

**Universidad Central de Venezuela  
Facultad de Ciencias  
Escuela de Computación**

**Generación de Formularios Web y  
Estructuras de Procesamiento a partir  
de Información Dinámica con uso de XML**



Trabajo especial de Grado presentado ante la ilustre  
Universidad Central de Venezuela por:

**Br. Greisy Milena Dos Santos Cañizalez C.I. 15.573.303**  
**Br. Patricia Vélez Flores C.I. 15.222.802**

Para optar al título de  
Licenciado en Computación

Tutor: Prof. Sergio Rivas  
Co-tutor: Cristian Saavedra

Caracas, Marzo de 2006

---

**ACTA**

Quienes suscriben, miembros del Jurado designado por el Consejo de Escuela de Computación, para examinar el trabajo Especial de Grado presentado por los Bachilleres Greisy Milena Dos Santos Cañizalez C.I. 15.573.303 y Patricia Vélez Flores C.I. 15.222.802, con el título: "Generación de Formularios Web y Estructuras de Procesamiento a partir de Información Dinámica con uso de XML ", a los fines de optar al título de Licenciado en Computación, dejen constancia de lo siguiente:

Leído como fue, dicho trabajo por cada uno de los miembros del jurado, se fijó el día 15 de Marzo de 2006 a las 2:00 pm, para que sus autores lo defendieran en forma pública, lo que hicieron en el Auditorio de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondieron las preguntas formuladas. Finalizando la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presenta Acta, en Caracas a los 15 días del mes de Marzo del año dos mil seis dejándose también constancia de que actuaron como Coordinadores del jurado los tutores el profesor Sergio Rivas y el Licenciado Cristian Saavedra.

\_\_\_\_\_  
Prof. Sergio Rivas (Tutor)

\_\_\_\_\_  
Lic. Cristian Saavedra (Tutor)

\_\_\_\_\_  
Prof. Rossana Díaz (Jurado)

\_\_\_\_\_  
Prof. Andrés Sanoja (Jurado)

## AGRADECIMIENTOS

En primer lugar doy gracias a Dios Creador del Universo y dueño absoluto de mi existencia por haberme permitido realizar mis estudios y culminar este Trabajo Especial de Grado.

A lo más grande y bello que existe en mi vida MI HIJA ORIANA PATRICIA: mi ángel eres el motivo principal por el cuál trabajé y me esforcé para que esta meta fuera finalmente alcanzada.

A mis padres y a mi hermanita por todo el apoyo que me brindaron en la última etapa de esta tesis, gracias por todos los cuidados y atenciones que dieron a mi bebé mientras yo trabajaba.

A mi hermano, gracias por ser mi guía espiritual y por estar allí conmigo en los momentos en que más lo necesite.

A mi amigo, novio y compañero Orlando H. por su apoyo, comprensión y amor. Gracias por todos sus consejos y por darme ánimos.

A mis hermanas Shearly y Leidy, por estar allí en los mejores momentos de mi vida y en este en especial.

A Mafer, Luisito y Caro, mis amigos de siempre, porque a pesar de que no pudimos terminar juntos de una forma u otra siempre lo estamos.

Mientras formé parte del Grupo de Mantenimiento de Aplicaciones de Cantv varias personas participaron de cierto modo en la concepción de esta Tesis: Cristian Saavedra, gracias por darme la confianza de llevar a cabo este proyecto y por la ayuda prestada; Pedro Rodríguez, gracias por atendernos siempre cuando corríamos a tu puesto de trabajo para aclarar una duda. Al resto del grupo gracias por todo el apoyo.

Por último agradezco enormemente a mi tutor Prof. Sergio Rivas, por su amabilidad, sus consejos y por toda la ayuda que nos brindó a lo largo del desarrollo de este Trabajo Especial de Grado.

*Patricia Vélez Flores*

## AGRADECIMIENTOS

En este momento en el que siento una inmensa felicidad por haber conseguido una de mis mas grandes metas agradezco principalmente a mi madre por haberme dado la vida y además haberme inculcado los valores que hoy forman parte de mí y haberme enseñado a perseverar por los sueños anhelados. Sin su apoyo incondicional en todos mis triunfos y mis fracasos no sería lo que hoy soy.

A mis padres que de una u otra forma estuvieron allí para brindarme apoyo cuando mas lo necesité también les agradezco enormemente.

Agradezco a Nelbert, mi hermano quien siempre ha estado dispuesto a ayudarme y prestarme su valioso apoyo inclusive en los contratiempos presentados durante el desarrollo de este trabajo especial de grado.

También agradezco a mi tío Gustavo, quien en más de una oportunidad me brindó apoyo a lo largo de la elaboración del trabajo especial de grado.

Igualmente agradezco a aquellas personas que estuvieron en los momentos en los que necesite una palabra de aliento y un abrazo que me diera fuerzas para continuar en los momentos más difíciles. Esas personas a pesar de q no las nombre saben quienes son y de verdad que agradezco enormemente su apoyo en los momentos que más los necesité.

De igual forma agradezco a aquellas personas que de una u otra forma nos prestó su colaboración para la culminación exitosa de nuestro trabajo.

A María Fernanda también agradezco el apoyo que nos prestó en algunos detalles de diseño gráfico en nuestro trabajo especial de grado.

A mis tutores, Sergio y Cristian, agradezco de manera especial por su colaboración y apoyo pero sobre todo por su nivel de exigencia lo cual me hizo dar mucho más de mí y demostrar de esta forma mi potencial.

No podía faltar en mis agradecimientos, mi compañera Patricia Vélez, ya que a lo largo de este trabajo y cada una de las fases que pasamos nos brindamos mutuamente apoyo y palabras de ánimo, lo cual fue de gran ayuda para la culminación de nuestro trabajo especial de grado.

*Greisy M. Dos Santos C.*

<b>INDICE DE TABLAS Y FIGURAS .....</b>	<b>7</b>
<b>RESUMEN .....</b>	<b>11</b>
<b>INTRODUCCIÓN .....</b>	<b>12</b>
<b>CAPÍTULO 1: MARCO TEÓRICO .....</b>	<b>14</b>
1.1 Proceso de Desarrollo WebML .....	14
1.2 Lenguaje de Modelación Web (WebML) .....	20
1.2.1 Modelos en WebML .....	21
1.2.1.2 Modelo de Hipertexto .....	22
1.2.1.2.1 Modelo de Composición .....	22
1.2.1.3 Modelo de Presentación .....	27
1.2.1.4 Modelo de Usuario .....	27
<b>CAPÍTULO 2: MARCO APLICATIVO .....</b>	<b>30</b>
2.1 Antecedentes .....	30
2.1.1 Descripción del Problema .....	32
2.1.2 Planteamiento de la solución .....	32
2.1.3.1 Objetivo General .....	35
2.1.3.2 Objetivos Específicos .....	35
2.2 Método Aplicado .....	35
2.3 Aportes al Conocimiento .....	36
2.4 GForm. Aplicación Generadora de Formularios .....	36
FASE 1: Especificación de Requerimientos (GForm) .....	36
FASE 2: Diseño de Datos (GForm) .....	40
FASE 3: Diseño de Hipertexto (GForm) .....	41
FASE 4: Diseño de Arquitectura (GForm) .....	53
FASE 5: Implementación (GForm) .....	55
2.4.1 Caso de Estudio para la Aplicación Generadora de Formularios: GForm .....	66
FASE 6: Prueba y Evaluación (GForm) .....	82
2.5 GenSigs. Sistema para el manejo de Componentes Cantv .....	86
FASE 1: Especificación de Requerimientos (GenSigs) .....	86
FASE 2: Diseño de Datos (GenSigs) .....	88
FASE 3: Diseño de Hipertexto (GenSigs) .....	88

FASE 4: Diseño de Arquitectura .....	97
FASE 5: Implementación .....	97
2.5.1 Caso de Estudio del Sistema para manejo de Componentes Cantv: GenSigs .....	98
FASE 6: Prueba y Evaluación (GenSigs).....	104
<b>CONCLUSIONES, RECOMENDACIONES Y APORTES DE LA INVESTIGACIÓN...</b>	<b>107</b>
<b>BIBLIOGRAFIA Y REFERENCIAS.....</b>	<b>109</b>

## INDICE DE TABLAS Y FIGURAS

<i>Fig.1. Proceso de Desarrollo WebML.....</i>	<i>15</i>
<i>Fig.2. Ejemplo de Modelo de Estructura simple mostrando sus entidades y relaciones .....</i>	<i>22</i>
<i>Fig.3. Notación gráfica WebML para la Unidad de Datos .....</i>	<i>22</i>
<i>Fig.4. Notación gráfica WebML para Unidades MultiDatos .....</i>	<i>23</i>
<i>Fig.5. Notación gráfica WebML para Unidades Índice.....</i>	<i>23</i>
<i>Fig.6. Notación gráfica WebML para Unidades de Desplazamiento .....</i>	<i>23</i>
<i>Fig.7. Notación gráfica WebML para Unidades de Entrada.....</i>	<i>24</i>
<i>Fig.8. Notación gráfica WebML para representar Páginas .....</i>	<i>25</i>
<i>Fig.9. Vistas de sitios en WebML .....</i>	<i>26</i>
<i>Fig.10. Área del sitio Web de SIGS-WEB .....</i>	<i>26</i>
<i>Fig.11. Ejemplo de Diagrama de Hipertexto .....</i>	<i>27</i>
<i>Fig.12. Diagrama para representar la jerarquía de los grupos de usuarios.....</i>	<i>28</i>
<i>Fig.13. Integración de los modelos para el desarrollo de un sitio Web utilizando WebML .....</i>	<i>29</i>
<i>Fig.14. Mapa de las funcionalidades del sistema SIGS-WEB.....</i>	<i>31</i>
<i>Fig.15. Proceso de recolección de información y creación de las estructuras adecuadas .....</i>	<i>34</i>
<i>Tabla N°1 . Especificación del grupo de usuarios registrados para la aplicación generadora de formularios GForm .....</i>	<i>37</i>
<i>Fig.16. Diagrama de Casos de Uso para los procesos soportados por la aplicación generadora de formularios GForm .....</i>	<i>38</i>
<i>Fig.17. Modelo de Estructura de la Aplicación Generadora de Formularios .....</i>	<i>41</i>
<i>Fig.18. Página Index.....</i>	<i>42</i>
<i>Fig.19. Página Acceso_Denegado .....</i>	<i>42</i>
<i>Fig.20. Página Componentes .....</i>	<i>42</i>
<i>Fig.21. Páginas Nombre_Componente y Eliminar_Componente.....</i>	<i>43</i>
<i>Fig.22. Página Vista_Previa_Componente .....</i>	<i>43</i>
<i>Fig.23. Páginas Opciones_Generar y Generador_XML .....</i>	<i>44</i>
<i>Fig.24. Página Pestañas .....</i>	<i>44</i>

---

<i>Fig.25. Páginas Nombre_Pestana, Modificar_Pestana y Eliminar_Pestana.....</i>	<i>45</i>
<i>Fig.26. Página Categorías.....</i>	<i>45</i>
<i>Fig.27. Páginas Nombre_Categoría, Modificar_Categoría y Eliminar_Categoría... </i>	<i>46</i>
<i>Fig.28. Página Campos .....</i>	<i>46</i>
<i>Fig.29. Páginas Modificar_Campo y Eliminar_Campo.....</i>	<i>46</i>
<i>Fig.30. Página Agregar_Campo.....</i>	<i>47</i>
<i>Fig.31. Página Control_Text .....</i>	<i>47</i>
<i>Fig.32. Páginas Control_RadioButton y Control_Check .....</i>	<i>47</i>
<i>Fig.33. Página Control_DropDown.....</i>	<i>48</i>
<i>Fig.34. Páginas Agregar_Elemento/Item, Modificar_Elemento/Item y Eliminar_Elemento/Item.....</i>	<i>48</i>
<i>Fig.35. Vista de Sitio Público para la Aplicación Generadora de Formularios GForm .....</i>	<i>49</i>
<i>Fig.36. Área Componentes .....</i>	<i>50</i>
<i>Fig.37. Área Pestañas .....</i>	<i>51</i>
<i>Fig.38. Área Categorías .....</i>	<i>52</i>
<i>Fig.39. Área Campos.....</i>	<i>53</i>
<i>Fig.40. Diseño de Arquitectura para la aplicación generadora de formularios .....</i>	<i>55</i>
<i>Fig.41. String de conexión a la base de datos (global.asax).....</i>	<i>56</i>
<i>Fig.42. Diagrama de base de datos de la aplicación generadora de formularios... </i>	<i>56</i>
<i>Fig.43. Estructura del archivo XML construido por GForm .....</i>	<i>57</i>
<i>Fig.44.Ejemplo de XML para el componente Router (XML_Router.xml) .....</i>	<i>59</i>
<i>Fig.45. Carga del XML (Generador_Forms.aspx).....</i>	<i>60</i>
<i>Fig.46. Transformación del XML a través del XSLT (Generador_Forms.aspx).....</i>	<i>60</i>
<i>Fig.47. Instanciación de StreamReader para la lectura y escritura de archivos.....</i>	<i>61</i>
<i>Fig.48. Proceso de creación de formularios.....</i>	<i>62</i>
<i>Fig.49. Método Page Load del XSLT de inclusión (XSLT_Inclusión.xslt).....</i>	<i>63</i>
<i>Fig.50. Estructura del directorio de la aplicación GForm.....</i>	<i>64</i>
<i>Fig.51. Carpetas de la aplicación GForm.....</i>	<i>65</i>
<i>Fig.52. Instancia para recorrer el XML (Creacion_tables.aspx) .....</i>	<i>65</i>
<i>Fig.53. Generación del archivo de texto con los scripts SQL (Creacion_txt.aspx) </i>	<i>66</i>

<b>Tabla N°2. Información para el diseño del componente ATM.....</b>	<b>67</b>
<b>Fig.54. Página de acceso a la aplicación.....</b>	<b>68</b>
<b>Fig.55. Página para especificar el nombre del componente .....</b>	<b>69</b>
<b>Fig.56. Página para editar los elementos del componente ATM.....</b>	<b>69</b>
<b>Fig.57. Página para agregar pestañas .....</b>	<b>70</b>
<b>Fig.58. Página para incluir las categorías .....</b>	<b>71</b>
<b>Fig.59. Página para agregar campos .....</b>	<b>72</b>
<b>Fig.60. Página para agregar un control Drop Down.....</b>	<b>72</b>
<b>Fig.61. Página para agregar un control Text Box.....</b>	<b>73</b>
<b>Fig.62. Página para agregar un control Radio Button .....</b>	<b>73</b>
<b>Fig.63. Página para agregar un control check box .....</b>	<b>74</b>
<b>Fig.64. Listado de opciones a generar para un componente.....</b>	<b>75</b>
<b>Fig.65. Datos específicos para la base de datos.....</b>	<b>75</b>
<b>Fig.66. Tablas generadas para el componente ATM.....</b>	<b>76</b>
<b>Fig.67. Formulario de Inclusión generado para ATM.....</b>	<b>77</b>
<b>Fig.68. Estructura de la carpeta Proyecto de la aplicación GForm .....</b>	<b>78</b>
<b>Fig.69. Archivo con scripts para ATM (SQLs_ATM. txt) .....</b>	<b>79</b>
<b>Fig.70. XML generado para el componente ATM (XML_Generado_ATM.xml) .....</b>	<b>81</b>
<b>Fig.71. Diagrama de Casos de Uso para los procesos soportados por el sistema para el manejo de componentes Cantv .....</b>	<b>87</b>
<b>Fig.72. Página Home del Sistema para el manejo de componentes Cantv .....</b>	<b>89</b>
<b>Fig.73. Página de Consulta para la inclusión del Sistema para el manejo de componentes Cantv .....</b>	<b>89</b>
<b>Fig.74. Página de Inclusión de un Componente del Sistema para el manejo de componentes Cantv .....</b>	<b>90</b>
<b>Fig.75. Página de Consulta General del Sistema para el manejo de componentes Cantv.....</b>	<b>90</b>
<b>Fig.76. Página de Selección de un Componente del Sistema para el manejo de componentes Cantv .....</b>	<b>90</b>
<b>Fig.77. Página de Inclusión desde Plantilla y Modificación del Sistema para el manejo de componentes Cantv.....</b>	<b>91</b>
<b>Fig.78. Página de Consulta de Componente .....</b>	<b>91</b>

---

<b>Fig.79. Vista de usuario registrado .....</b>	<b>94</b>
<b>Fig.80. Área de Inclusión del Sistema para el manejo de componentes CANTV....</b>	<b>95</b>
<b>Fig.81. Área de Inclusión desde Plantilla y Modificación .....</b>	<b>96</b>
<b>Fig.82. Área de Consulta de un Componente.....</b>	<b>97</b>
<b>Fig.83. Página principal del sistema GenSigs.....</b>	<b>98</b>
<b>Fig.84. Página principal de consulta del sistema GenSigs .....</b>	<b>99</b>
<b>Fig.85. Página de solicitud de inclusión del sistema GenSigs .....</b>	<b>100</b>
<b>Fig.86. Página de consulta para inclusión desde plantilla del sistema GenSigs..</b>	<b>101</b>
<b>Fig.87. Página de solicitud de inclusión de un componente ya registrado en el sistema GenSigs .....</b>	<b>102</b>
<b>Fig.88. Página de modificación de un componente específico del sistema GenSigs .....</b>	<b>102</b>
<b>Fig.89. Página de consulta de un componente específico del sistema GenSigs .</b>	<b>103</b>

## RESUMEN

Actualmente en Cantv existe un sistema que permite llevar a cabo la tarea de gestión de componentes de red a través de un conjunto de formularios llamado SIGS-WEB. Este sistema solo permite gestionar routers y servidores, por lo que es evidente la necesidad de crear formularios para gestionar otros componentes de red utilizados en la empresa.

En respuesta a esta necesidad se planteó el desarrollo de una aplicación que permita automatizar el proceso de creación de los formularios y estructuras de procesamiento utilizados en la gestión de componentes de red.

Para el desarrollo de esta aplicación se utilizó el proceso WebML el cuál esta orientado fundamentalmente a aplicaciones de tipo Web. El lenguaje que lo conforma permite el soporte de todas las actividades y perspectivas de diseño de sitios Web de datos apoyado en especificaciones gráficas involucradas en un completo diseño de los procesos.

El ambiente de programación utilizado fue ASP.NET que presenta ventajas tales como simplicidad, escalabilidad y disponibilidad además de permitir herramientas superiores de desarrollo. Para la estructuración de la información fue usado XML que aporta potencia y flexibilidad a las aplicaciones basadas en el Web.

Como resultado tenemos la implementación de la aplicación GForm (Generadora de Formularios) que permite a los usuarios crear, modificar y eliminar el diseño de los mismos para un nuevo componente de red y a partir de este diseño generar los scripts en SQL para la creación de las tablas que soporten el almacenamiento de los datos o bien crear dichas tablas de manera automática en una base de datos específica. Para efectos de esta entrega solo se implementaron las tareas de inclusión, modificación y consulta de componentes de red del sistema SIGS-WEB.

## INTRODUCCIÓN

Con el crecimiento de Internet y del Web se han abierto infinidad de posibilidades en cuanto al acceso y uso de información desde cualquier parte del mundo. Con los avances en tecnología cada vez se demandan aplicaciones más rápidas, ligeras y robustas que permitan ser usadas sin importar el lugar u horario.

Las herramientas Web proporcionan grandes ventajas, entre los aspectos que hacen provechoso su uso se tiene la centralización de los datos y la facilidad de integración de datos de múltiples fuentes. Estas herramientas son utilizadas actualmente para muchas finalidades, ya que permiten la posibilidad de participar en un determinado proceso a distancia.

