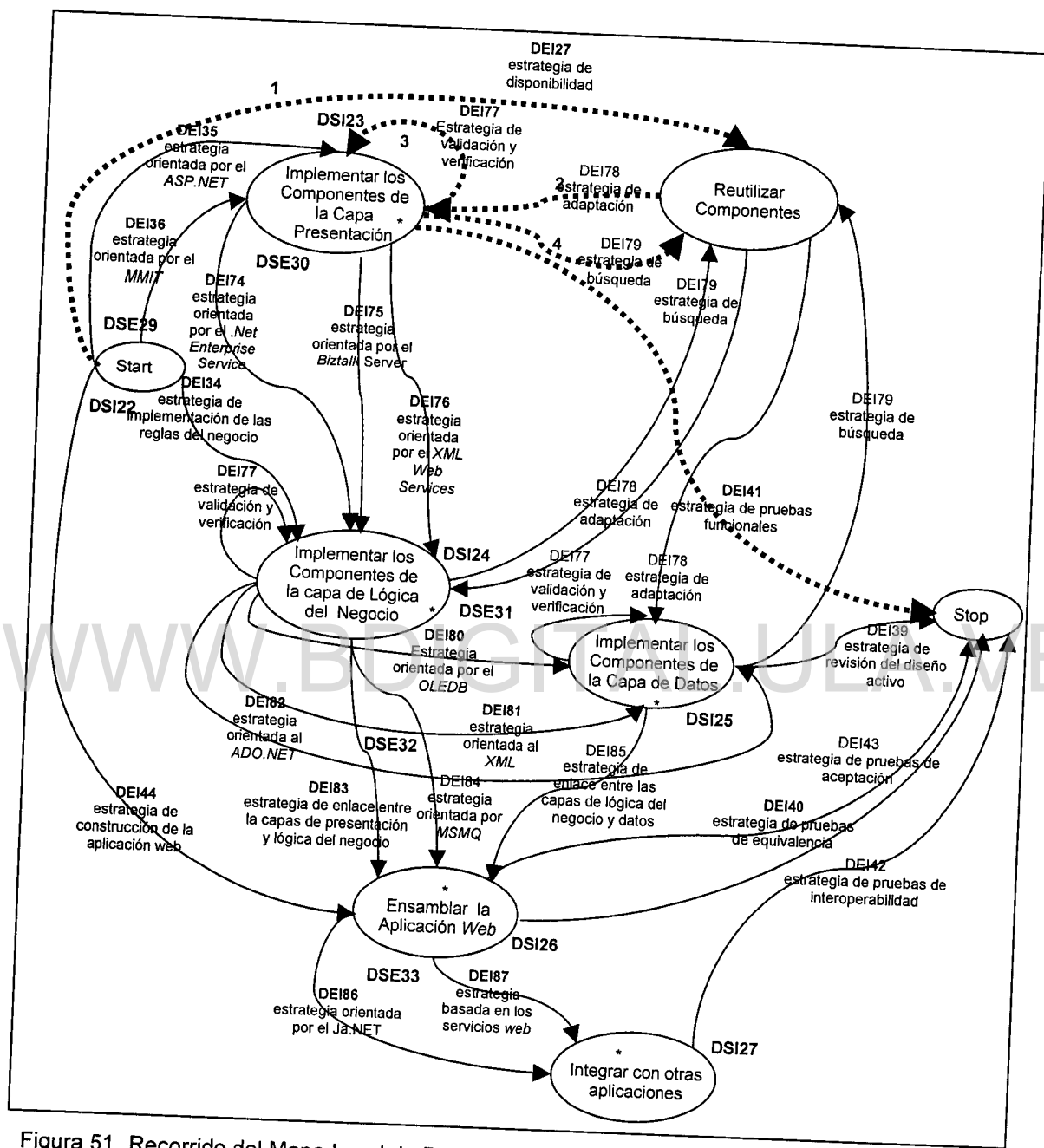


Figura 51. Recorrido del Mapa Local de Bajo Nivel de la Intención de Implementar la Aplicación Web

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

Tabla No. 24. Directivas de Ejecución de Intención (DEI) del Mapa Local de Bajo Nivel de la Intención de *Implementar la Aplicación web* ejecutadas en el estudio de un caso

Directiva de Ejecución de Intención	Descripción	Situación Inicial	Argumento
DEI77. Estrategia de validación y verificación	Comprobación de que el código fuente de cada uno de los componentes de las capas de presentación, lógica del negocio y datos es consistente con el diseño de la aplicación <i>web</i> .	Componentes de las capas de presentación, lógica del negocio y datos implementados.	(A77):El grupo de desarrollo se preocupa por asegurar la calidad de los componentes implementados y/o la organización tiene cierto grado de madurez y/o el grupo de desarrollo requiere comprobar que los componentes implementados se corresponde con el diseño realizado anteriormente.
DEI78. Estrategia de adaptación	Creación de interfaces que permitan adaptar los componentes reutilizados en la aplicación <i>web</i>	Componentes reutilizados.	(A78):El grupo de desarrollo requiere aumentar la productividad en la implementación de la aplicación <i>web</i> y/o el grupo de desarrollo tiene experiencia en las técnicas de adaptación de componentes y/o los componentes encontrados están probados y certificados.
DEI79. Estrategia de búsqueda	Localización de componentes de las capas de presentación, lógica del negocio y datos en librerías de componentes reutilizables.	Librerías de componentes reutilizables.	(A79):El grupo de desarrollo requiere aumentar la productividad en la implementación de la aplicación <i>web</i> y/o el grupo de desarrollo dispone de librerías de componentes reutilizables y/o es necesario disponer rápidamente de la aplicación <i>web</i> .

4.7. RECORRIDO DEL MAPA LOCAL DE BAJO NIVEL DE LA INTENCIÓN “DESARROLLAR LA APLICACIÓN WEB”: PROBAR LA APLICACIÓN WEB

El proceso para probar la aplicación *web* comienza con la intención “Start” (figura 52). La necesidad de probar aquellos servicios que son relevantes para la implementación del prototipo, nos ayuda a decidir que el proceso debe comenzar por la intención de “Probar el sistema” a través de la “estrategia de pruebas funcionales” (1).

Finalmente decidimos ir a la intención de “Stop” por la “estrategia de paralización” (2), ya que a este nivel ya se ha cumplido con el objetivo de desarrollar la aplicación *web*.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