Muchas aplicaciones Web son creadas para un propósito inicial específico pero pueden requerir una posterior adaptación a nuevas necesidades, o se puede demandar cierta optimización de las mismas. Tal es el caso del sistema para la gestión y administración de componentes de red de la empresa Cantv llamado Sistema Integrado de Gestión de Servicios (SIGS-WEB), el cuál es utilizado por el personal del Centro de Operaciones de la Red para llevar a cabo tareas de gestión de equipos. En la actualidad sólo es posible administrar routers y servidores por lo que es preciso crear las interfaces (formularios y estructuras de procesamiento) que permitan la gestión de nuevos componentes.

A causa del diseño de programación que presenta la aplicación SIGS-WEB y de la complicada naturaleza de los equipos de red, la creación de nuevos formularios implica un pesado y arduo trabajo por parte del programador, por lo que se propuso la elaboración de una aplicación que permita automatizar este proceso y de este modo dar solución a este tipo de limitaciones.

Para llevar a cabo el desarrollo de la propuesta mencionada el proyecto fue estructurado de la siguiente manera:

**Capítulo 1. Marco Teórico:** contiene las bases teóricas necesarias para el desarrollo de la aplicación propuesta, iniciando con un amplio estudio del Proceso de Desarrollo WebML junto a cada una de las fases que lo componen. Posteriormente se presenta el Lenguaje de Modelación utilizado en este proceso y los modelos que comprende: Modelo de Estructura, Modelo de Hipertexto, Modelo de Composición y de Navegación; y los Modelos de Presentación y de Usuario.

**Capítulo 2. Marco Aplicativo:** en este capítulo se presenta el porqué de la elaboración del sistema, lo cuál incluye la descripción del problema y el planteamiento de la solución. Se refiere el Objetivo General y los Objetivos Específicos planteados para el desarrollo de la propuesta. Además se describe el Método Aplicado en donde se

presentan los modelos correspondientes a la aplicación de las fases necesarias del Lenguaje de Modelación WebML.

En la parte final se presentan las conclusiones y recomendaciones del Trabajo Especial de Grado.

## CAPÍTULO 1: MARCO TEÓRICO

En el siguiente capítulo se presenta un minucioso y preciso análisis de las bases teóricas que rigen el desarrollo del proyecto propuesto como solución al problema planteado.

La sección 1.1 contiene el estudio del Proceso de Desarrollo WebML como proceso de desarrollo de software; se refieren los distintos roles que pueden presentarse durante todo el proceso, así como también la descripción de las distintas fases que lo componen: Fase de Especificación de Requerimientos, Diseño de Datos, Diseño de Hipertexto, Diseño de Arquitectura, Implementación, Prueba y Evaluación, Mantenimiento y Evolución.

En la sección 1.2 se presenta el lenguaje utilizado en el proceso de desarrollo WebML, su definición y características principales. Posteriormente se explican cada uno de los modelos que se deben construir en cada fase del proceso de desarrollo junto con los elementos de notación correspondientes.

### 1.1 Proceso de Desarrollo WebML

WebML se define como *“lenguaje conceptual para el diseño de sitios Web en alto nivel”* [17]

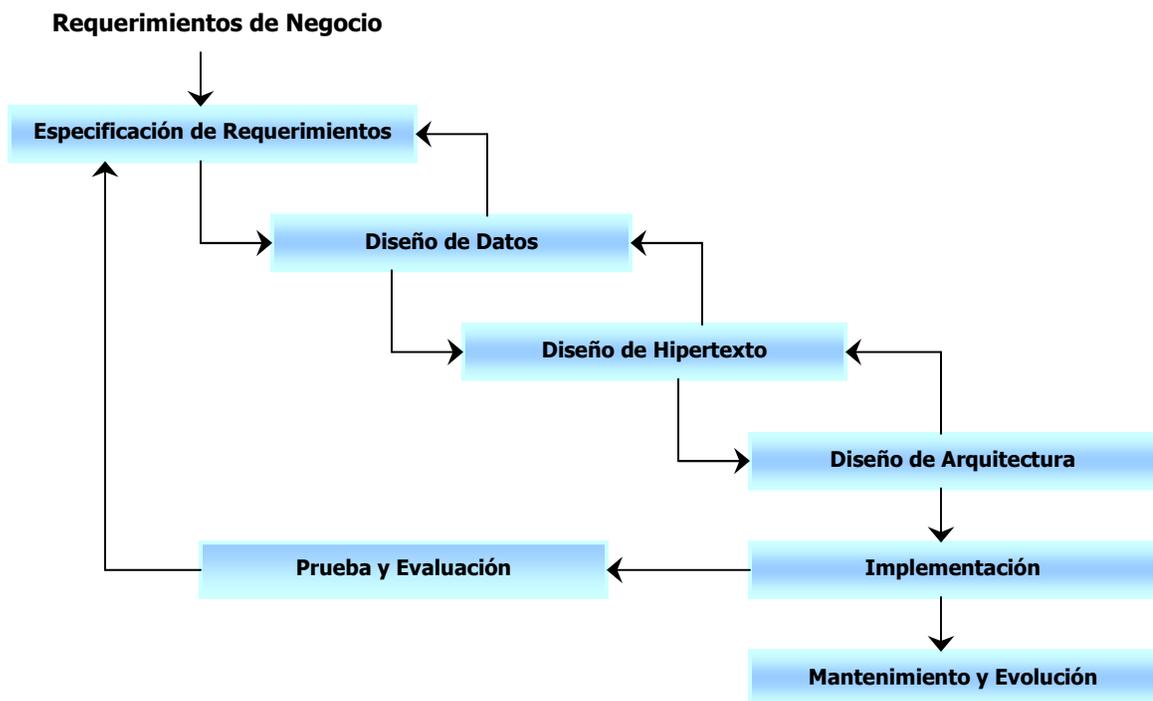
El proceso de desarrollo WebML permite especificar tanto el contenido estructural de una aplicación Web como la organización y presentación de su contenido. El desarrollo de aplicaciones Web es una actividad multifacética que envuelve diferentes actores con distintas metas. WebML asume un proceso de desarrollo dónde diferentes tipos de especialistas tienen distintos roles con habilidades complementarias y metas:

- 1) El **analista de la aplicación** recolecta los requerimientos del negocio y construye la especificación con los requisitos de la aplicación.
- 2) El **arquitecto de datos** enfoca los requisitos de la aplicación que tratan con los datos y produce el modelo de datos conceptual.
- 3) El **arquitecto de la aplicación** enfoca la parte de los requisitos de la aplicación que tiene que ver con los servicios a ser entregados y diseña el hipertexto de la aplicación especificando vistas de sitios construidas encima de los datos.
- 4) El **diseñador gráfico** concibe los estilos de presentación de las páginas basado en los requerimientos del negocio que tienen que ver con los estándares de la organización en cuánto a identidad visual y comunicación.

- 5) El **administrador del sitio y desarrollador** es el responsable del diseño de la arquitectura, implementación del sitio y del desarrollo y evolución de las vistas del sitio. En particular enfoca los requerimientos no funcionales del sistema.

Los roles descritos no son necesariamente desempeñados por diferentes personas, en aplicaciones simples puede ocurrir que los papeles convergen en la misma persona.

Las fases que se deben llevar a cabo para el desarrollo de aplicaciones con WebML deben aplicarse de una manera iterativa e incremental. Estas fases se muestran en la siguiente figura:



**Fig.1. Proceso de Desarrollo WebML**

Las primeras tres fases (*Especificación de Requerimientos*, *Diseño de Datos* y *Diseño de Hipertexto*) corresponden al análisis y diseño del sitio Web a crear, mientras que las demás hacen referencia a los aspectos físicos de la aplicación.

El proceso sufre varios ciclos, cada ciclo produciendo un prototipo o una versión parcial de la aplicación que permite dirigir pruebas y evaluaciones desde las primeras fases de desarrollo. Algunas de las fases del proceso pueden obviarse en el caso del desarrollo de una aplicación Web simple.

A continuación se explican en detalle cada una de estas fases:

---

- **Fase 1: Especificación de Requerimientos**

El responsable de la ejecución de esta fase es el *analista de la aplicación* el cuál debe realizar un levantamiento de información general acerca del dominio de la aplicación y de la solución. Para ello se deben tomar como entrada los Requerimientos del Negocio y se deben definir aspectos tales como: **Identificación de Usuarios, Requerimientos Funcionales, Requerimientos de Datos, Requerimientos de Personalización, Requerimientos de Dispositivos Específicos y Requerimientos No Funcionales**. A continuación se explican con detalle cada uno de estos aspectos:

- ✓ **Identificación de usuarios:** consiste en reconocer quiénes serán los usuarios de la aplicación y agruparlos según ciertas características comunes tales como metas y comportamiento. Cada grupo identificado es asociado a una vista de sitio distinta la cuál envuelve los contenidos y funciones necesarias para cumplir los requisitos de los usuarios del grupo.
- ✓ **Requerimientos Funcionales:** requiere la identificación de los procesos soportados por la aplicación. Un proceso se define como *“un conjunto cohesivo de actividades que deben ser realizadas por los usuarios que interactúan con la aplicación Web”* [4]. Una manera práctica de recoger los requerimientos funcionales es identificar y examinar varios escenarios representativos (o utilizar Casos de Uso en la terminología UML).
- ✓ **Requerimientos de Datos:** se deben identificar los datos que podrían ser manejados por la aplicación. El punto de partida para estos requerimientos es la investigación del “donde, cuando, y por quien” se produce y se consume el contenido de los datos. Las entidades para investigar son las organizaciones que suministran datos a la aplicación, y las organizaciones o usuarios que usan estos datos. Los procesos de negocio detrás de la aplicación representan otra fuente fundamental de conocimiento ya que su investigación permite que el analista identifique los datos intercambiados por los actores comerciales y producidos y consumidos por sus actividades.
- ✓ **Requerimientos de Personalización:** requiere la formalización de diferentes necesidades de contenido y servicio a los distintos individuos, basados en preferencias y derechos de acceso. La personalización de aplicaciones Web envuelve tres aspectos diferentes:
  1. Colección y almacenamiento de datos acerca de los usuarios.
  2. Análisis de los datos del usuario para inferir aquellas características que pueden manejar la entrega de contenido o servicios personalizados.

3. La construcción real de un hipertexto que incluya contenido y servicios personalizados con respecto a las características del usuario.

- ✓ **Requerimientos de Dispositivos Específicos:** consiste en la especificación de las demandas generadas para aplicaciones de múltiples dispositivos. En este escenario es importante identificar los dispositivos que puedan ser usados para acceder a la aplicación, agruparlos en familias con capacidades homogéneas y establecer las restricciones de presentación que las interfaces de la aplicación podrían presentar para cada clase de dispositivo.
- ✓ **Requerimientos No Funcionales:** incluyen “todos los demás requerimientos” que son relevantes para lograr las metas comerciales, pero no están específicamente relacionados a alguna funcionalidad del sistema. Estos requerimientos miden una gran variedad de aspectos, entre los cuáles se tienen: Usabilidad, Rendimiento, Disponibilidad, Escalabilidad, Seguridad y Mantenimiento.

#### ▪ **Fase 2: Diseño de Datos**

En esta fase el *arquitecto de datos* organiza la información principal de los objetos identificados durante la especificación de los requerimientos en un comprensivo y coherente modelo de datos conceptual denominado **Modelo de Estructura** el cuál se basa en el modelo entidad-relación conocido.

#### ▪ **Fase 3: Diseño de Hipertexto**

Es la fase en la cual se transforman los requisitos funcionales identificados durante la especificación de los requerimientos en una o más vistas de sitios incluyendo la información necesaria y los servicios de manipulación de datos. Este diseño opera a nivel conceptual y en el se identifican las páginas y unidades que conforman el hipertexto, uniéndolas y asignándolas a las principales entidades y relaciones del esquema de la estructura. El modelo que se construye en esta fase es el **Modelo de Hipertexto** y el responsable de su construcción es el *arquitecto de la aplicación*.

En esta fase pueden desarrollarse dos actividades de manera opcional:

- **Diseño de Presentación:** una vez que las páginas son suficientemente estables el arquitecto de estilo Web añade a cada página un estilo de presentación. En esta fase se construye el **Modelo de Presentación**.
- **Diseño de Usuario y Grupo:** el administrador Web define las características de los perfiles de usuario, basado en los requerimientos de personalización. Se mapean usuarios potenciales y grupos de usuario a los usuarios y grupos de WebML y posiblemente se crea una vista de sitio diferente para cada grupo. En esta fase se construye el **Modelo de Usuario**.

---

- **Fase 4: Diseño de la Arquitectura**

Consiste en la definición de los componentes de red, hardware y software que constituyen la arquitectura en que la aplicación entrega sus servicios a los usuarios. El responsable de este diseño es el *administrador del sitio y desarrollador*. La meta de este diseño es encontrar la mezcla de estos componentes que mejor se adapte a los requerimientos de la aplicación en términos de rendimiento, seguridad, disponibilidad y escalabilidad, para esto se estudian los siguientes aspectos:

- **Dimensiones del diseño de la arquitectura**

- ✓ **Metas del Diseño de la arquitectura:** se estudian los aspectos tales como rendimiento, escalabilidad, disponibilidad, mantenimiento del estado y seguridad que se persiguen con el desarrollo de la aplicación para asegurar la entrega de un nivel de servicio adecuado.
- ✓ **Requerimientos del Diseño de la arquitectura:** se toman en cuenta factores que pueden representar restricciones durante el desarrollo de la aplicación, tales como costos, complejidad, estándares e infraestructuras corporativas, etc.
- ✓ **Escenarios de despliegue de la arquitectura:** se determina el escenario a utilizar según el tipo de mantenimiento que se le prestará a la arquitectura diseñada. Los posibles escenarios son:
  - ✓ **Interno:** la arquitectura de la aplicación se encuentra físicamente dentro de la empresa y es mantenida por un departamento interno de la misma.
  - ✓ **Alojado:** la arquitectura de la aplicación es mantenida por un departamento interno de la empresa pero está instalada físicamente en un proveedor de servicio externo.
  - ✓ **Huésped:** la arquitectura de la aplicación se encuentra físicamente en un proveedor de servicio externo, el cual también la mantiene.

- **Diseño de la arquitectura del hardware y de la red**

Se estudia el escenario de arquitectura a utilizar considerando los 3 principales componentes: la aplicación en ejecución, el servidor de aplicaciones y el servidor Web. Asimismo, según el escenario especificado se describen los aspectos favorables que este ofrece en cuanto a rendimiento, escalabilidad, seguridad, entre otros.

- **Fase 5: Implementación**

Es la fase en la cuál se producen los módulos de software necesarios para transformar los datos y el diseño de hipertexto en una aplicación que corra en la arquitectura

---

seleccionada. Esta fase se subdivide en dos: **implementación de los datos** e **implementación del hipertexto**.

1. **Implementación de los datos:** asocia las entidades, atributos y relaciones del nivel conceptual a estructuras de datos físicas donde serán almacenados los datos.
2. **Implementación del Hipertexto:** se refiere a la producción de scripts de programas en el lenguaje seleccionado, a través de la traducción de las páginas y unidades del nivel conceptual especificadas en las vistas de sitios.

El responsable de la elaboración de esta fase es el *administrador del sitio* y *desarrollador*.

#### ▪ **Fase 6: Prueba y Evaluación**

En esta fase se verifica la conformidad de la aplicación implementada con respecto a los requerimientos funcionales y no funcionales. Los aspectos más relevantes en la prueba y evaluación de aplicaciones Web son los siguientes:

1. **Pruebas Funcionales:** el comportamiento de la aplicación se verifica con respecto a los requisitos funcionales. Esta comprobación puede verse afectada por actividades clásicas de prueba de módulo, prueba de integración y comprobación del sistema.
2. **Pruebas de Usabilidad:** los requerimientos no funcionales tales como facilidad de uso, comunicación efectiva y cumplimiento de las normas de uso deben verificarse contra las vistas del sitio producidas.
3. **Pruebas de Rendimiento:** el tiempo de respuesta de la aplicación debe ser evaluada en promedio a las condiciones de trabajo máximas. En caso de que el nivel de servicio sea inadecuado la arquitectura de despliegue debe ser supervisada y analizada para identificar y remover los cuellos de botella.

Esta fase es realizada por el *administrador del sitio* y *desarrollador*.

#### ▪ **Fase 7: Mantenimiento y Evolución**

Abarca todas las modificaciones efectuadas después de que la aplicación se ha desplegado en el ambiente de la producción. A diferencia de las demás fases de desarrollo, el mantenimiento y evolución se aplican a un sistema existente que incluye la aplicación que está corriendo y su documentación relacionada. Al igual que la fase anterior, el responsable del mantenimiento y evolución es el *administrador del sitio* y *desarrollador*.

---

## 1.2 Lenguaje de Modelación Web (WebML)

El lenguaje utilizado para el desarrollo del proceso anteriormente descrito es el Lenguaje de Modelación Web WebML. Este se define como un lenguaje conceptual orientado al soporte de todas las actividades y perspectivas de diseño de sitios Web de datos. Webml provee especificaciones gráficas involucradas en un completo diseño de los procesos, las cuales son asistidas por herramientas visuales de diseño.

Los principales objetivos de Webml son:

- Diseñar procesos que sean la expresión de la estructura de la aplicación Web con un alto nivel de descripción
- Proporcionar múltiples vistas del mismo contenido
- Separar la información del contenido desde su composición en la página, navegación y presentación la cual puede ser definida y desarrollada independientemente
- Almacenar la información recolectada durante el diseño de los procesos en un repositorio, permitiendo el acceso a la misma posteriormente, la cual puede ser utilizada durante el tiempo de vida de la aplicación y para generar páginas Web dinámicamente.

Webml puede ser usado conjuntamente con un diseño de procesos top-down o bottom-up. En el caso top-down los recursos de datos, son definidos conjuntamente con el sitio Web. En el caso bottom-up los recursos de datos existen previamente al sitio Web y pueden ser integrados ambos mediante un acceso online.

En el caso bottom-up, son aplicables los mismos objetivos y principios de diseño, pero el proceso de diseño debe incluir el acotamiento explícito de las estructuras de datos conceptuales descritas en WebML hacia las fuentes de los datos físicas existentes, cuyas estructuras y primitivas de acceso pueden tener influencia y pueden presentar restricciones en el diseño del sitio con WebML.

WebML permite a los diseñadores expresar las principales características de un sitio Web a alto nivel, sin especificar los detalles de arquitectura. Los conceptos de WebML se encuentran apoyados en una intuitiva representación gráfica la cuál puede ser fácilmente soportada por herramientas case y comunicada a los demás miembros del equipo de desarrollo como por ejemplo diseñadores gráficos [17].

## 1.2.1 Modelos en WebML

Durante todo el proceso de desarrollo WebML se construyen varios modelos:

- Modelo de Estructura
- Modelo de Hipertexto: compuesto a su vez por
  - ✓ Modelo de Composición
  - ✓ Modelo de Navegación
- Modelo de Presentación
- Modelo de Usuario

A continuación se describen cada uno de estos modelos:

### 1.2.1.1 Modelo de Estructura

Especifica la organización de los datos representando los objetos publicados en el sitio Web y la relación entre estos objetos. Es compatible con el Modelo entidad-relación y con el Diagrama de Clases de UML.

Este modelo está formado por los siguientes elementos:

- **Entidades:** son las estructuras básicas del modelo de estructura WebML. Cada entidad se asocia a un conjunto de objetos del mundo real cuyas propiedades se describen por medio de atributos. Se denotan por un rectángulo con el nombre de la entidad en el tope seguido de una lista de atributos.
- **Atributos:** en el modelo de WebML se utilizan los atributos para representar las propiedades del mundo real, y que se manejan en el sitio de Web. Un atributo tiene un tipo, escogido por el desarrollador en el momento de definir los tipos de WebML.
- **Relaciones:** las relaciones representan las conexiones semánticas entre las entidades. La relación más simple es la binaria, a través de la cuál se conectan dos entidades. Las relaciones que envuelven más de dos entidades son llamadas relaciones n-arias y también son soportadas por WebML.