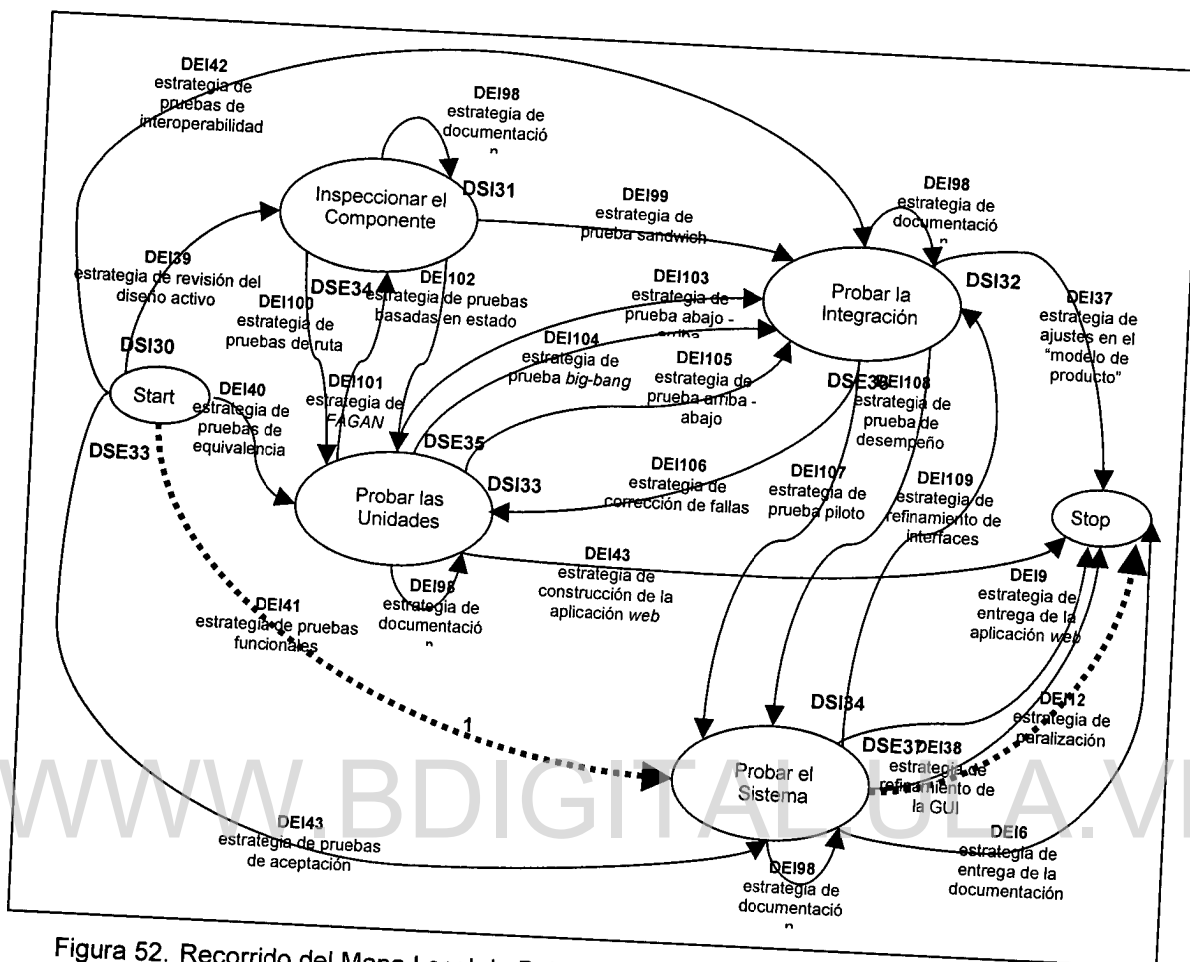


Figura 52. Recorrido del Mapa Local de Bajo Nivel de la Intención de Probar la Aplicación Web

5. CONCLUSION

En este capítulo se presentó el modelo de procesos asociado al método para desarrollar aplicaciones web. Para ello se utilizaron las tres (3) técnicas de la Ingeniería de Métodos: el ensamblaje de fragmentos de métodos, el enfoque basado en la extensión y el enfoque basado en el paradigma.

Por tanto, este método, cumple con los principios de la Ingeniería de Métodos y de la Ingeniería Web y posee un enfoque orientado al contexto. El mismo, ofrece la posibilidad, al grupo de desarrollo, de elegir entre múltiples alternativas cuál es la más conveniente, en función del dominio de aplicación y de la situación particular de modelado enfrentada por tal grupo de desarrollo. Por tal razón, este modelo de procesos se expresó en términos de intenciones, directivas de selección de

Licencia Creative Commons:

estrategia (DSE), directivas de selección de intención (DSI) y directivas de ejecución de intención (DEI).

Asimismo, se aplicó el método a un estudio de caso real, el cual consistió en el desarrollo de un prototipo de una aplicación *web*, cuyo fin es el de implementar el método situacional para el desarrollo de aplicaciones *web*.

Se obtuvo como resultado el prototipo de la aplicación *web* implementado y probado, además de las rutas realizadas en cada uno de los mapas que forman parte del modelo de proceso. Logrando probar que el método puede TAMBIEN ser usado en desarrollo de prototipos, ya que el mismo fue construido para utilizarse tanto en el desarrollo de aplicaciones *web* pequeñas y simples como en el de aplicaciones grandes y complejas. Siendo posible, además, utilizarlo en diferentes contextos de modelado, a través de la toma de decisiones según la situación particular enfrentada en cada proyecto; y garantizar así, una guía completa al equipo de desarrollo, tomando en cuenta las necesidades particulares del dominio de aplicación seleccionado (comercio electrónico, gobierno electrónico, etc).

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

CONCLUSIONES

Existen varios aspectos relevantes que han surgido durante el desarrollo de esta investigación entre los cuales se destacan los siguientes:

1. En relación a la propuesta

- Se diseñó un modelo de producto genérico que representa los conceptos más comúnmente encontrados en una aplicación *web*, tomando como base las características y elementos particulares de cinco (5) tipos de aplicaciones *web*: portales, comercio electrónico, negocios inteligentes, sistemas de información *web* y gobiernos electrónicos. Este modelo de producto es parte fundamental del método situacional para desarrollar aplicaciones *web*.
- Se agruparon las fases comunes de los métodos para el desarrollo de aplicaciones *web* seleccionados, con el fin de utilizarlas como punto de partida en la definición de las intenciones globales de nuestro método situacional, a través de la Reutilización de algunos fragmentos de dichos métodos.
- El método situacional propuesto asocia las actividades con los productos, permite la toma de decisiones según la situación particular enfrentada en cada proyecto; y garantiza una guía completa al equipo de desarrollo, tomando en cuenta las necesidades particulares del dominio de aplicación seleccionado (comercio electrónico, gobierno electrónico, etc).
- El método situacional es un valor muy importante, en relación con los métodos para desarrollar aplicaciones *web* que existen actualmente, debido a que estos últimos dependen de la disponibilidad de la experiencia individual, mientras que en el método situacional esta experiencia forma parte explícita

del método, lo que permite guiar al grupo de desarrollo indistintamente de la experiencia que éste posea.

- Se aplicó el método situacional a un estudio de un caso real, el cual consistió en el desarrollo de un prototipo de una aplicación *web*, cuyo fin es el de implementar el método situacional para el desarrollo de aplicaciones *web*. Se obtuvo como resultado el prototipo de la aplicación *web* implementado y probado, además de las rutas realizadas en cada uno de los mapas que forman parte del modelo de proceso asociado al método. Logrando probar que el método puede TAMBIEN ser usado en desarrollo de prototipos, ya que el mismo fue construido para utilizarse tanto en el desarrollo de aplicaciones *web* pequeñas y simples como en el de aplicaciones grandes y complejas.

- El método situacional para el desarrollo de aplicaciones *web* puede ser usado en el desarrollo de aplicaciones *web* pequeñas, como la presentada en el estudio del caso. Además, es posible utilizarlo en el desarrollo de aplicaciones *web* grandes y complejas y en diferentes contextos de modelado, a través de la toma de decisiones según la situación particular enfrentada en cada proyecto; y garantizar así, una guía completa al equipo de desarrollo, tomando en cuenta las necesidades particulares del dominio de aplicación seleccionado (comercio electrónico, gobierno electrónico, etc).

2. En relación a la Ingeniería de Métodos

- En la construcción del método situacional para desarrollar aplicaciones *web* se utilizó una combinación de las tres técnicas para la construcción de métodos situacionales de la Ingeniería de Métodos: el ensamblaje de fragmentos de métodos, la Ingeniería de Métodos basada en la extensión y la Ingeniería de Métodos basada en el enfoque basado en el paradigma, lo que

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

demuestra que este método cumple con los principios de la Ingeniería de Métodos y posee un enfoque orientado al contexto.

- El proceso de construcción de un método situacional para desarrollar aplicaciones *web* es complejo, debido a que involucra la integración de una diversidad de elementos, tales como: principios de la Ingeniería de Métodos, principios de la Ingeniería Web, análisis de diversos dominios de aplicación, evaluación de los métodos existentes, conocimientos de expertos en diferentes áreas (patrones de diseño, tecnologías de integración, desarrollo de aplicaciones).
- En el proceso de construcción del método situacional a través de mapas y directivas, no se cuenta con una notación formal y estandarizada que guíe el proceso de construcción de los mapas globales y locales, y la definición de las directivas. Asimismo, los mapas se hacen más difíciles de leer a medida que se incrementa el conocimiento, por no contar con una herramienta que automatice dicho proceso.
- El método situacional para desarrollar aplicaciones *web* contempla el ciclo de vida de una aplicación *web*. Sin embargo, el objetivo fundamental de esta investigación consistió en abarcar el proceso de desarrollo de la misma. Por tal razón, se presentó de manera muy general los modelos de producto y de proceso relacionados con la gestión e implantación y mantenimiento de una aplicación *web*.
- El modelo de procesos del método situacional propuesto consta de ocho (8) mapas, cuarenta y nueve (49) intenciones, cuarenta y un (41) Directivas de Selección de Estrategia (DEI), treinta y seis (36) Directivas de Selección de Intención y ciento dieciocho (118) Directivas de Ejecución de Intención (DEI). Esto muestra que un método situacional es un trabajo arduo que requiere de

Licencia Creative Commons

tiempo y dedicación, no obstante esto se convierte en beneficio para quienes lo utilizan.