En la figura 2 se muestra un ejemplo de un Modelo de Estructura simple:

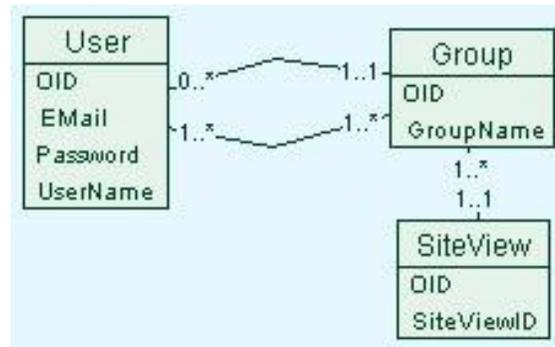


Fig.2. Ejemplo de Modelo de Estructura simple mostrando sus entidades y relaciones

### 1.2.1.2 Modelo de Hipertexto

El modelo de hipertexto especifica la Composición y Navegación del Sitio.

#### 1.2.1.2.1 Modelo de Composición

Muestra las **Unidades de Contenido** o elementos de información atómica que pueden aparecer en el sitio Web y las **Páginas** mediante las cuales se agrupa la información para ser entregada al usuario.

WebML soporta cinco tipos de unidades básicas:

- **Unidades de Datos:** muestran la información sobre un solo objeto y se definen para seleccionar información que proporcione una vista significativa del concepto dado en el modelo de estructura. La siguiente figura muestra la notación gráfica para representar unidades en WebML y un ejemplo de la unidad Componente para la aplicación a desarrollar.



Fig.3. Notación gráfica WebML para la Unidad de Datos

- **Unidades Multidatos:** representan diferentes instancias de una entidad o componente, repitiendo la presentación de varias unidades de datos iguales. Una especificación de una Unidad Multidatos consta de dos partes: el

*Contenedor* con las instancias que serán mostradas, las cuáles pueden referirse a una entidad, relación o componente y la *Unidad de Datos* utilizada para la presentación de cada instancia.



**Fig.4. Notación gráfica WebML para Unidades MultiDatos**

- **Unidades Índice:** representan múltiples instancias presentes en una entidad o componente, como por ejemplo una lista, mostrando cada objeto como una entrada en la lista. La figura 5 muestra la notación WebML para representar unidades índice y un ejemplo aplicado al sitio Web a desarrollar :



**Fig.5. Notación gráfica WebML para Unidades Índice**

- **Unidades de Desplazamiento:** proporcionan los comandos para realizar el desplazamiento de los objetos dentro del contenedor, bien sea de todas las instancias de una entidad o todos los objetos asociados a otro objeto a través de una relación. Son usadas normalmente junto con la unidad de datos que representa el elemento que se encuentra dentro del contenedor.



**Fig.6. Notación gráfica WebML para Unidades de Desplazamiento**

- **Unidades de Entrada:** soportan formularios basados en datos de entrada. Son usadas para obtener la entrada que será utilizada para localizar los objetos de una entidad cuyos atributos contengan una palabra clave dada o para proporcionar los parámetros de las operaciones tales como actualizaciones, login y otras operaciones externas.



**Fig.7. Notación gráfica WebML para Unidades de Entrada**

Las **Páginas** también forman parte del Modelo de Composición y surgen porque la granularidad de unidades puede ser muy fina para los requerimientos de la aplicación. Además WebML provee la notación de página para hacer la descripción progresiva del hipertexto y la organización física de las páginas. Una página se define como “...la abstracción de una región autónoma de la pantalla que se trata como una unidad de la interfaz atómica” [4]. Ejemplos de implementaciones concretas del concepto abstracto de páginas en lenguajes específicos pueden ser un archivo HTML, un marco o una portada WebML.

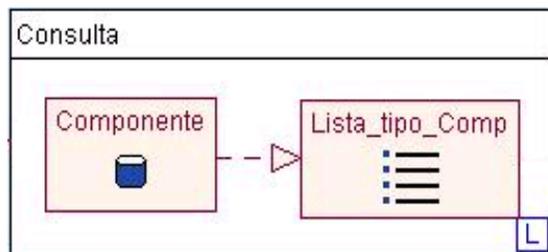
La relación entre unidades y páginas es la contención. Para el caso simple todas las páginas contienen exactamente una unidad, pero existirán casos en los que una página simple puede contener diversas unidades relacionadas.

Las páginas pueden ser de diversos tipos:

- ✓ **Home Page** es la página por defecto que se presenta al cargar el sitio o luego de que el usuario se autentifica en la aplicación. Debe ser única en la vista del sitio y se representa a través de una pequeña letra “H” dentro del icono de la página.
- ✓ **Default Page** es la página presentada por defecto cuando su área adjuntada es accedida. Debe ser única dentro del área y se representa a través de una letra “D” en el icono de la página.
- ✓ **Landmark Page** es alcanzable por las demás páginas o áreas dentro de un modulo adjuntado. Se especifica gráficamente a través de la letra “L”.

Una página se representa gráficamente con un rectángulo que encierra una o más unidades. En la siguiente figura (Fig.8) se muestra una página de tipo Landmark

llamada **Consulta** que contiene la unidad de datos *Componente* y la unidad índice *Lista\_tipo\_comp*.



**Fig.8. Notación gráfica WebML para representar Páginas**

### 1.2.1.2.2 Modelo de Navegación

Tiene como propósito especificar la forma como las unidades y páginas estarán conectadas para formar una estructura de hipertexto con significado y para definir el flujo de información necesaria para procesar las unidades y operaciones. Para alcanzar este propósito WebML cuenta con la noción de **Link** el cuál consiste en una conexión directa entre unidades o páginas [4].

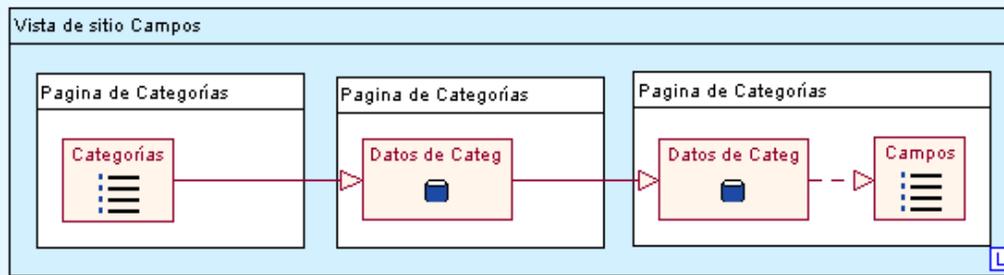
En WebML se definen dos tipos de links: **no-contextuales y contextuales**. Los primeros se establecen entre páginas y en ellos no se transmite ningún tipo de información de contexto, mientras que los links contextuales permiten la transmisión de información de contexto entre unidades fuente y destino.

### Organización del Hipertexto

Para organizar jerárquicamente el hipertexto se utilizan constructores modulares tales como Vistas de Sitio y Áreas.

- **Vistas de Sitio**

Consisten en construcciones modulares en las cuáles se empaqueta el hipertexto WebML dentro de una aplicación adjuntando las páginas enlazadas y las unidades. Se caracterizan por un nombre de usuario definido y contienen un conjunto de páginas y/o áreas. En la figura siguiente se muestra un ejemplo de este concepto:



**Fig.9. Vistas de sitios en WebML**

## ■ Áreas

Las **áreas** se utilizan para organizar el hipertexto y contienen páginas o sub-áreas utilizadas para organizar jerárquicamente las vistas de sitio. En la figura 10 se muestra una barra de navegación tomada del sitio Web del sistema SIGS-WEB la cuál muestra enlaces a diversas áreas del sitio.



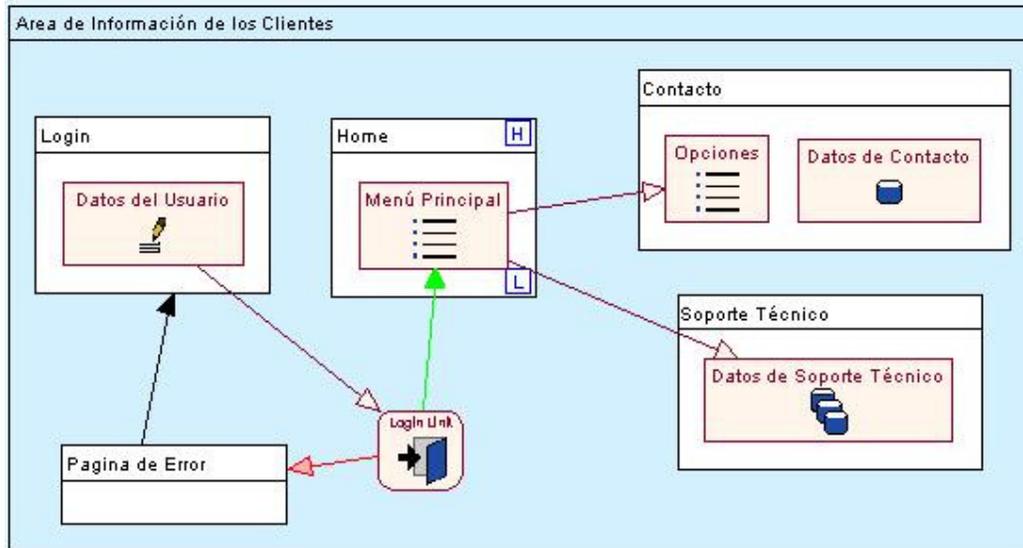
**Fig.10. Área del sitio Web de SIGS-WEB**

En la siguiente figura (Fig. 11) se muestra un diagrama de hipertexto completo de un Área de Información de los Clientes, dentro de la cuál se pueden observar el siguiente conjunto de páginas:

- **Página Login:** formada por una unidad de entrada para que el usuario introduzca el login y password.
- **Página Home:** constituida por una unidad índice. Es de tipo Home por ser la página que se presenta por defecto luego de que el usuario se autentica en la aplicación y Landmark ya que esta disponible a las demás páginas o áreas del sitio.
- **Página Contacto:** formada por una unidad índice y una unidad de datos.
- **Página Soporte Técnico:** constituida por una unidad multidados

Adicionalmente se presenta una unidad de operación llamada Login Unit provista por WebML para verificar la identidad de un usuario que accede a un sitio. Esta unidad toma dos parámetros (nombre de usuario y password) cuyos valores deben

ser pasados a través de un enlace y verifica su validez. Si dichos parámetros son inválidos se muestra al usuario una página de error pero en caso contrario se permite la entrada del usuario al sitio.



*Fig.11. Ejemplo de Diagrama de Hipertexto*

### 1.2.1.3 Modelo de Presentación

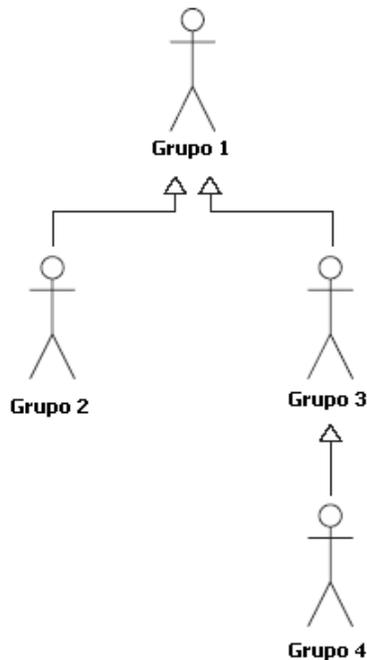
Describe la apariencia gráfica de las páginas. WebML no incluye un modelo específico para expresar la presentación del nivel conceptual, pero se acerca a los estándares más familiares gráficos y técnicos de la comunicación. La presentación es tarea ortogonal de definir el diseño de las páginas en una vista del sitio.

Desde que las especificaciones de WebML pueden representarse usando XML, la presentación es considerada como una transformación del documento enlazando la especificación WebML de una página en una página escrita en un lenguaje de implementación concreto como JSP o ASP.NET. Por consiguiente, la presentación se dirige en WebML anexándole una hoja de estilos XSL a las vistas del sitio, páginas, unidades y sub-elementos de la unidad.

### 1.2.1.4 Modelo de Usuario

El modelo de Usuario de WebML incluye como entidades predefinidas **Grupos y Usuarios**. Los grupos describen conjuntos de usuarios con características comunes, considerando la denotación particular de los usuarios. Ambas entidades pueden tener sus atributos internamente sub-estructurados, clasificados por medio de la herencia o relacionados a otras entidades a través de las relaciones [10].

Los grupos pueden ser organizados jerárquicamente y expresados a través del siguiente diagrama:



**Fig.12. Diagrama para representar la jerarquía de los grupos de usuarios**

A su vez, para cada uno de los grupos de usuarios identificados se deben especificar los siguientes elementos:

- **Nombre:** nombre del grupo.
- **Descripción:** descripción concisa del criterio de agrupamiento que define los miembros del grupo.
- **Datos del Perfil:** conjunto de atributos que caracterizan a los miembros del grupo y la manera en la cuál los datos del perfil son creados (preguntándole explícitamente al usuario o calculándolos implícitamente)
- **Super-grupo:** el grupo que generaliza las características comunes de varios sub-grupos (opcional).
- **Sub-grupo:** la lista de sub-grupos que expresan las propiedades especiales de selección de los miembros del grupo (opcional).
- **Casos de Uso relevantes:** la lista de los casos de uso en los cuáles el usuario participa.
- **Derechos de Acceso:** los datos esenciales que pueden ser accedidos o manejados por los usuarios del grupo. Son clasificados en dos sub-grupos: los

objetos que pueden accederse en modo “solo lectura” y los que pueden accederse en modo “manejo de contenido”.

Tanto los grupos como los usuarios sirven para dos propósitos distintos: **Control de Acceso** y **Personalización**. Las reglas de Control de acceso distinguen dos tipos de usuarios, los **registrados** y los **no registrados**. Los **usuarios no registrados** pertenecen al grupo predefinido, considerando que los **usuarios registrados** pueden pertenecer a uno o más grupos definidos por el administrador.

La personalización puede darse en vistas de sitio de usuarios registrados, donde el usuario identificado y el ID del grupo actual son automáticamente asociados en el momento en el que un usuario hace login y puede usarse para manejar el contenido de las unidades, a través del uso de selectores.

En la figura 13 se observa la integración de los modelos estudiados para el desarrollo de un sitio Web:

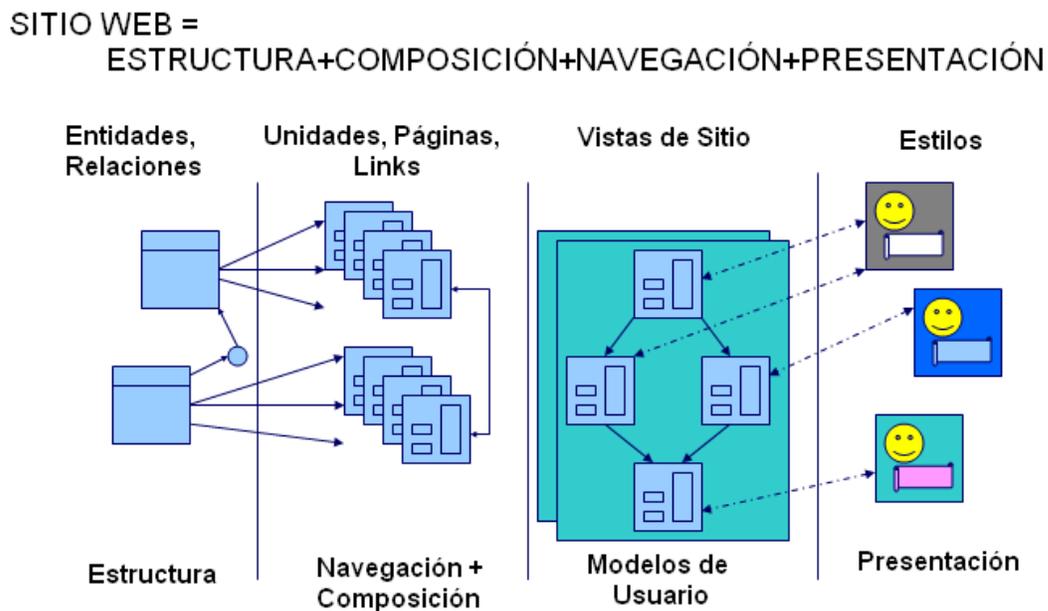


Fig.13. Integración de los modelos para el desarrollo de un sitio Web utilizando WebML

---

## CAPÍTULO 2: MARCO APLICATIVO

En el presente capítulo se especifican tanto los antecedentes a la propuesta planteada como el método aplicado para el desarrollo de la misma.

En la sección 2.1 se presentan los antecedentes del proyecto, así como la descripción del problema y el planteamiento de la solución. Adicionalmente se muestra el planteamiento del Objetivo General del proyecto así como también los Objetivos Específicos que conllevarán al cumplimiento del mismo.

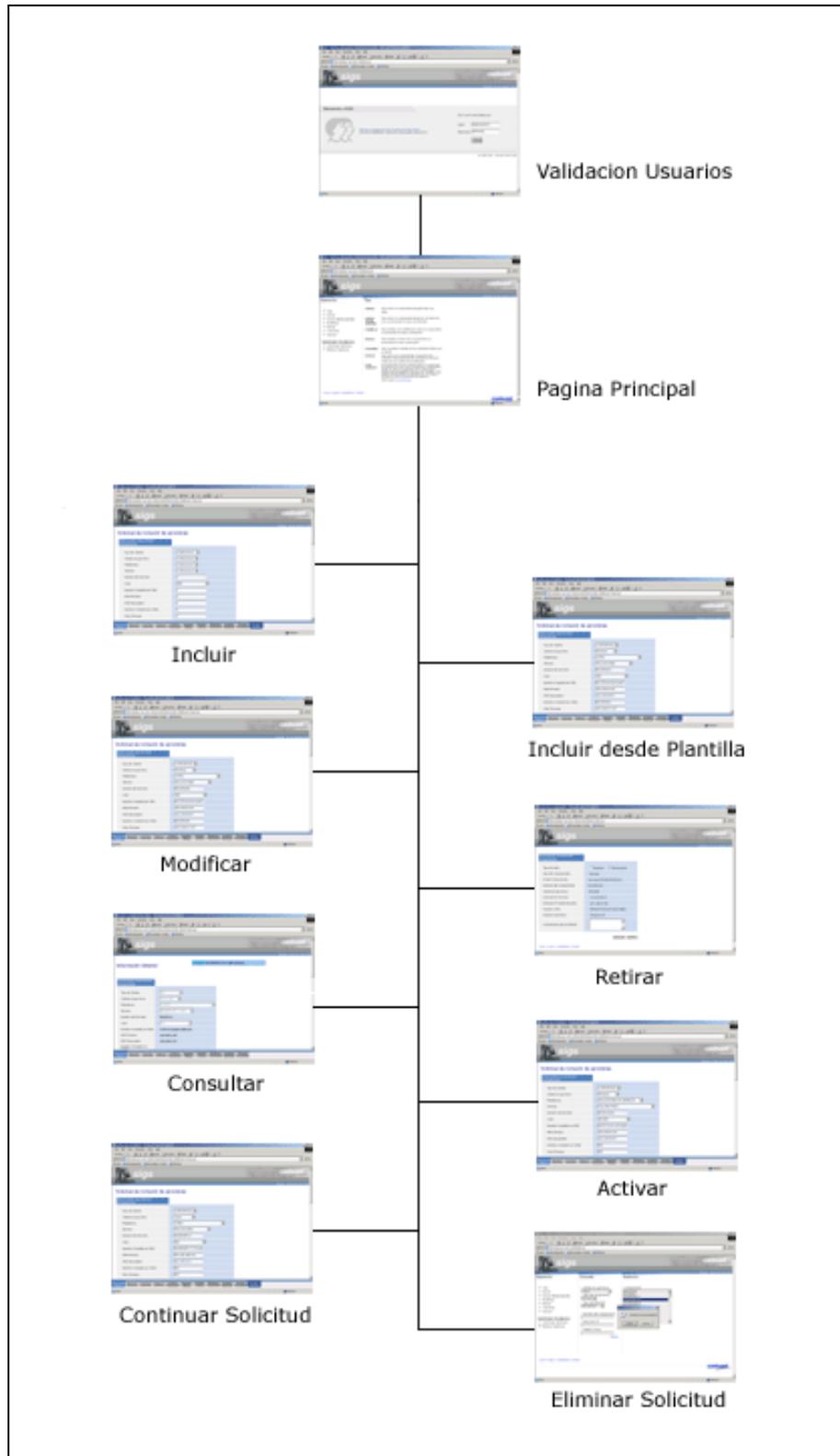
La sección 2.2 contiene las actividades realizadas durante cada una de las fases del proceso WebML que se consideraron necesarias.

### 2.1 Antecedentes

Hace algún tiempo en Cantv.net surgió la necesidad de desarrollar un sistema que permitiese a los clientes hacer los requerimientos de: inclusión, modificación y retiro de componentes de redes en la plataforma de monitoreo y cuya gestión se llevase a través de un flujo de trabajo para la aprobación de monitoreo de estos componentes. Adicionalmente, el almacenamiento de la información de la plataforma de monitoreo se debería hacer de manera actualizada, estructurada y estándar.

Como resultado de esta necesidad se tiene la implementación del sistema SIGS-WEB (Sistema Integrado de Gestión de Servicios versión WEB), que permite a los clientes hacer las siguientes solicitudes: inclusión, inclusión desde plantilla, modificación, retiros permanentes y temporales, activaciones y consultas de los distintos componentes. Además almacena toda la información del componente en una base de datos relacional. [12]

A continuación se presenta en forma de árbol las distintas funcionalidades de SIGS-WEB:



**Fig.14. Mapa de las funcionalidades del sistema SIGS-WEB**

### **2.1.1 Descripción del Problema**

En la actualidad el sistema SIGS-WEB sólo hace posible la gestión de routers y servidores, sin embargo, es evidente la necesidad de registrar y gestionar nuevos componentes de red para lo cuál se debe contar con los formularios destinados a tal fin. La creación de nuevas interfaces junto con las estructuras de procesamiento y de datos que dan soporte a las mismas (incluyendo tablas en la base de datos y formularios) representa un proceso arduo y laborioso como resultado del diseño de programación de la aplicación y de la gran cantidad de información que debe ser registrada para cada tipo de componente.

Por estas razones surge la necesidad de desarrollar una aplicación que facilite la creación de formularios para nuevos tipos de componentes, de tal manera que este proceso pueda ser automático y dinámico.

### **2.1.2 Planteamiento de la solución**

Debido a la ineficiencia en el proceso de creación de formularios para el registro de nuevos tipos de componentes de red de SIGS-WEB, se determinó como solución la creación de una aplicación Web que permita la generación dinámica de estos formularios a partir de información suministrada por el usuario.

El usuario debe especificar el diseño de los nuevos formularios que desea crear para la gestión de un nuevo componente de red, para lo cuál debe suministrar información detallada de los mismos. Esta información una vez recopilada es procesada y constituye la base para la creación de un documento XML cuya estructura lógica contiene elementos y atributos con datos específicos para los formularios [6]. A partir de este documento XML y a través de un conjunto de transformaciones del mismo se pueden generar los formularios, los scripts SQL para las tablas o bien crear dichas tablas de forma automática en una base de datos específica.

De manera detallada se puede especificar este proceso de recolección y transformación de datos en los siguientes pasos:

1. El usuario crea el diseño de los formularios para el nuevo componente de red, para lo cuál debe proporcionar toda la información necesaria respecto al mismo.
2. Con los datos del componente la aplicación generadora de formularios construye el documento XML.
3. Se une el XML construido con archivos XSLTs específicos, formando un objeto que debe ser procesado previamente para modificar y adaptar ciertas estructuras y posteriormente generar los formularios.

4. Utilizando un formulario Web que toma cierta información del XML originado se generan y ejecutan los scripts SQLs para la creación de las tablas en la base de datos señalada por el usuario.
5. De igual modo que en el paso anterior se realizan otras transformaciones del XML original a través de un formulario Web y se puede crear un archivo de texto que contiene los scripts en SQL para crear las tablas que soportarán el almacenamiento de los datos de los formularios en cualquier base de datos que desee el usuario.

En la siguiente figura se observa de manera gráfica el proceso anterior:

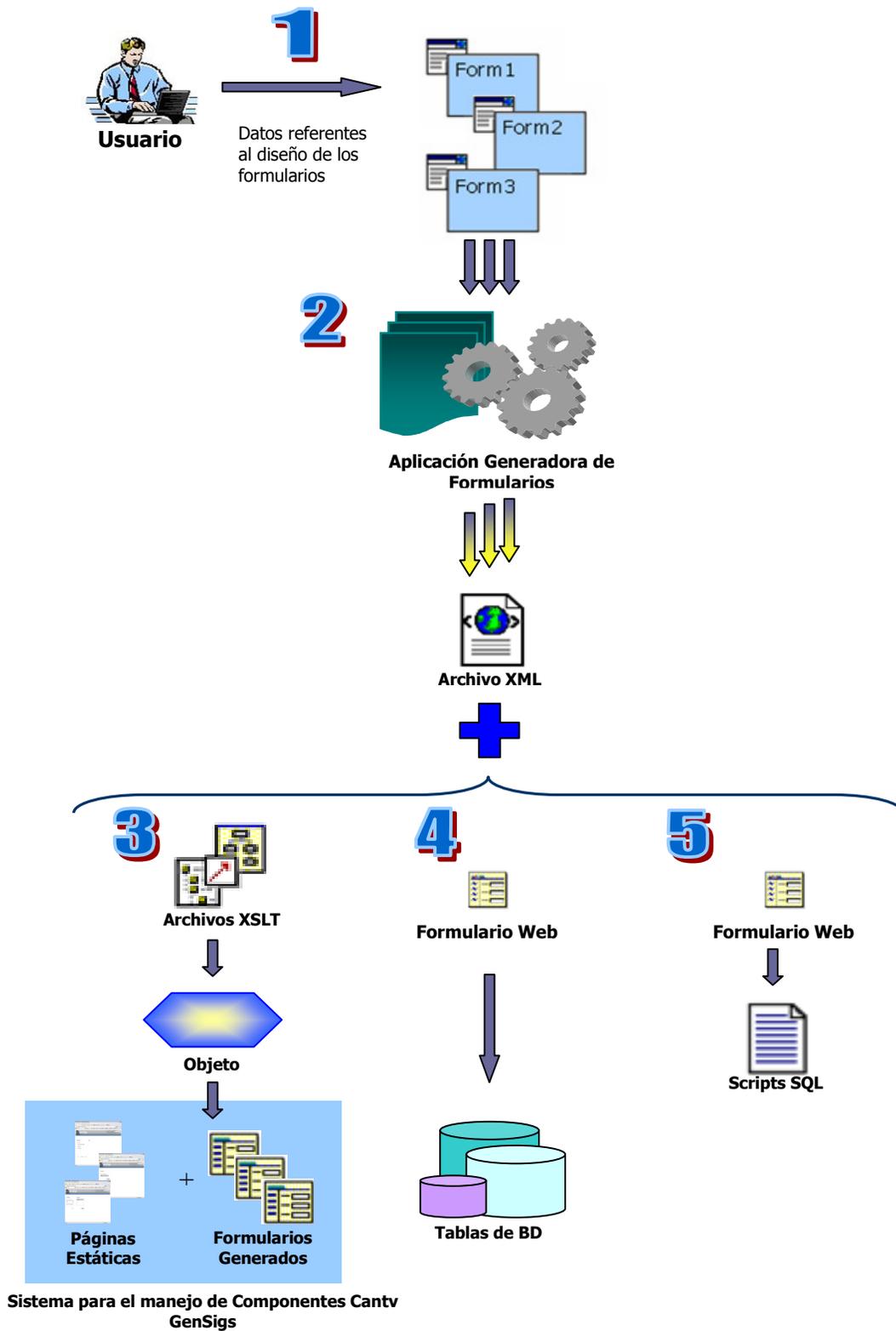


Fig.15. Proceso de recolección de información y creación de las estructuras adecuadas

### **2.1.3.1 Objetivo General**

Desarrollar e implantar una aplicación Web, que permita generar a partir de datos suministrados dinámicamente por el usuario, formularios y estructuras de procesamiento para el registro de información relativa a componentes que forman parte de la red interna de Cantv.

### **2.1.3.2 Objetivos Específicos**

En base al objetivo general surgen un conjunto de objetivos específicos los cuáles son:

- Utilizar el proceso de desarrollo WebML con las actividades que sean necesarias construyendo los modelos correspondientes a cada una de las fases.
- Generar un documento XML para representar la estructura de los elementos a crear y definir archivos XSLT que permitan transformar dicho documento y crear los formularios.
- Crear de modo automático los scripts en SQL de las tablas que soportarán el almacenamiento de los datos para los formularios a partir del XML generado.
- Crear de manera automática las tablas para el soporte de los formularios en la base de datos especificada por el usuario.
- Proponer un caso de estudio y analizarlo para verificar el cumplimiento de los objetivos.

## **2.2 Método Aplicado**

El proceso de desarrollo a utilizar para la creación de la aplicación es WebML.

Este proceso permite especificar los aspectos de composición y navegación de una aplicación Web a través de herramientas gráficas y formales incluidas en un completo proceso de diseño. En esencia, WebML consiste en un sencillo concepto visual para expresar el hipertexto como un conjunto de páginas, unidades y operaciones relacionadas entre sí.

WebML está constituido por un conjunto de fases que deben aplicarse de manera iterativa e incremental y que corresponden tanto al análisis y diseño del sitio Web a crear como a los aspectos físicos de la aplicación. [17]

WebML cuenta con su propio lenguaje de modelación, el cuál esta orientado al soporte de todas las actividades y perspectivas de diseño de sitios Web de datos.

Las fases de WebML serán aplicadas a la aplicación Generadora de Formularios GForm y al sistema resultante, Sistema para el manejo de Componentes Cantv GenSigs.

## 2.3 Aportes al Conocimiento

Este trabajo de grado contribuye a incrementar el conocimiento en el diseño de sitios Web haciendo uso del proceso de desarrollo de software WebML.

WebML aparte de que cuenta con su propio lenguaje de modelación provee una herramienta para el diseño de los diagramas que se construyen en las distintas fases del proceso, denominada WebRatio.

Adicionalmente se fundamentan los conocimientos en la utilización de XML, cuya tecnología permite compartir datos a través de varias aplicaciones y soportes; en XSLTs que permiten transformar documentos XML y en la creación de aplicaciones Web con ASP.NET.

En la siguiente sección se presenta un análisis detallado de la Aplicación Generadora de Formularios a construir, incluyendo el resultado de aplicar cada una de las fases del proceso WebML y un caso de estudio que permite verificar las funcionalidades del sistema.

## 2.4 GForm. Aplicación Generadora de Formularios

A través de esta aplicación el usuario podrá crear, modificar y eliminar el diseño de los formularios para el nuevo componente a través del suministro de datos concretos para el mismo. Una vez finalizada la creación del diseño de un nuevo componente se podrán generar los formularios, scripts SQL o bien crear las tablas en una base de datos específica señalada por el usuario.

A continuación se presenta el desarrollo de las fases de WebML para esta aplicación:

### **FASE 1: Especificación de Requerimientos (GForm)**

En esta fase se estudian los requerimientos de negocio en los que se basa el desarrollo de la aplicación así como toda la información referente al contexto técnico y organizacional donde debe operar la misma. El resultado de aplicar este análisis a la aplicación generadora de formularios es el siguiente:

#### **1. Identificación de Usuarios (GForm)**

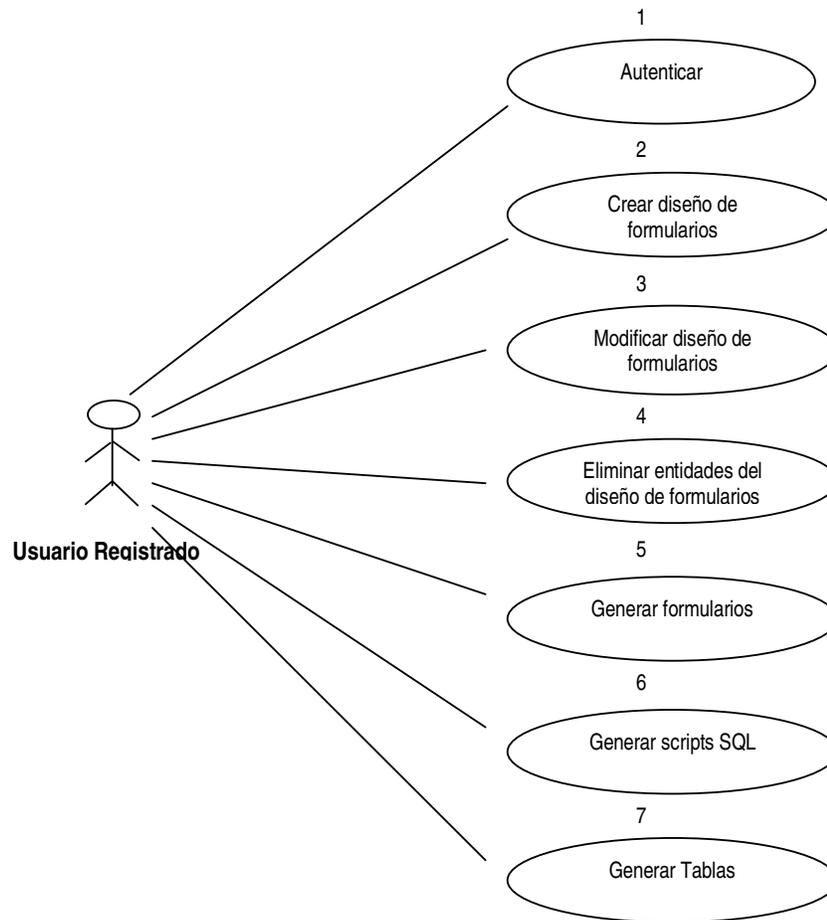
Los usuarios son de tipo Interno, es decir, usuarios registrados que necesitan autenticarse para ingresar y utilizar la aplicación. La siguiente tabla muestra la especificación de este grupo de usuarios:

<b>Nombre del Grupo</b>	Usuarios Registrados
<b>Descripción</b>	Usuarios interesados en utilizar la aplicación para crear un conjunto de formularios para un nuevo tipo de componente.
<b>Perfil de Datos</b>	Id, Nombre, Apellido, E-mail, Login, Password
<b>Casos de Uso</b>	Autenticar, Crear diseño de formularios, Modificar diseño de formularios, Eliminar entidades del diseño de formularios, Generar formularios, Generar Tablas, Generar scripts SQL.

**Tabla N°1 . Especificación del grupo de usuarios registrados para la aplicación generadora de formularios GForm**

## **2. Requerimientos Funcionales (GForm)**

El Diagrama de Casos de Uso que representa los procesos soportados por la aplicación generadora de formularios es el siguiente:



**Fig.16. Diagrama de Casos de Uso para los procesos soportados por la aplicación generadora de formularios GForm**

### **2.1 Especificación de los Casos de Uso (GForm)**

A continuación se describe con más detalle cada una de las funciones del sistema que el actor va a utilizar:

#### **1) Autenticar**

Permite a los usuarios de la aplicación ingresar sus datos para tener acceso a la misma.

#### **2) Crear Diseño de Formularios**

Esta funcionalidad engloba todo el proceso a través del cuál el usuario especifica de manera detallada el diseño que tendrán los nuevos formularios que desea crear asociado a un tipo de componente, incluyendo pestañas, categorías, campos y controles que forman parte de cada campo.

### **3) Modificar Diseño de Formularios**

Permite al usuario modificar los datos referentes al diseño de los formularios específicos para un componente.

### **4) Eliminar entidades del diseño de Formularios**

Permite al usuario eliminar entidades del diseño de un formulario específico para un componente.

### **5) Generar Formularios**

Una vez que el usuario haya creado el diseño de los formularios para el nuevo componente podrá generarlos a través de esta funcionalidad.

### **6) Generar Scripts SQL**

Permite generar en un archivo de texto las sentencias SQL para la creación de las tablas que soportan el almacenamiento de los datos para los nuevos formularios.

### **7) Generar Tablas**

Permite la generación automática de las tablas que soportan el almacenamiento de los datos de los formularios en la base de datos especificada por el usuario.

## ***4. Requerimientos de Personalización (GForm)***

Los usuarios de esta aplicación corresponden a un solo tipo, por lo que no se ofrecen opciones de personalización en cuanto a visualización, ni acceso a la información o a operaciones.

## ***5. Requerimientos de Dispositivos Específicos (GForm)***

La aplicación sólo podrá ser visualizada desde una PC convencional, por lo que no se toman en cuenta visualizaciones desde otro tipo de dispositivos.

## ***6. Requerimientos No Funcionales (GForm)***

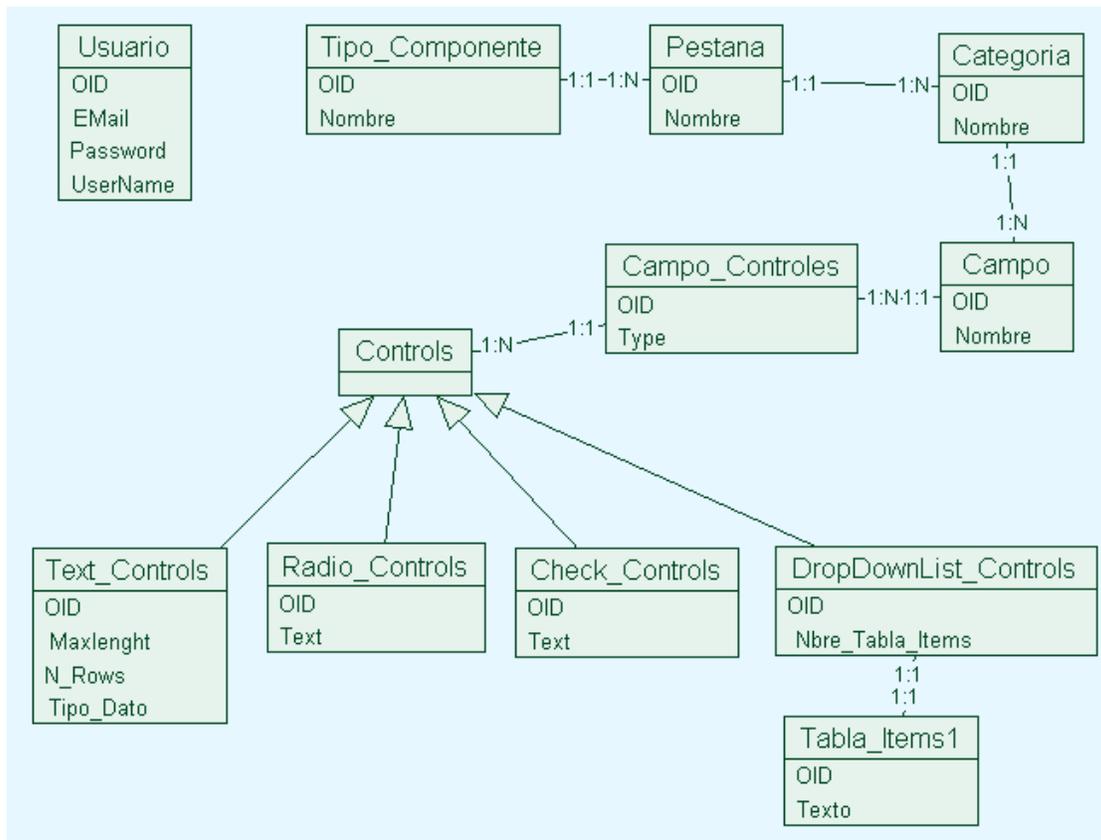
Como requerimientos adicionales a las funciones de la aplicación se consideran algunos aspectos como facilidad de uso, rendimiento y escalabilidad:

- Los requerimientos funcionales de la aplicación se implementarán a través de interfaces sencillas lo cuál proporcionará facilidad de uso y en donde los objetos tales como menús, enlaces, botones, etc., presentarán un uso coherente.