- Dada que la construcción de un método situacional es un proceso complejo y a las limitaciones de tiempo, las Directivas de Ejecución de Intención (DEI) que forman parte del método situacional se presentaron de manera general, ya que no se construyó el nivel de contexto, el cual está conformado por el contexto plan, el contexto guía y el contexto de ejecución.

WWW.BDIGITAL.ULA.VE

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

RECOMENDACIONES

Considerando la gran cantidad de aspectos que no se desarrollaron por estar fuera del alcance de esta tesis, se recomienda lo siguiente:

- Completar el nivel de contexto de las Directivas de Ejecución de Intención (DEI), es decir, construir los contextos plan, de selección y de ejecución.
- Construir de manera detallada los modelos de producto y de procesos relacionados con la gestión, implantación y mantenimiento de una aplicación *web*, e integrarlos con los modelos de producto y de procesos relacionados con el desarrollo de las mismas con el fin de completar el método situacional propuesto en esta investigación.
- Aplicar a casos de la vida real el método situacional para el desarrollo de aplicaciones *web*, con el fin de validarlo y de perfeccionarlo.
- Construir una herramienta que apoye el uso del método situacional para el desarrollo de aplicaciones *web*, con el fin de facilitar la utilización del mismo en diversos casos.
- Mejorar y actualizar continuamente el método, a medida que vayan surgiendo nuevas tecnologías y nuevo conocimiento suministrado por expertos.
- A los expertos en el área de Ingeniería de Métodos, se les recomienda que continúen trabajando en el área a fin de facilitar la construcción e implementación de nuevos métodos, a través del uso de notaciones estandarizadas, claras y fáciles de usar y de herramientas que faciliten el proceso de construcción de este tipo de métodos.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

REFERENCIAS Y BIBLIOGRAFIA

- ✓ Acebal Cesar y Cueva Juan M.; Extreme Programing (XP): un nuevo método de desarrollo de software; Universidad de Oviedo, Depto. de Informática, Área de lenguajes y sistemas informáticos, 2002.
- ✓ Alanis, Macedonio; Diplomado de Gobierno Electrónico; Planeación, Diseño e Implantación de una Estrategia de Gobierno; Tecnológico de Monterrey; México, 2003.
- ✓ Anastasopoulos M, Bayer J. y otros; A Process for Product Line Architecture Creation and Evaluation, PuLSE-DSSA-Version 2.0. Technical Report IESE-038.00/E, Fraunhofer Institut Experimentelles Software Engineering, June 2000.
- ✓ Barrios Judith; Material del seminario de Ingeniería de Métodos; Universidad de Los Andes, Postgrado en Computación, 2002.
- ✓ Barrios, Judith y Montilva, Jonás; Business Modelling Through Roadmaps; Universidad de Los Andes, Facultad de Ingeniería, Escuela de Ingeniería de Sistemas, Departamento de Computación; Mérida – Venezuela; 2003.
- ✓ Barrios, Judith y Montilva, Jonás; A Methodological Framework For Business Modelling; Computing Department, Faculty of Engineering, University of Los Andes; Mérida – Venezuela, 2002.
- ✓ Barrios, Judith y Surcan S.; MEDIC: A Method Engineering Proposal for the Analysis and representation of the Organisational Impact Of Change; SERP'2002 Conference held in Las Vegas, 2002.
- ✓ Bayer J., DeBaud J. y otros; PuLSE: A Methodology to Develop Software Product Lines. In Symposium on Software Reusability (SSR '99), May 1999.
- ✓ Beck, Kent; Extreme Programming Explained: Embracing Change; Reading, MA: Addison – Wesley, 2000.
- ✓ Boban Marija y Sertié Hrvoje; Organization of Software Development for e-Business; Faculty of Law, University of Split; Domovonskog; 2003.
- ✓ Brinkkemper, S.; Method Engineering: Engineering of Information Systems Development Methods and Tools; Information and Software Technology, Vol. 37, No. 11, 1995.
- ✓ Bruegge, B. y Dutoit A.; Object-Oriented Software Engineering; Technical University of Munich. Department of Computer Science. Prentice Hall. Munich Germany, 2000

- ✓ Cáceres, Paloma y Marcos Esperanza; Procesos Ágiles para el Desarrollo de Aplicaciones Web. Universidad Rey Juan Carlos, Departamento de Ciencias Experimentales e Ingeniería, 2001.
- ✓ Canós J., Letelier P. y Penadés M.; Metodologías Ágiles en el Desarrollo de Software; DSIC; Universidad Politécnica de Valencia; Valencia – España, 2003.
- ✓ Cap Gemini Ernst & Young; Dynamic Systems Development a recipe for repeatable success; DSDM Competence Center; 2002.
- ✓ Cameron, Robert; Web-Site Architecture; Febrero; 2003. [Version electronica]. <http://www.cs.sfu.ca/~cameron/Teaching/470/site-arch.html>
- ✓ Capanegra R.; Método para la Implantación de Gobierno Electrónico; Diplomado de Gobierno Electrónico; Tecnológico de Monterrey; México, 2003.
- ✓ Carnegie Mellon Software Engineering Institute; Capability Maturity Model Integration (CMMI). Version 1.1; CMU/SEI 2002.
- ✓ Cormack G., Griffiths G. y otros; Web Engineering: methods & Tools for Education, School of Computing & Mathematics, University of Teesside Middlesbrough, 2002.
- ✓ Costagliola, Genaro; Ferrucci, Filomena y Francese Rita; Web Engineering: Models and Methodologies for the Design of Hypermedia Applications; Dipartimento di Matematica e Informatica – Università di Salerno, Italia, 2002.
- ✓ Cronholm, Stefan & Agerfalk, Pär; On the Concept of Method in Information Systems Development; Dept. of Computer and information science, Linköping University, Sweden, 1999.
- ✓ Czarnecki, K.; Levaring Reuse Through Domain-Specific Software Architectures. In *Workshop on Institutionalizing Software Reuse (WISR'8)*, Columbus, Ohio, Marzo, 1997. Position Paper.
- ✓ E-Government Interoperability Framework; PART 2: Policies and Specifications; Version 5.1.; Crown 2003.
- ✓ Escalona M. y Koch N.; Ingeniería de Requisitos en Aplicaciones para la Web: Un estudio comparativo; Universidad de Sevilla; Lenguaje y Sistemas Informáticos; España, 2003.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)

- ✓ Fernández, Daniel; Definición de una Arquitectura de Software para el Diseño de Aplicaciones Web Basadas en Tecnología Java-J2EE; Postgrado en Informática; Curso Tecnologías Web; Universidad de Oviedo; España, 2003.
- ✓ Fingar, Peter; Marco de referencia basado en componentes para E-commerce, ACM, 2000.
- ✓ Fowler Martín; The New Methodology; Chief Scientist, ThoughtWorks, June 2002.
- ✓ Ginige, Athula y Murugesan San; Web Engineering: An Introduction; Universidad de Western Sidney, Australia, 2001.
- ✓ Griss, M., Favaro y D'Alessandro. Integrating Feature Modeling with the RESB. In *Fifth International Conference on Software Reuse (ICSR'98)*, June 1998.
- ✓ Gruhn Volker y Schope Lothar; Software Processes for the Development of Electronic Commerce Systems; Dortmund University; Dortmund – Germany; Information and Software Technology, 2002.
- ✓ Guadalajara, Javier; Como crear un Portal en Internet; Distribucion Logista; España, 2001
- ✓ Hamar, Vanessa; Aspectos Metodológicos del Desarrollo y Reutilización de Componentes de Software; Universidad de Los Andes, Facultad de Ingeniería, Postgrado en Computación; Mérida – Venezuela, 2004.
- ✓ Harmon, Paul; Realizing E-Business UIT Components; Distributed Computing & e-Business Service; 2001.
- ✓ Harsu Maarit; A Survey on Domain Engineering; Institute of Software Systems; Tampere University of Technology; Tampere, 2003.
- ✓ Hennicker, Rolf y Koch Nora; Modelling the User Interface of Web Applications with UML; Institute of Computer Science; Ludwig-Maximilians; University of Munich; Munchen – Germany, 2002.
- ✓ Highsmith, Jim y Cutter Consortium; What is Agile Software Development?; CROSSTALK The Journal of Defense Software Engineering, Oct 2002 Issue.
- ✓ Holck, Jesper; 4 Perspectives on Web Information Systems; Copenhagen Business School, Dept. of Informatics; Proceedings of the 36th Hawaii International Conference on System Sciences; 2003.
- ✓ ISO/IEC 9126; ISO 9126: The Standard of Reference; Information technology – Software Product Evaluation – Quality Characteristics and Guidelines for their use.