- La aplicación presentará un tiempo de respuesta aceptable a las peticiones realizadas por los usuarios de no más de 1 minuto por página, lo cuál obedece al rendimiento de la Intranet Corporativa.
- Se asegura la escalabilidad de la aplicación al poder adaptar nuevas funcionalidades a través de la adición de módulos que surjan de necesidades presentadas por los usuarios.

### **FASE 2: Diseño de Datos (GForm)**

De acuerdo al análisis respectivo de los requerimientos funcionales, se planteó una estructura de datos para soportar el procesamiento de las operaciones requeridas, y la misma se elaboró utilizando la herramienta Web Ratio. Web Ratio es un ambiente integrado para la especificación visual de aplicaciones que permite generar automáticamente las funcionalidades correspondientes a los diagramas WebML. [18] La estructura de datos construida es mostrada en la siguiente figura:



**Fig.17. Modelo de Estructura de la Aplicación Generadora de Formularios**

### FASE 3: Diseño de Hipertexto (GForm)

En esta fase se especifican las Vistas de Sitio a construir para representar los datos definidos en el diseño de datos y de este modo realizar la publicación y manipulación de los servicios identificados durante la fase de análisis. El modelo a construir en esta fase es el Modelo de Hipertexto, compuesto por el Modelo de Composición y el de Navegación, los cuales fueron contruidos al igual que el diseño de datos en la herramienta Web Ratio:

**Modelo de Composición:** en este modelo se muestran los nodos que conforman el hipertexto contenido en el sitio Web, especificado en páginas y unidades de contenido. En la Aplicación Generadora de Formularios se identificaron las siguientes páginas:

- **Página Index:** esta página muestra un formulario que permite el ingreso del login y el password del usuario. Contiene una unidad de entrada **Datos del usuario**.



Fig.18. Página Index

- **Página Acceso\_Denegado:** esta página será desplegada en caso de que el login o el password introducido por el usuario sea inválido.

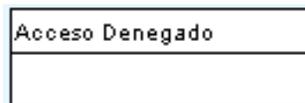


Fig.19. Página Acceso\_Denegado

- **Página Componentes:** Esta página es de tipo Home, debido a que es la página que se le muestra al usuario una vez que este introduce un login y un password válidos y es de tipo Default porque es la página por defecto del área donde se encuentra (Area Componentes). La misma contiene una unidad de datos **Tipo\_Componente** y una unidad índice **Opciones\_Componentes** que contiene las operaciones que se pueden realizar respecto al componente: Agregar Componente, Modificar Sub-elementos del Componente, Eliminar Componente, Vista previa de Componente y Generar.

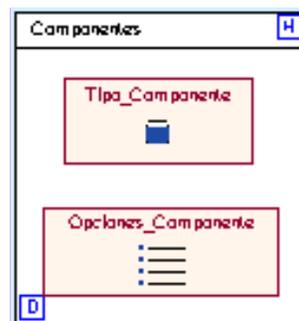


Fig.20. Página Componentes

- **Páginas *Nombre\_Componente* y *Eliminar\_Componente*:** Estas permiten agregar el nombre de un nuevo tipo de componente y eliminarlo respectivamente. Al eliminar el nombre de un tipo de componente se eliminan todos los elementos asociados a el.

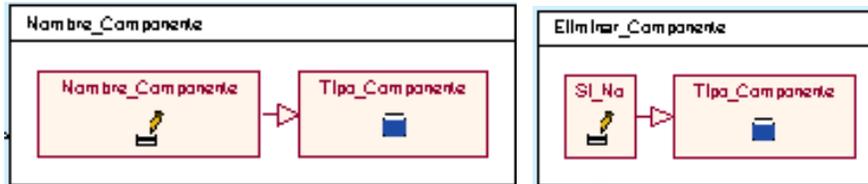


Fig.21. Páginas *Nombre\_Componente* y *Eliminar\_Componente*

- **Página *Vista\_Previa\_Componente*:** Esta página proporciona una vista previa del diseño de los formularios, recopilando la información de las unidades de datos *Tipo\_Componente*, *Pestana*, *Categoría* y de las respectivas unidades de datos de los controles asociados (*Text\_Controls*, *Radio\_Controls*, *Check\_Controls* y *DropDownList\_Controls*)

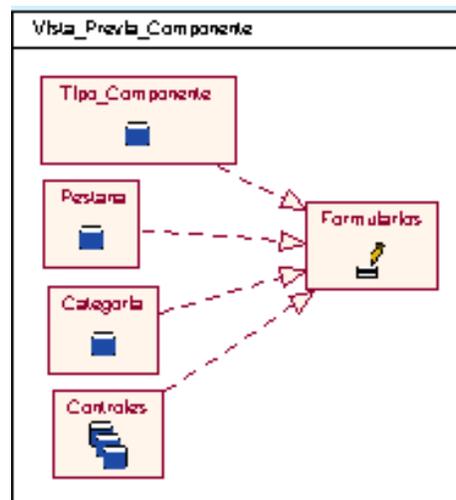


Fig.22. Página *Vista\_Previa\_Componente*

- **Páginas *Opciones\_Generar* y *Generador\_XML*:** En la página *Opciones\_Generar* se representa esencialmente la unidad índice *Opciones* la cual despliega una pequeña lista con los posibles productos que se pueden obtener con la aplicación, entre estas opciones tenemos: Formularios, Tablas en la BD y Scripts SQLs. Independientemente de cual sea la opción elegida se enlaza con la página *Generador\_XML*, la cual consulta las unidades de datos *Tipo\_Componente*, *Pestana*, *Categoría* y las respectivas unidades de datos de los controles asociados (*Text\_Controls*, *Radio\_Controls*, *Check\_Controls* y *DropDownList\_Controls*) y se genera un archivo XML con una estructura definida. A partir de este XML se realizan ciertas

transformaciones con archivos XSLT u otras transformaciones para generar los formularios, scripts o la creación de las tablas en la base de datos.

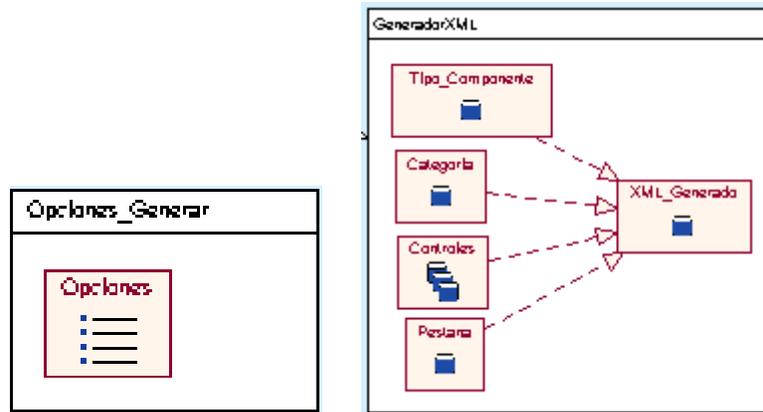


Fig.23. Páginas *Opciones\_Generar* y *Generador\_XML*

- **Página Pestañas:** Esta página es de tipo Default debido a que es la página por defecto del área donde se encuentra (Área Pestañas). La misma contiene una unidad de datos **Pestañas** y una unidad índice **Opciones\_Pestañas** que contiene las operaciones que se pueden realizar respecto a las pestañas: Agregar Pestaña, Modificar Nombre de Pestaña, Modificar Sub-componentes de la Pestaña, Eliminar Pestaña y Vista previa de Componente Actual.

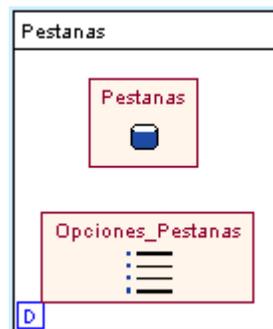


Fig.24. Página *Pestañas*

- **Páginas Nombre\_Pestana, Modificar\_Pestana y Eliminar\_Pestana:** Estas páginas permiten al usuario agregar el nombre de una nueva pestaña para el tipo de componente actual, modificarla y eliminarla respectivamente. Al eliminar el nombre de una pestaña se eliminan todos los elementos dependientes de ella.

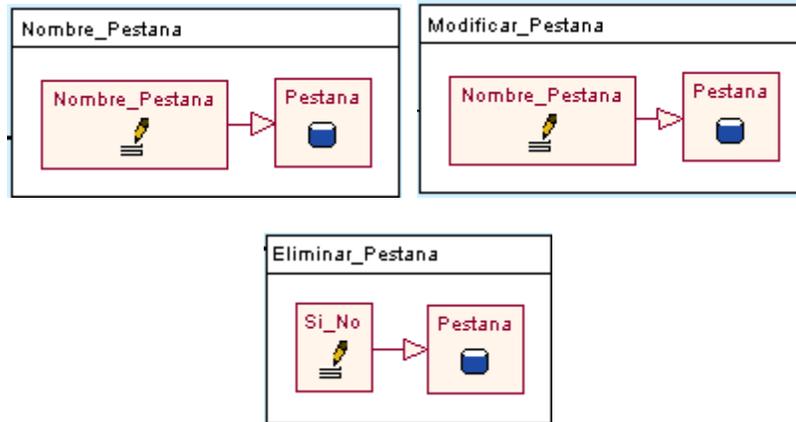


Fig.25. Páginas Nombre\_Pestana, Modificar\_Pestana y Eliminar\_Pestana

- **Página Categorías:** Esta página es de tipo Default debido a que es la página por defecto del área donde se encuentra (Área Categorías). Contiene una unidad de datos **Categorías** y una unidad índice **Opciones\_Categorías** que contiene las operaciones que se pueden realizar respecto a las Categorías: Agregar Categoría, Modificar Nombre de Categoría, Modificar Sub-componentes de la Categoría, Eliminar Categoría y Vista previa de Componente Actual.

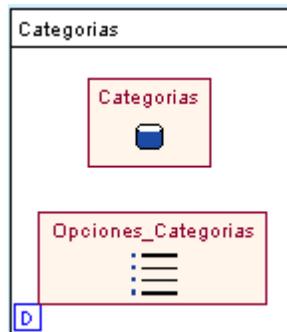


Fig.26. Página Categorías

- **Páginas Nombre\_Categoría, Modificar\_Categoría y Eliminar\_Categoría:** A través de estas páginas se permite al usuario agregar el nombre de una nueva categoría para el tipo de componente y pestaña actual, modificarla y eliminarla respectivamente. Al eliminar el nombre de una categoría se eliminan todos los elementos dependientes de ella.

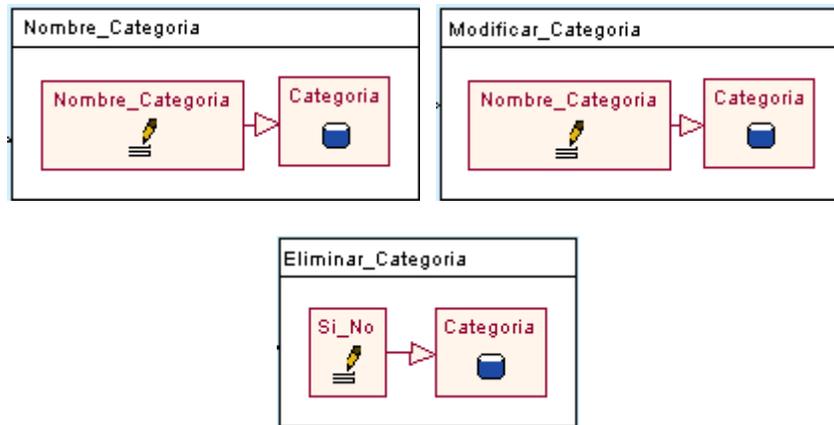


Fig.27. Páginas Nombre\_Categoría, Modificar\_Categoría y Eliminar\_Categoría

- Página Campos:** Esta página es de tipo Default debido a que es la página por defecto del área donde se encuentra (Área Campos). La misma contiene una unidad de datos **Campos** y una unidad índice **Opciones\_Campos** que contiene las operaciones que se pueden realizar respecto a los campos: Agregar Campo, Modificar Nombre de Campo, Modificar Sub-componentes del Campo, Eliminar Campo y Vista previa de Componente Actual.

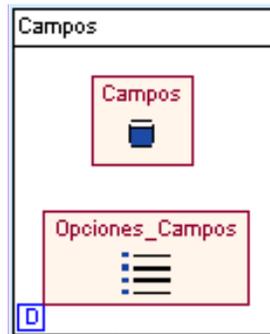


Fig.28. Página Campos

- Páginas Modificar\_Campo y Eliminar\_Campo:** Estas permiten modificar el nombre de un campo para el tipo de componente, pestaña y categoría actual, y eliminarlo respectivamente. Al eliminar el nombre de un campo se eliminan todos los elementos dependientes de el.

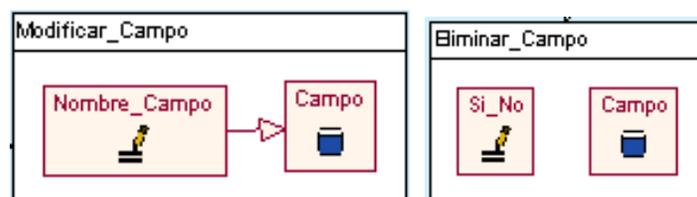


Fig.29. Páginas Modificar\_Campo y Eliminar\_Campo

- **Página Agregar\_Campo:** Esta página permite indicar el nombre del campo que se desea agregar y el control que se desea asociar al mismo, lo cual se representa a través de la unidad de entrada de datos **Nombre\_Campo** y la unidad índice **Tipos de Control**, dicha información se almacena en las unidades de datos representadas: **Campos** y **Campo\_Controles**

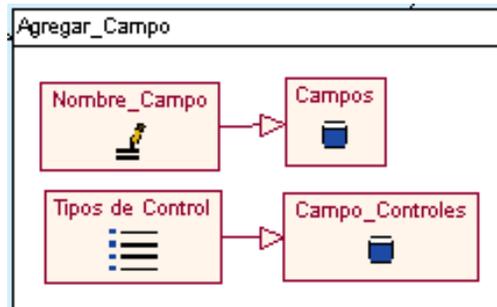


Fig.30. Página Agregar\_Campo

- **Página Control\_Text:** En esta página se representa una unidad de entrada de datos **Datos del Control**, la cual permite ingresar información propia del control y la misma es almacenada en la unidad de datos representada **Text\_Controls**.

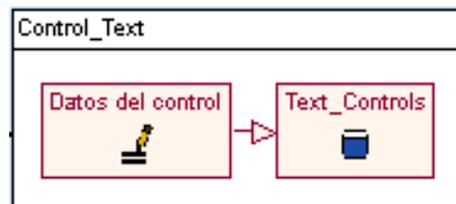


Fig.31. Página Control\_Text

- **Páginas Control\_RadioButton y Control\_Check:** Estas páginas contienen una unidad de datos **Radio\_Controls** ó **Check\_Controls** dependiendo del caso y una unidad índice **Opciones Elementos** que contiene las operaciones que se pueden realizar respecto a los elementos de estos controles: Agregar Elemento, Modificar Elemento, Eliminar Elemento y Vista previa de Componente Actual.

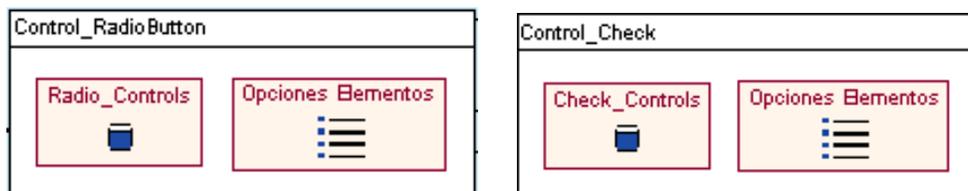


Fig.32. Páginas Control\_RadioButton y Control\_Check

- **Página Control\_DropDown:** Esta página contiene las unidades de datos **DropDownList\_Controls** y **Table\_Items**, las cuales representan las entidades donde se va a almacenar la información referente al control y los ítems que lo conforman. Se tiene en la página adicionalmente una unidad índice **Opciones\_Elementos** que contiene las operaciones que se pueden realizar respecto a los elementos de este control: Agregar Item, Modificar Item, Eliminar Item y Vista previa de Componente Actual.

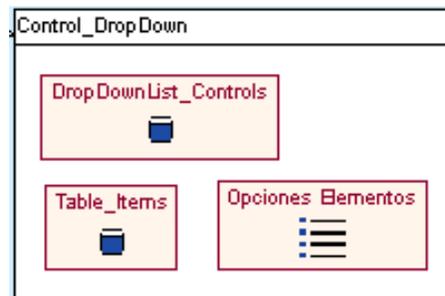


Fig.33. Página Control\_DropDown

- **Páginas Agregar\_Elemento/Item, Modificar\_Elemento/Item y Eliminar\_Elemento/Item:** Estas páginas permiten al usuario agregar un nuevo elemento a los controles Drop Down List, Radio Button y Check Box, así como también modificar y eliminar uno de sus elementos.

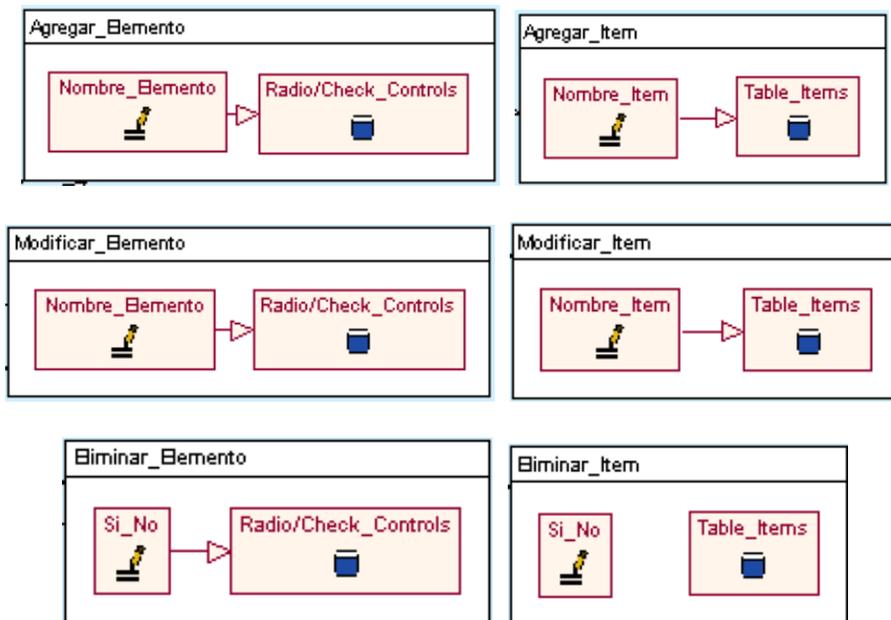
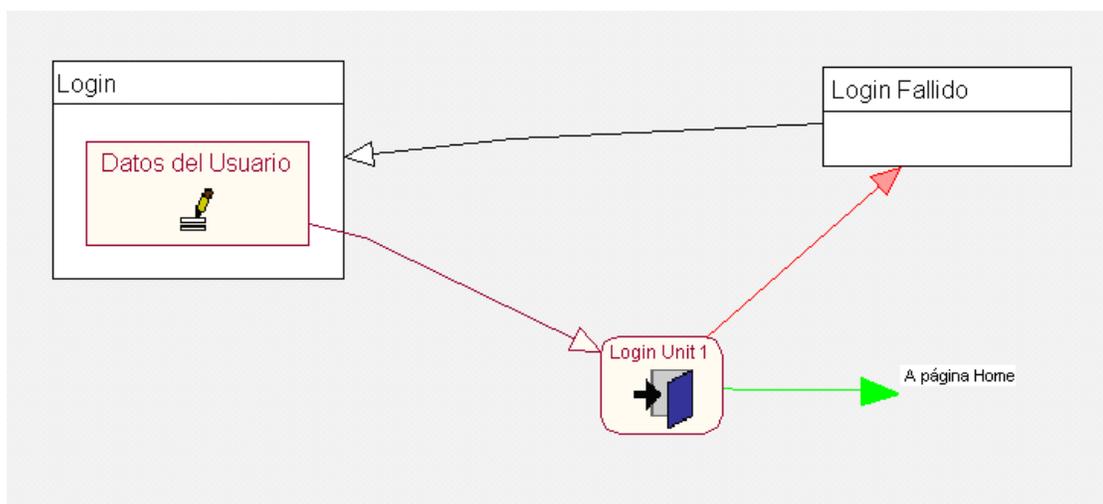


Fig.34. Páginas Agregar\_Elemento/Item, Modificar\_Elemento/Item y Eliminar\_Elemento/Item

**Modelo de Navegación:** a continuación se presentan las vistas de sitio incluyendo las unidades y operaciones y el flujo de información necesario para el procesamiento de las mismas:

**1) Vista de Sitio Público:** esta vista de sitio está formada por las páginas que pueden ser accedidas públicamente y la constituyen las páginas Login y Login Fallido. El usuario introduce los datos de Login y Password en la página Login y los mismos son enviados a la unidad de operación **Login Unit 1** la cuál realiza la validación de los mismos. En el caso de ser válidos se despliega la Página Home y en caso contrario la página de Login Fallido.



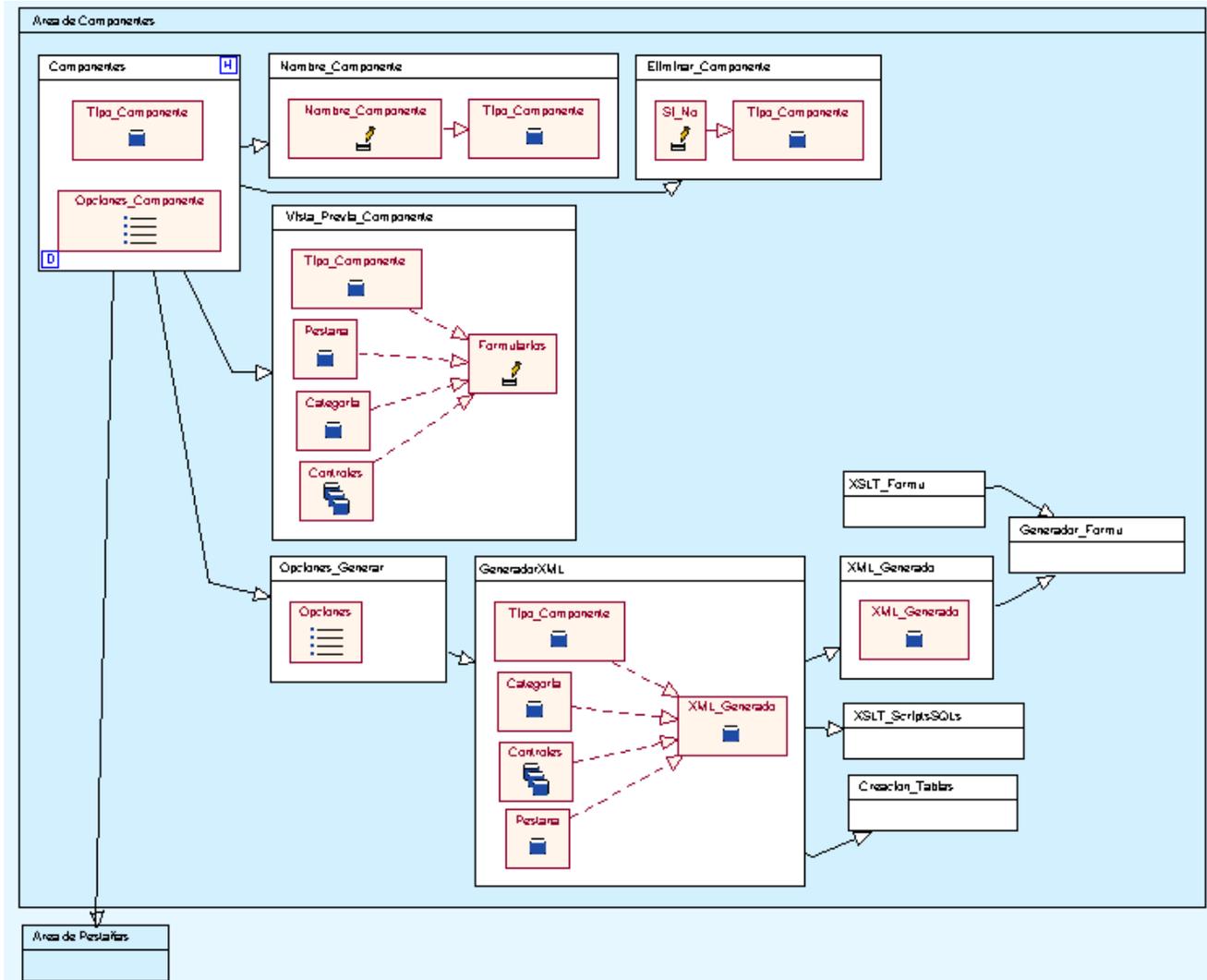
**Fig.35. Vista de Sitio Público para la Aplicación Generadora de Formularios GForm**

**2) Vista de Sitio Usuario Registrado:** la vista de sitio asociada a un usuario registrado una vez que introduce un login y un password válidos está dividida en cuatro áreas fundamentales:

- Área Componentes (1)
- Área Pestañas (2)
- Área Categorías (3)
- Área Campos (4)

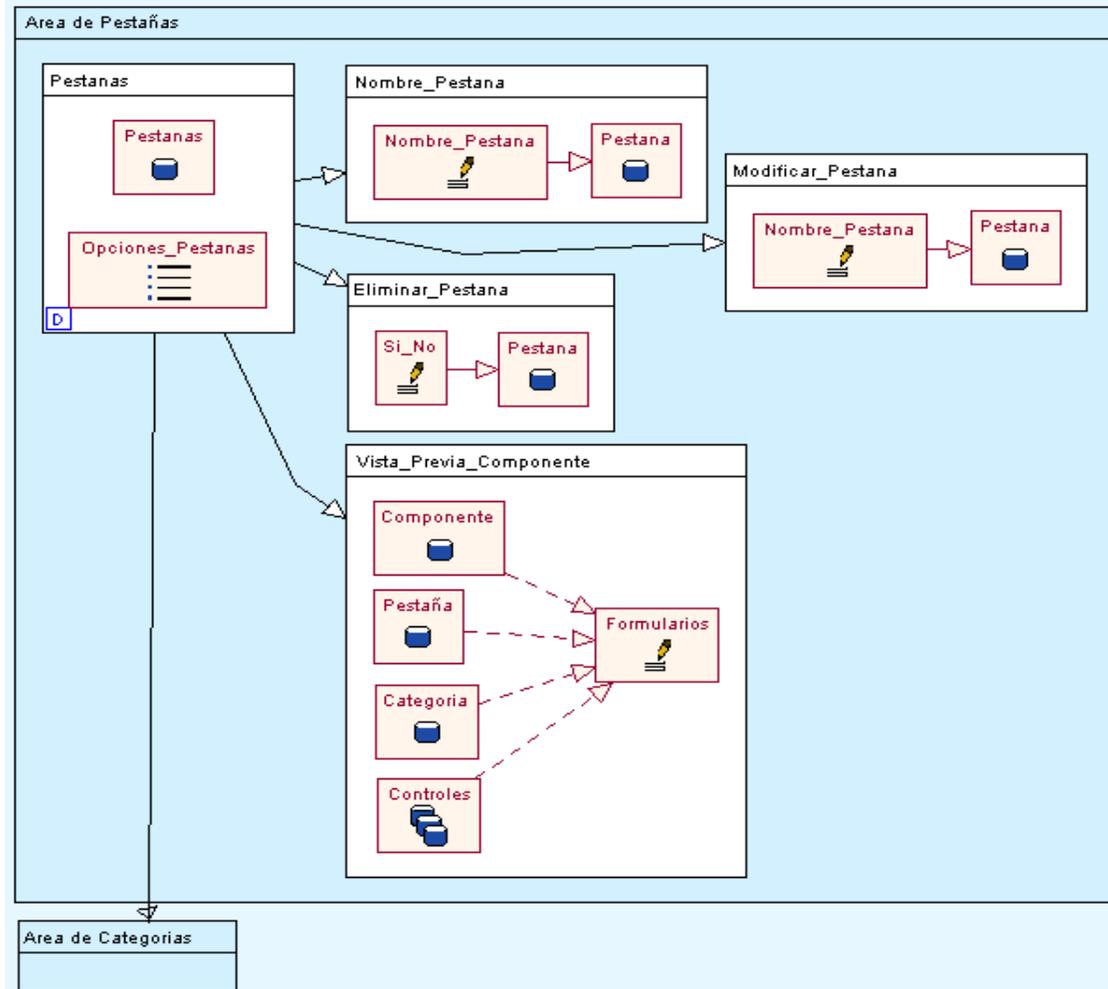
**2.1) Área Componentes:** a través de la cuál se puede organizar el hipertexto referente al componente, está formada por las páginas **Componentes**, **Nombre\_Componente**,

**Eliminar\_Componente, Vista\_Previa\_Componente, Opciones\_Generar, Generador XML, XML\_Generado, XSLT\_ScriptsSQL, Creación\_Tablas, Generador\_Formu y XSLT\_Form.**



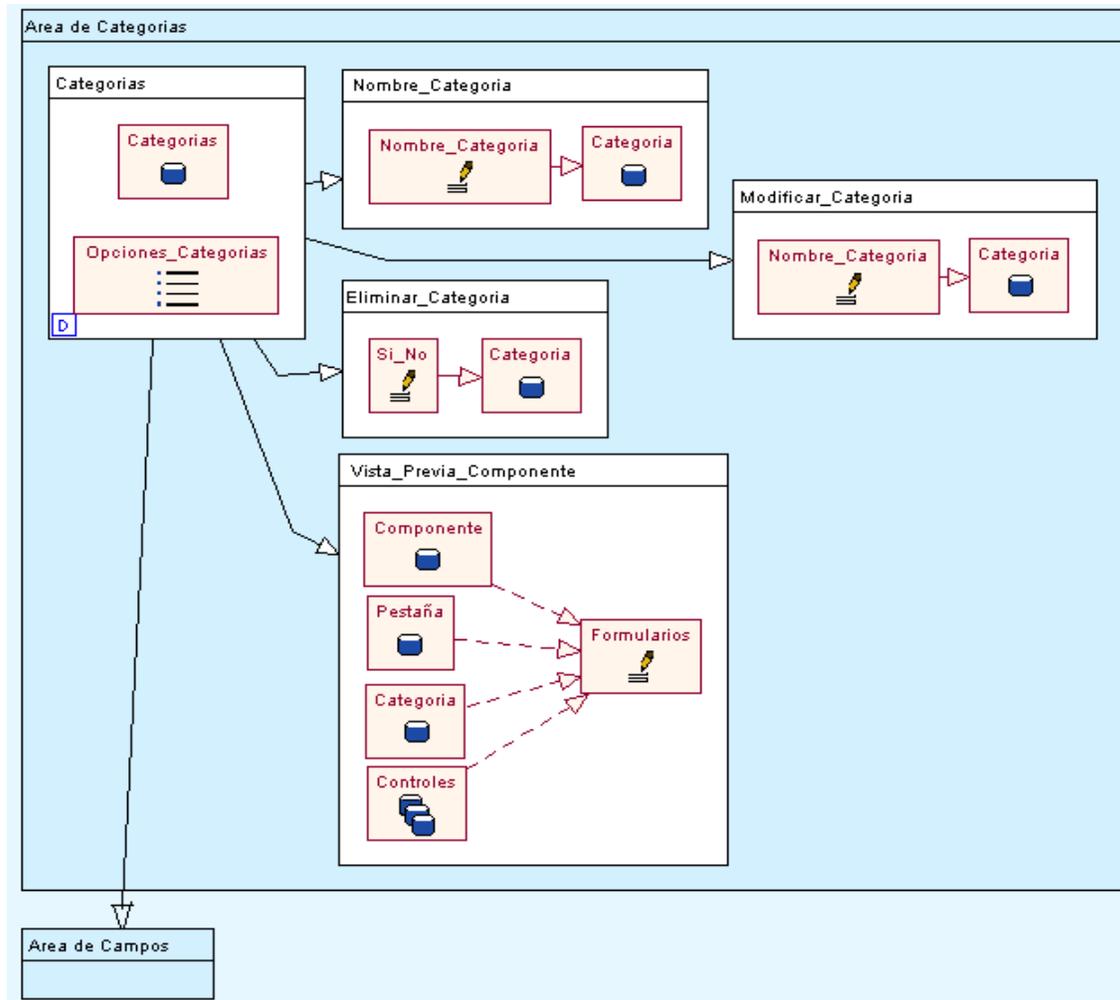
**Fig.36. Área Componentes**

**2.2) Área Pestañas:** formada por las páginas **Pestañas, Nombre\_Pestana, Eliminar\_Pestana, Vista\_Previa\_Componente y Modificar\_Pestana**, permite la organización del hipertexto referente a las Pestañas de los formularios.



**Fig.37. Área Pestañas**

**2.3) Área Categorías:** permite organizar el hipertexto referente a las Categorías de los formularios a generar; se encuentra formada por las páginas **Categorías**, **Nombre\_Categoría**, **Eliminar\_Categoría**, **Vista\_Preview\_Componente** y **Modificar\_Categoría**.



**Fig.38. Área Categorías**

**2.4) Área Campos:** esta área permite la organización de todas las páginas referentes a los campos de los formularios y los controles asociados a cada uno de ellos. Se encuentra formada por las páginas **Campos**, **Modificar\_campo**, **Agregar\_campo**, **Vista\_Previa\_Componente**, **Eliminar\_Campo**, **Control\_XX**, **Control\_Text**, **Control\_Radio Button**, **Control\_Check**, **Control\_DropDown**, **Agregar\_Elemento**, **Modificar\_Elemento**, **Eliminar\_Elemento**, **Agregar\_Item**, **Modificar\_Item**, **Eliminar\_Item**.

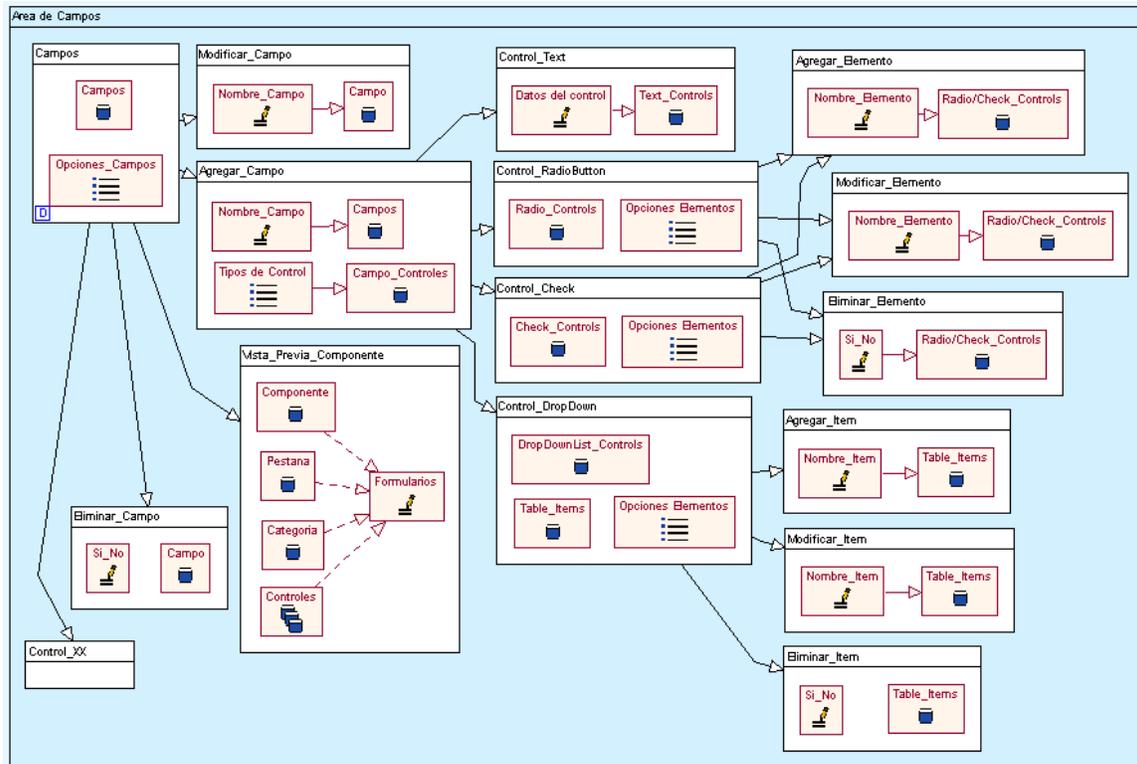


Fig.39. Área Campos

## FASE 4: Diseño de Arquitectura (GForm)

Se decidió alojar la aplicación en un servidor Web Microsoft IIS 5.0 Web Server con Sistema Operativo Windows 2000 Advanced Server con Service Pack 4.

El Sistema Manejador de Base de Datos utilizado es SQL Server 2000 el cual soporta acceso simultáneo a los datos por parte de gran cantidad de usuarios, y para lo cual se utilizará un servidor independiente.

### 4.1 Dimensiones del diseño de la arquitectura

- **Metas del Diseño de la Arquitectura**

Se persigue la obtención de una aplicación cuya arquitectura definida asegure el logro del nivel adecuado de servicio con respecto a:

- ✓ **Rendimiento**
- ✓ **Escalabilidad**
- ✓ **Disponibilidad**

- **Requerimientos del Diseño de la arquitectura**

- ✓ **Costos:** la creación de la aplicación implicó costos con respecto al tiempo empleado para su desarrollo. En cuanto a hardware y software no hubo implicaciones ya que se aprovechó el existente.
- ✓ **Complejidad:** no es necesario instalar una nueva aplicación en los equipos de los usuarios finales; solo es necesario tener disponible el Servidor IIS.
- ✓ **Estándares e infraestructuras corporativos:** en el desarrollo de la aplicación fue utilizado el entorno de programación Visual Studio.Net Versión 2003 específicamente la tecnología ASP.Net y Microsoft SQL Server 2000 como sistema manejador de base de datos, siguiendo los estándares de la organización.