- ✓ Kraiem, N., Bourguiba, S. y Selmi S.; Situational Method For Information System Project; Laboratoire RIADI-GDL, ENSI – ISG, Sousse Tunis, 1002, Tunisia, 2000.
- ✓ Kumar K. y Welke R.; Methodology Engineering: A Proposal for Situation-Specific Methodology Construction. In Challenges and Strategies for Research in System Development; 1992.
- ✓ Lafuente Guillermo y Olsina Luis; Catalogando Métricas Web; GIDIS, Grupo de I + D en Ingeniería de Software; Departamento de Informática, Facultad de Ingeniería, UNLPam; La Pampa – Argentina, 2001.
- ✓ Lee Tae-Hwan, Suh Eui-Ho y Hwang Jun-Seok; Design and Implementation of a Web Site for Efficient Navigation based on Customer Purchasing Patterns; Korea Securities Computer Corporation, Seoul, South Korea, 2001.
- ✓ Linthicum, David; Enterprise Application Integration; Addison-Wesley Information Technology Series; 2000.
- ✓ Marcos E; MIDAS: A Methodology for Web Information Systems Development; Rey Juan Carlos University, Kybele Research Group, 2002.
- ✓ Microsoft .NET; 2003. [Version electrónica]
<http://www.microsoft.com/net/basics/>
- ✓ Montilva, Jonas; Sandia, Beatriz y Barrios, Judith; Developing Instructional Web Sites – A Software Engineering Approach; Universidad de Los Andes, Postgrado en Computación, 2001.
- ✓ Montilva, Jonas y Barrios Judith; A Component-Based Method for Developing Web Applications; Universidad de Los Andes, Facultad de Ingeniería, Escuela de Ingeniería de Sistemas, Departamento de Computación, Mérida, Venezuela, 2002.
- ✓ Montilva, Jonás y Barrios Judith; A Business Modelling Method for Information Systems Development; Universidad de Los Andes, Facultad de Ingeniería, Escuela de Ingeniería de Sistemas, Departamento de Computación, Mérida – Venezuela; 2003.
- ✓ Montilva, Jonas; Aplicando modelos de procesos de software al desarrollo de aplicaciones hipermedia. Universidad de Los Andes, Facultad de Ingeniería, 1996.
- ✓ Montilva Jonas; Factores Críticos de Éxito en el Desarrollo de Software; Universidad de Los Andes, Facultad de Ingeniería, Dpto. de Computación; Mérida – Venezuela. 2002.

- ✓ Montilva, Jonás; Ingeniería de Componentes e Integración de Software; Material del seminario 2002, Universidad de Los Andes, Facultad de Ingeniería, Postgrado en Computación, Mérida – Venezuela, 2002.
- ✓ Montilva, J., Barrios J. y Sandia B.; A Method Engineering Approach to Evaluate Instructional Products; Universidad de Los Andes, Facultad de Ingeniería, Departamento de Computación; Mérida – Venezuela, 2002.
- ✓ Montilva, J. y Barrios J.; The WATCH Method for Developing Web Applications; Universidad de Los Andes, Facultad de Ingeniería, Departamento de Computación; Mérida – Venezuela, 2003.
- ✓ Muller Pierre-Alain; Modelado de Objetos con UML; Ediciones Gestion 2000, S.A.; Barcelona, 1997.
- ✓ Musso, Roberto; Portales y Estrategia; Soluciona STI; 2003.
- ✓ Norfolk David; Understanding DSDM; Pc Network Advisor, Issue 67 Page 15.
- ✓ Oliveira, Angel; Metodología para Web Sites y Portales: Una Guía para el desarrollo exitoso; Consultor – Soluciones de Negocio con Contenido en Web; Caracas, Venezuela, Julio 2000.
- ✓ Olsina, Luis; Lafuente Guillermo y Pastor Oscar; Towards a Reusable Repository for Web Metrics; Journal of Web Engineering, Vol. 1, No. 1 (2002) 061-073.
- ✓ Paolos, Juan; Catalogo de Patrones de Diseño J2EE; Traducido del original de Sun Microsystem; España, 2003.
- ✓ Papazoglou, Mike y Yang, Jian; Design Methodology for Web Services and Business Processes; Infolab, Tilburg University; Netherlands, 2003.
- ✓ Pastor, Oscar y Fons, Joan; OOWS: Un Método de Producción de Software en Ambientes Web; Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, 2002.
- ✓ Pérez, Martín; Modelos de Desarrollo de Aplicaciones Distribuidas con J2EE; Universidad de A Coruña; 2003
- ✓ Petrie, Charles y Sahai Akhil; Business Processes on the Web; Internet Computin; IEEE; 2003.
- ✓ Pflieger, Shari; Software Engineering Theory and Practice; Prentice – Hall; Upper Saddle River, Ner Jersey; 1998.