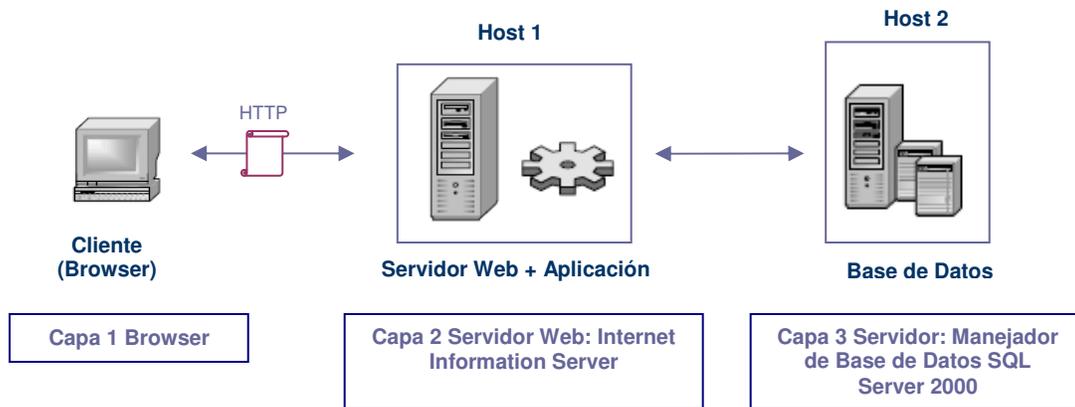
- **Escenarios de despliegue de la arquitectura**

El escenario que se aplica en este caso es el interno, ya que la aplicación se encuentra físicamente dentro de la empresa y es mantenida por un departamento interno de la misma.

#### **4.2 Diseño de la arquitectura del hardware y de la red**

- **Separación del servidor de la base de datos**

Para la implementación de la aplicación se utiliza un modelo de tres capas, donde se separa el servidor Web de la base de datos tal como lo muestra la siguiente figura:



**Fig.40. Diseño de Arquitectura para la aplicación generadora de formularios**

Esta configuración tiene impactos positivos sobre los siguientes aspectos:

- ✓ **Rendimiento:** el hecho de que estén separados físicamente el servidor Web y la base de datos contribuye notablemente al rendimiento de la aplicación, debido a que se cuenta con un procesador adicional para el manejo de los datos.
- ✓ **Escalabilidad:** al estar separados el servidor Web y el programa en ejecución de los datos, se contribuye a reducir el embotellamiento, ya que se aprovecha primero la capacidad máxima del servidor Web y luego la del servidor de base de datos, aumentando de esta forma el rendimiento total de procesamiento.
- ✓ **Disponibilidad:** aunque las fallas a nivel de la propia aplicación y a nivel de los datos son independientes, aún así se dificulta la disponibilidad ya que se tiene un solo punto tanto de la aplicación como de los datos. Para contribuir a la disponibilidad se tendría que utilizar la replicación.

### **FASE 5: Implementación (GForm)**

En esta fase se comienza la construcción de la aplicación la cuál se divide en Implementación de Datos e Implementación de Hipertexto.

#### **Implementación de Datos**

El modelo de estructura presentado en la fase 2 de Diseño de Datos es implementado a través de un conjunto de scripts en lenguaje SQL los cuáles se ejecutan en la base de datos correspondiente.

Para la conexión a la base de datos se utilizó la clase **SqlConnection** la cuál representa una conexión abierta con una base de datos de SQL Server. Una vez establecida esta conexión se utiliza un objeto de la clase **SqlCommand** para ejecutar las respectivas sentencias SQLs de creación de tablas, inserciones, consultas y modificaciones en la base de datos.

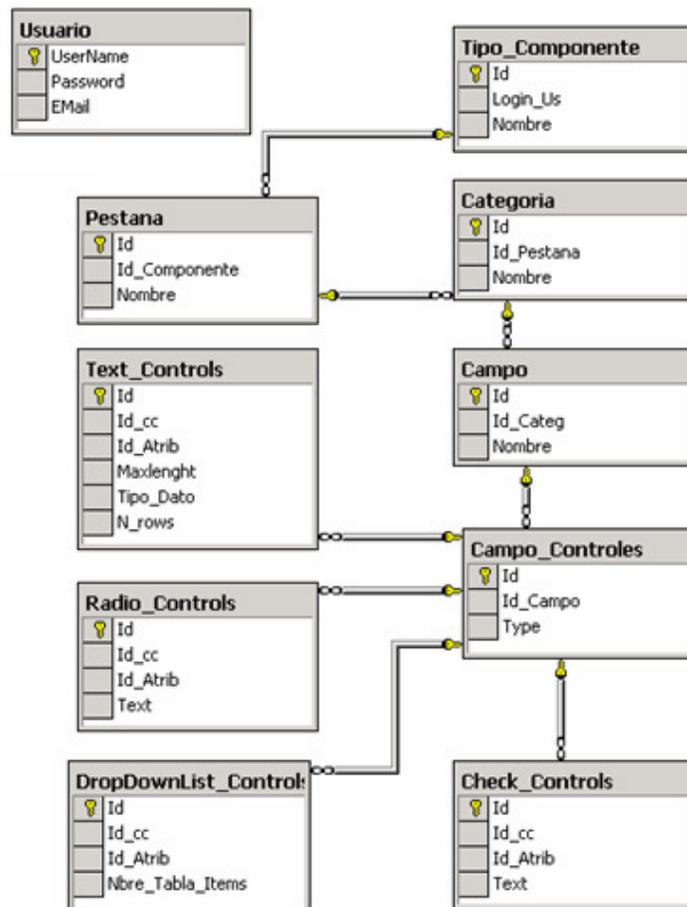
El string de conexión a la base de datos se introdujo en la sección **Application\_Start** del archivo `global.asax`, dicha sección se desencadena al inicio de la aplicación. Para esto se utilizó una variable de aplicación tal como se muestra:

```

Sub Application_Start(ByVal sender As Object, ByVal e As EventArgs)
    'String de Conexion para la BD de la aplicación Generadora de Formularios
    Application("String_Con1") = "data source=161.196.180.80;user id=gdsantos;"
    Application("String_Con1") = Application("String_Con1") & "password=milenagrei;database=Gen_Dynamic;"
    Application("String_Con1") = Application("String_Con1") & "persist security info=False"
End Sub
    
```

**Fig.41. String de conexión a la base de datos (global.asax)**

El diagrama de base de datos generado en el sistema manejador SQL Server se anexa a continuación:



**Fig.42. Diagrama de base de datos de la aplicación generadora de formularios**

## Implementación de Hipertexto

Para su implementación la aplicación fue dividida en dos módulos: el primero consta de todas las páginas a través de las cuáles el usuario introduce los datos referentes al diseño de los formularios y el segundo esta compuesto por la construcción del documento XML junto con las transformaciones aplicadas al mismo para generar el producto deseado.

Para la construcción del primer módulo se utilizaron solo formularios Web en Visual Basic.NET. Por su parte en el segundo módulo se utilizaron adicionalmente archivos XML y XSLT para realizar transformaciones.

El archivo XML construido presenta la siguiente estructura:

```
<?xml version="1.0" encoding="utf-8" ?>
<componente nombre="">
  <form nombre="" archivo="" id="" ide="">
    <categoria nombre="" abv="" ide="">
      <campo nombre="" abv="" tipo="" ide="" categ="">
        <control type="" id="" maxlength="" tipo_dato="" textmode="">
        </control>
      </campo>
    </categoria>
  </form>
</componente>
```

**Fig.43. Estructura del archivo XML construido por GForm**

A continuación un XML de ejemplo para un componente Router:

```
<?xml version="1.0" encoding="utf-8"?>
<componente nombre="Router">
  <form nombre="Informacion General" archivo="Incluir_Router_form1.aspx" id="form1" ide="1">
    <categoria nombre="Informacion General del Componente" abv="ROUTE_InfGendelCom" ide="1">
      <campo nombre="Tipo de Cliente" abv="TipdeCli" tipo="varchar(100)" ide="1" categ="1">
        <control type="asp:TextBox" id="txt_Tip_de_Cli" maxlength="50" tipo_dato="Caracter" textmode="MultiLine" />
      </campo>
      <campo nombre="Cliente al que sirve" abv="Clialquesir" tipo="varchar(100)" ide="2" categ="1">
        <control type="asp:TextBox" id="txt_Cli_al_que_sir" maxlength="50" tipo_dato="Caracter" textmode="MultiLine" />
      </campo>
      <campo nombre="Plataforma" abv="Pla" tipo="varchar(100)" ide="3" categ="1">
        <control type="asp:DropDownList" id="dropdown_Pla" table="TabItems_Pla_Rou_6" att="texto" read="myReader_Pla" />
      </campo>
      <campo nombre="Solicitudes" abv="Sol" tipo="varchar(100)" ide="4" categ="1" />
      <campo nombre="Nuevas solicitudes" abv="Nuesol" tipo="varchar(100)" ide="5" categ="1">
        <control type="asp:TextBox" id="txt_Nue_sol" maxlength="20" tipo_dato="Caracter" textmode="MultiLine" />
      </campo>
      <campo nombre="Direcciones" abv="Dir" tipo="varchar(100)" ide="ult" categ="1">
        <control type="asp:DropDownList" id="dropdown_Dir" table="TabItems_Dir_Rou_1" att="texto" read="myReader_Dir" />
      </campo>
    </categoria>
  </form>
  <form nombre="Informacion Fisica" archivo="Incluir_Router_form2.aspx" id="form2" ide="2">
    <categoria nombre="Direccion IP" abv="ROUTE_DirIP" ide="2">
      <campo nombre="Direccion IP" abv="DirIP" tipo="varchar(100)" ide="ult" categ="2">
        <control type="asp:CheckBox" id="check_Dir_IP231" Text="IP Publica" ide="1" />
        <control type="asp:CheckBox" id="check_Dir_IP232" Text="IP Privada" ide="ult" />
      </campo>
    </categoria>
  </form>
  <form nombre="Interfaces Loopback" archivo="Incluir_Router_form3.aspx" id="form3" ide="3">
    <categoria nombre="Protocolos de Enrutamiento" abv="ROUTE_ProdeEnr" ide="3">
      <campo nombre="RIP" abv="RIP" tipo="varchar(100)" ide="ult" categ="3">
        <control type="asp:RadioButton" id="radio_RIP241" GroupName="RadioGroup24" Text="SI" ide="1" />
        <control type="asp:RadioButton" id="radio_RIP242" GroupName="RadioGroup24" Text="NO" ide="ult" />
      </campo>
    </categoria>
    <categoria nombre="Access List" abv="ROUTE_AccLis" ide="4">
      <campo nombre="Lista de Acceso" abv="LisdeAcc" tipo="varchar(100)" ide="ult" categ="4">
        <control type="asp:TextBox" id="txt_Lis_de_Acc" maxlength="30" tipo_dato="Numérico" textmode="MultiLine" />
      </campo>
    </categoria>
  </form>
</componente>
```

```
<form nombre="Interfaces LAN" archivo="Incluir_Router_form4.aspx" id="form4" ide="4" />  
  <form nombre="Interfaces WAN" archivo="Incluir_Router_form5.aspx" id="form5" ide="ult" />  
</componente>
```

**Fig.44.Ejemplo de XML para el componente Router (XML\_Router.xml)**

Como se puede observar el documento XML consta de información referente al componente el cuál se encuentra compuesto por formularios que comprenden a su vez categorías, campos y controles.

Tomando como base el XML generado se procede a su transformación para la creación de las estructuras planteadas:

**Creación de los Formularios:** a través de un formulario Web denominado **Generador\_Forms.aspx** se toma el XML generado para el componente específico junto con los XSLTs (XSLT\_Inclusion, XSLT\_Inclusion\_Plantilla, XSLT\_Modificacion, XSL\_Consulta) para realizar la transformación. Para ello se utilizan las clases **XMLDocument** y **XslTransform**, específicamente el método **Load** para cargar los archivos:

```

archi_xml = "XML_Generado_" + compuesto + ".xml"
'CARGAR EL DOCUMENTO XML
Dim xmlDoc As XmlDocument = New XmlDocument()
xmlDoc.Load(Server.MapPath(archi_xml))

'CARGAR EL DOCUMENTO XSL
Dim xslDoc As XslTransform = new XslTransform()
xslDoc.Load(Server.MapPath("XSLT_Inclusion.xslt"))

```

**Fig.45. Carga del XML (Generador\_Forms.aspx)**

Posteriormente a través del método **Transform** de la clase **XslTransform** se realiza la transformación y se generan archivos .aspx llamados **Object\_Inclusion**, **Object\_InclPlant**, **Object\_Modificacion** y **Object\_Consulta**, a través del objeto **XmlTextWriter**:

```

file_writer = "Object_Inclusion.aspx" 'Nombre del archivo de salida
Dim TWrtr As New XmlTextWriter(Server.MapPath("Object_Inclusion.aspx"), System.Text.Encoding.Default)
xslDoc.Transform(xmlDoc, Nothing, TWrtr, Nothing)
TWrtr.Close()

```

**Fig.46. Transformación del XML a través del XSLT (Generador\_Forms.aspx)**

Estos archivos deben ser procesados para reemplazar caracteres especiales tales como “&” por “&”, “&lt;” por “<” y “&gt;” por “>”. Este procesamiento se realiza a través del formulario web **Replace\_Object.aspx** en el cuál se crea una instancia del objeto **StreamReader** para leer el archivo de entrada y otra instancia del objeto **StreamWriter** para generar finalmente los archivos que contendrán los formularios:

***Inclusion\_XX.aspx, Inclusion\_Plantilla\_XX.aspx, Modificacion\_XX.aspx y Consulta\_XX.aspx.***

```
Dim reader1 = New System.IO.StreamReader(Server.MapPath("Object_Inclusion.aspx"))
Dim reader2 = New System.IO.StreamReader(Server.MapPath("directivas.txt"))
Dim writer1 = New System.IO.StreamWriter(Server.MapPath("Proyecto\" & file_final))
```

***Fig.47. Instanciación de StreamReader para la lectura y escritura de archivos***

El proceso descrito anteriormente para la creación de formularios se muestra en la siguiente figura:

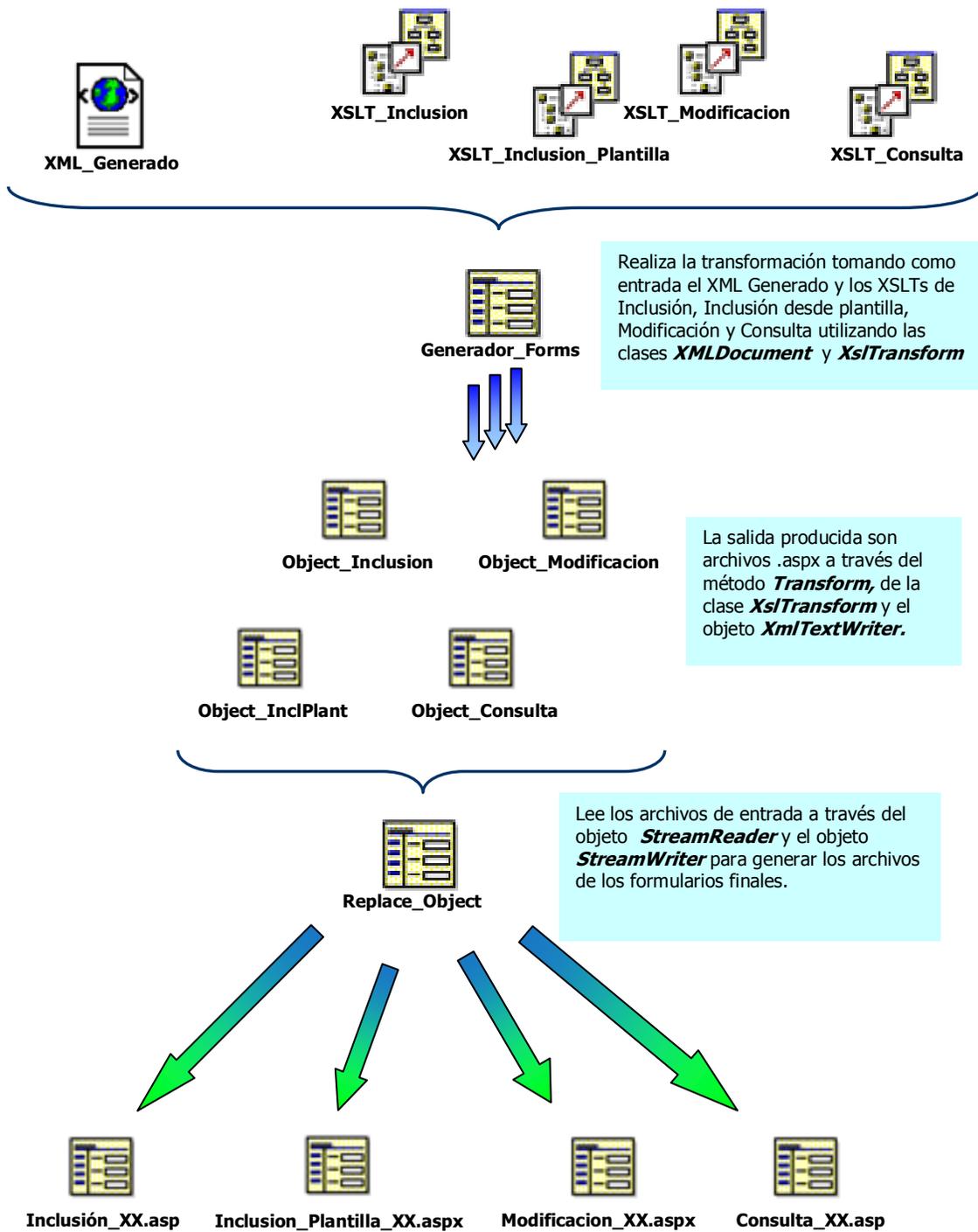


Fig.48. Proceso de creación de formularios

El archivo XSLT utilizado para realizar la transformación del XML esta compuesto por secciones `<![CDATA[ ]>` las cuáles permiten introducir código que no requiere de análisis. A continuación se muestra parte del código del archivo XSLT (*Método Page\_Load*) utilizado para generar el formulario de inclusión de componente:

```

<![CDATA[
Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

    Dim myConnString As String = Application("String_Con2")
    Dim myConnection As New System.Data.SqlClient.SqlConnection(myConnString)
    myConnection.Open()
]]>
<xsl:for-each select="form/categoria/campo/control">
  <xsl:if test="@type='asp:DropDownList'">
    <![CDATA[ Dim ]]> <xsl:value-of select="@query"/> <![CDATA[ As String = "SELECT * FROM ]]>
    <xsl:value-of select="@table"/><![CDATA[ ORDER BY ]]><xsl:value-of select="@att"/><![CDATA[ ASC]]>
    <![CDATA[ Dim ]]> <xsl:value-of select="@comm"/> <![CDATA[ As New System.Data.SqlClient.SqlCommand]]>
    <xsl:value-of select="@query"/><![CDATA[,myConnection]]>
    <![CDATA[ Dim ]]> <xsl:value-of select="@read"/> <![CDATA[ As System.Data.SqlClient.SqlDataReader ]]>

    <![CDATA[ If Not IsPostBack Then]]>
      <xsl:value-of select="@read"/> <![CDATA[=]]> <xsl:value-of select="@comm"/><![CDATA[.ExecuteReader() ]]>
      <xsl:value-of select="@id"/><![CDATA[.DataSource = ]]><xsl:value-of select="@read"/> <![CDATA[ ]]>
      <xsl:value-of select="@id"/><![CDATA[.DataTextField = "]]><xsl:value-of select="@att"/><![CDATA[" ]]>
      <xsl:value-of select="@id"/><![CDATA[.DataBind()]]>
      <xsl:value-of select="@read"/><![CDATA[.Close()]]>
    <![CDATA[
    End If
    myConnection.Close()
  ]]>
</xsl:if>
</xsl:for-each>
<![CDATA[End Sub

```

**Fig.49. Método Page Load del XSLT de inclusión (XSLT\_Inclusión.xslt)**

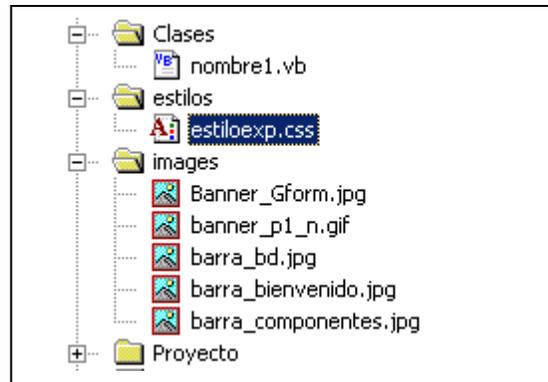
## Estructura del Directorio de la Aplicación GForm

La aplicación Gform corresponde a un proyecto en Visual Studio.NET de nombre **“Generator Dynamic”** cuya estructura se presenta en la siguiente figura:



**Fig.50. Estructura del directorio de la aplicación GForm**

Dentro de la carpeta **clases** se encuentra la clase llamada “**nombre1.vb**”, la cuál es utilizada para crear los nombres de los archivos generados. La carpeta **estilos** contiene el archivo “**estiloexp.css**” con las hojas de estilo utilizadas en la aplicación. Del mismo modo, la carpeta **images** contiene todas las imágenes y por último dentro de la carpeta **Proyecto** es donde se almacenan todos los formularios generados de forma dinámica por la aplicación (el detalle del contenido de esta carpeta se mostrará en la sección 2.5, específicamente en la Fase de Implementación de la aplicación GenSigs)



**Fig.51. Carpetas de la aplicación GForm**

**Creación de las Tablas:** una vez que son obtenidos los datos de IP del servidor, puerto, nombre, login y password de la base de datos en donde el usuario desea crear las tablas, a través de un Web Form se van construyendo las sentencias SQLs de creación de las mismas tomando los datos del XML correspondiente al componente específico. Para recorrer el XML se utiliza una instancia específica del objeto **XmlTextReader** que tal como su nombre lo indica permite la lectura de un archivo XML:

```

archi_xml = "XML_Generado_" + compuesto + ".xml"
Dim rdr As New XmlTextReader(Server.MapPath(archi_xml))
ReadandWrite(rdr)

```

**Fig.52. Instancia para recorrer el XML (Creacion\_tables.aspx)**

Adicionalmente son utilizados los métodos **Read, Name y NodeType** de este objeto para hacer el recorrido.