- ✓ Press Meghan-Kiffer; E-Commerce: The Third Wave; from of book Enterprise E-Commerce; 2000.
- ✓ Pressman, Roger; Ingeniería de Software: Un enfoque práctico; quinta edición; Mc Graw Hill, 2002.
- ✓ Prince 2; Dynamic Systems Development Method Overview; BBs Ltd 2002.
- ✓ Proforma Corporation; E-Business Process Modelling; Preparing processes for the Web using ProVision, 2000.
- ✓ Ralité J., Deneckere R. y Rolland C.; Toward a Generic Model for Situational Method Engineering; CRI, University of Paris Sorbonne, 90 rue de Tolbiac, 75013 Paris, France; 2003.
- ✓ Retalis, Symeon y Avgeriou Paris, Modelling Web-Based Instructional Systems, Journal of information Technology Education, Volume 1 No. 1, 2002.
- ✓ Rolland C. y Pilmoh V.; Using Generic Method Chunk to Generate Process Models Fragments; University of Paris I Panthéon-Sorbonne, C.R.I.; Paris – Francia; 1995.
- ✓ Rolland C y Prakash N.; A proposal For Context-Specific Method Engineering; University Paris, 1996.
- ✓ Saeki Motoshi; Software Specification & Design Methods And Method Engineering; Dept. of Computer Science, Tokyo Institute of Technology, Ookayama; Tokyo – Japón, 1997.
- ✓ Samentiger, Johannes; Software Engineering with Reusable Components; Berlin, Springer – Verlag; 1997.
- ✓ Sanjiv, Augustine; Buiding Flexible, Large-Scale Application Architecture; C.C. Pace Systems, Inc; 2001.
- ✓ Serle y Chen; Modelo de Ciclo de Vida W y sus Metodologías Asociadas para el Desarrollo de Sitios Web Corporativos; 2001.
- ✓ Silva Darío y Mercerat Bárbara; Construyendo Aplicaciones Web con una Metodología de Diseño Orientada a Objetos; Revista Colombiana de Computación; Volumen 2, Numero 2; 2002.
- ✓ Sorm, Milan y Netrofová; The Theory of Web Information Systems; Institute of Informatics, Faculty of Business and Economics, Mendel University of Agriculture and Forestry in Brno, Czech Republic; 2003.

- ✓ Tolvamen Julia-Pekka, Rossi M. y Liu H.; Method Engineering: Current research Directions and Implications for Future Research; Department of Computer Science and Information Systems; University of Jyvasyla; Finlandia, 1996.
- ✓ Torrealba, Juan Carlos, Método para el análisis y diseño de sistemas de información web (SIW); Trabajo especial de grado para optar al título de magíster en Computación, Universidad de Los Andes, Facultad de Ingeniería, Postgrado en Computación, 1998.
- ✓ Tudor, Dot; Is DSDM appropriate for e-business projects?; Business Process The magazine for members of the National Computing Centre, Issue 15 June 2002.
- ✓ Valderrama, Ramiro; Antecedentes y justificación del Gobierno Electrónico; Diplomado de Gobierno Electrónico; Tecnológico de Monterrey; México, 2003.
- ✓ Weiss D. y Robert C.; Software Product-Line Engineering: A Family-Based Software Development Process. Addison-Wesley, 1999.
- ✓ Yang H. y Xue D.; Recent Research on Developing Web-based Manufacturing Systems: a review; Taylor & Francis Group; International Journal of Productions Research ISSN 0020-7543; 2003.
- ✓ Yang Jian; Web Service Componentization: Towards Service and Specialization; INFOLAB, Tilburg University, The Netherland, 2003.

Licencia Creative Commons:

Atribución - No Comercial - Compartir Igual 3.0 Venezuela
(CC BY-NC-SA 3.0 VE)