**Creación de Scripts:** al igual que para la creación de las tablas, para generar el archivo de texto con los scripts SQL se va recorriendo el XML correspondiente al componente

específico a través de una instancia del objeto *XmlTextReader*. Adicionalmente se instancia un objeto de la clase *StreamWriter* para escribir las sentencias SQL en un archivo de texto cuyo nombre es indicado al usuario.

```
archi_script = "SQLS_" + compuesto + ".txt"

writer1 = New System.IO.StreamWriter(Server.MapPath(archi_script))

archi_xml = "XML_Generado_" + compuesto + ".xml"

Dim rdr As New XmlTextReader(Server.MapPath(archi_xml))

ReadandWrite(rdr)

writer1.Close()

lblStatus.Text = "Fue generado el archivo " + archi_script + " con los Scripts solicitados"
```

**Fig.53. Generación del archivo de texto con los scripts SQL (Creacion\_txt.aspx)**

En la sección siguiente se presenta un caso de estudio para la elaboración de los formularios de un componente llamado ATM. Se incluyen las pantallas utilizadas en dicho proceso así como el detalle del mismo para dar una idea más clara del funcionamiento del sistema.

#### **2.4.1 Caso de Estudio para la Aplicación Generadora de Formularios: GForm**

Para diseñar un formulario que pueda ser utilizado en la gestión de un componente ATM es necesario incluir la siguiente información (se muestra la clasificación en pestañas, categorías, campos y controles para visualizarla más claramente):

PESTAÑAS	CATEGORIAS	CAMPOS	CONTROLES
INFORMACIÓN GENERAL (*)	Información General del Componente (*)	Nombre del Componente (*)	Text Box
		Tipo de Cliente	Drop Down List
		Cliente al que sirve	Drop Down List
		Plataforma	Text Box
		Código del ATM	Text Box
	Ubicación Geográfica	Ciudad	Text Box
		Estado	Text Box
DATOS DEL ENLACE	Información del Enlace	Carrier	Drop Down List
		Posee respaldo	Radio Button
	Ancho de Banda	Ancho de Banda	Text Box
		CIR	Text Box
		EIR	Text Box
DATOS DEL ROUTER	Información del Router	Marca	Text Box
		Modelo	Text Box
		Posee Port Sharing	Radio Button

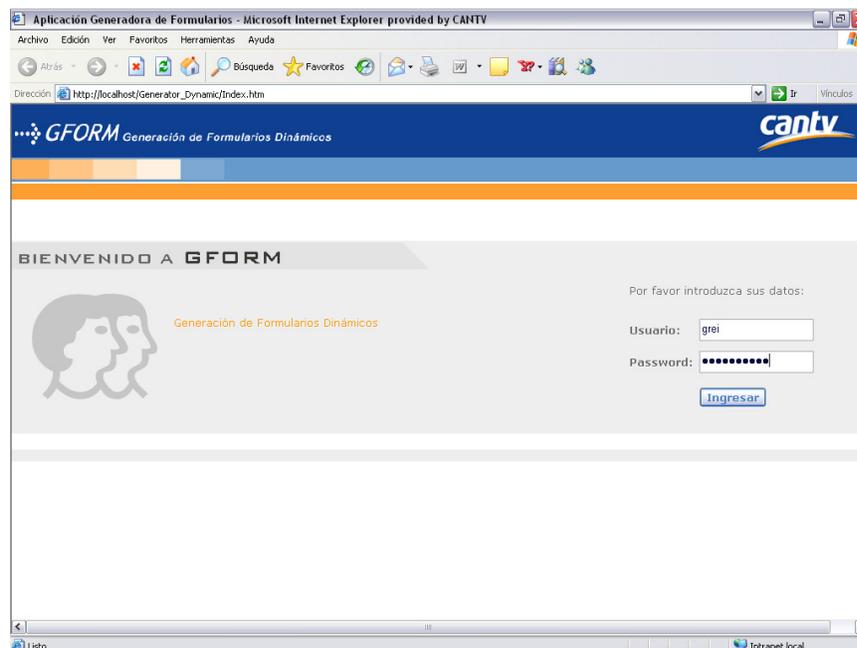
**Tabla N°2. Información para el diseño del componente ATM**

(\*) Elementos obligatorios que aparecerán por defecto en todos los formularios

Es importante señalar que para efectos de la consulta de un componente específico en la aplicación GenSigs, es necesario incluir en los formularios el campo **Nombre del Componente**, para poder realizar la búsqueda del mismo en base a dicho campo y al tipo de componente señalado por el usuario. Este se ubica en la categoría **Información General del Componente** dentro de la pestaña **Información General**. Todos los formularios que genere el usuario contendrán por defecto estos elementos por lo que no podrán ser modificados ni eliminados. Sin embargo, si el usuario así lo desea podrá incluir otros campos adicionales en esta categoría por defecto.

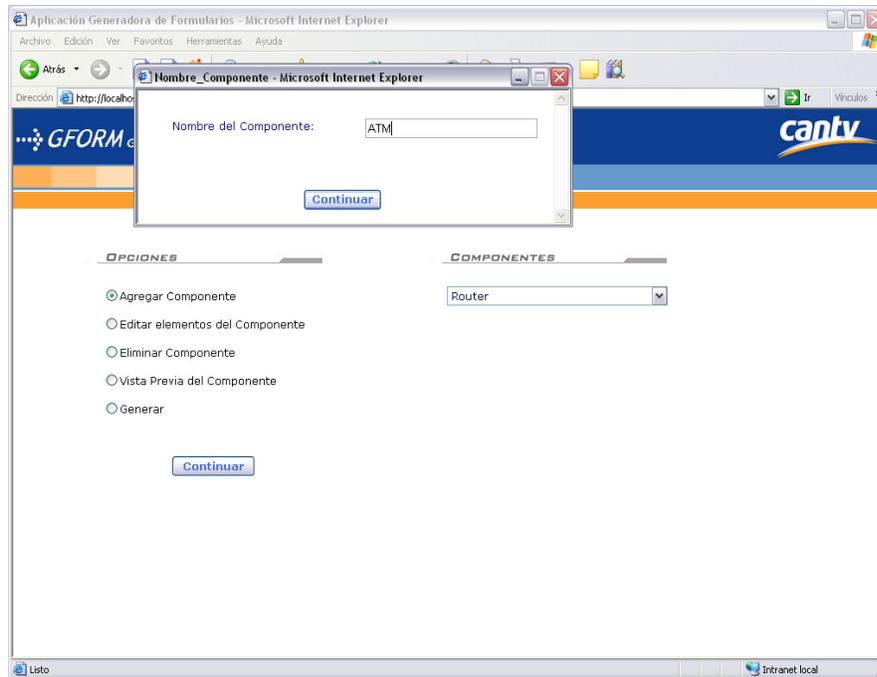
Los pasos para la creación de los formularios, scripts y tablas del componente ATM con los datos anteriores son los siguientes:

1. El usuario introduce su login y password en la página principal del sistema.



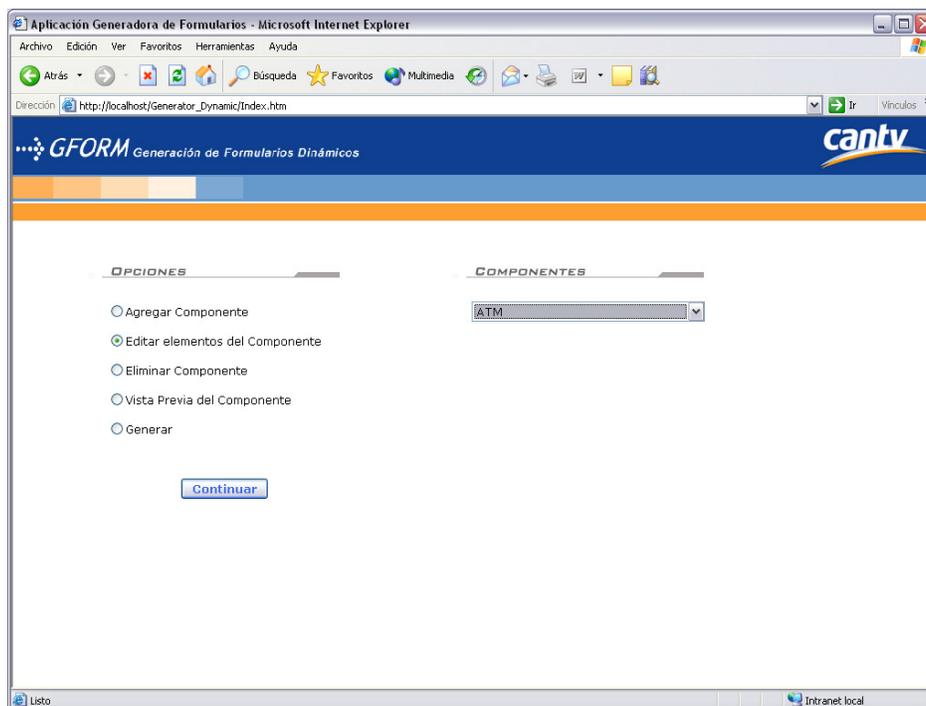
**Fig.54. Página de acceso a la aplicación**

2. Se elige la opción *Agregar Componente* y se introduce el nombre "ATM", tal como se muestra:



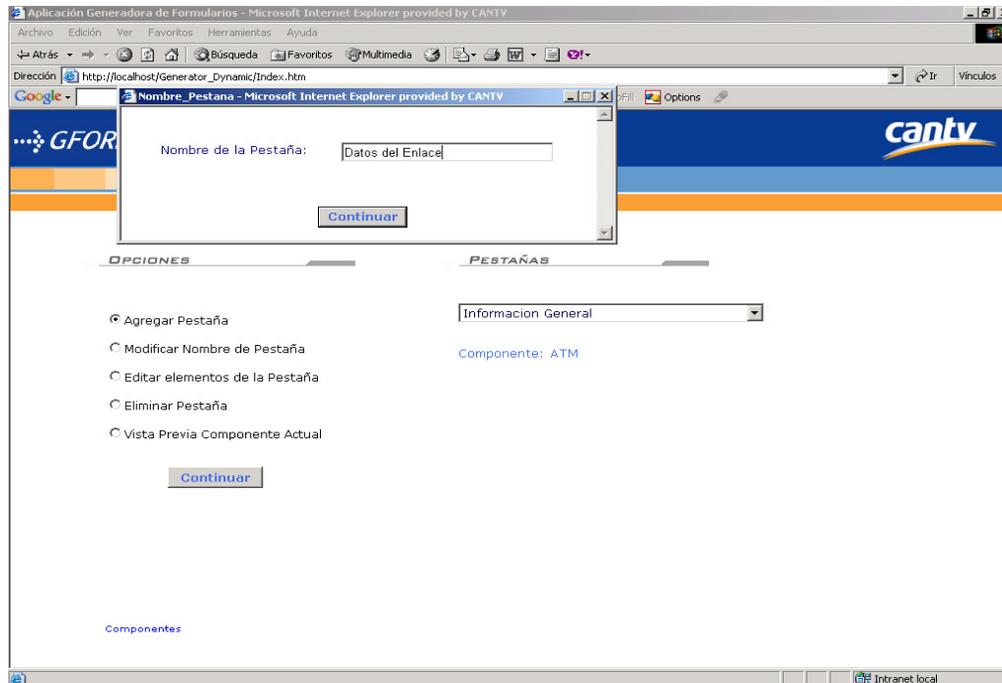
**Fig.55. Página para especificar el nombre del componente**

3. Una vez seleccionado ATM en la lista de componentes se elige la opción *Editar Elementos del Componente* para agregar las Pestañas del formulario.



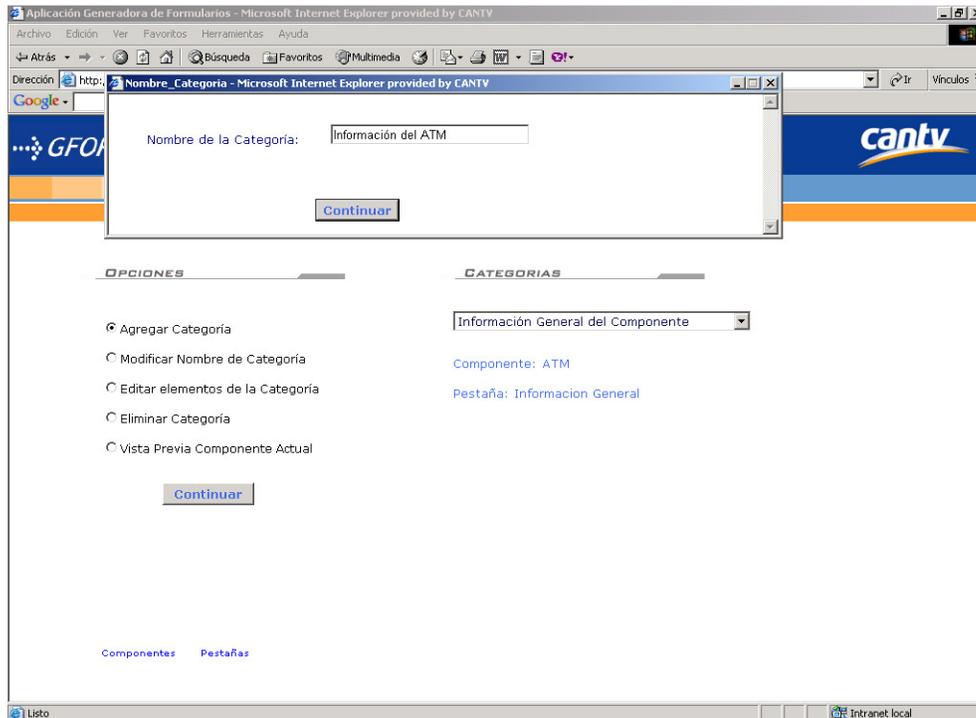
**Fig.56. Página para editar los elementos del componente ATM**

4. En la página siguiente se agregan las demás pestañas, para lo cual se introducen uno a uno el nombre de las mismas: Datos del Enlace y Datos del Router, tal como se muestra a continuación:



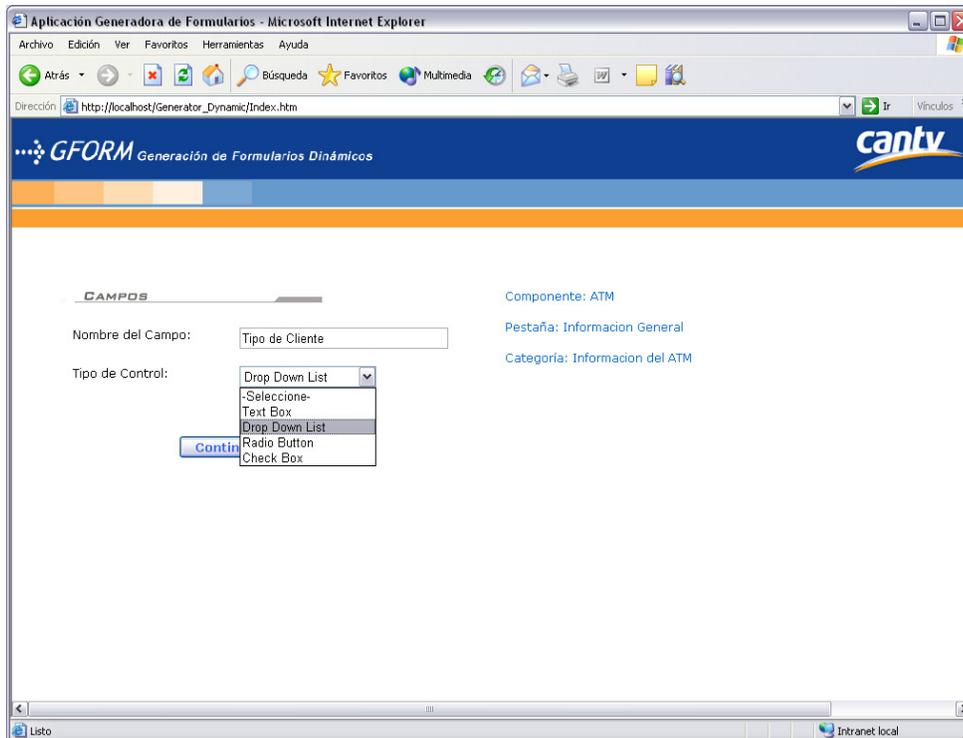
**Fig.57. Página para agregar pestañas**

5. Para agregar las categorías a cada pestaña, se selecciona cada una de la lista desplegada y se elige la opción *Editar Elementos de la Pestaña*. Una vez en la página de categorías se escoge la opción *Agregar Categoría*. Para la primera pestaña aparecerá por defecto la categoría Información General del Componente, luego se agrega la categoría Ubicación Geográfica. Para la pestaña Datos del Enlace se añade Información del Enlace y Ancho de Banda y por ultimo en la pestaña Datos del Router se agrega Información del Router:



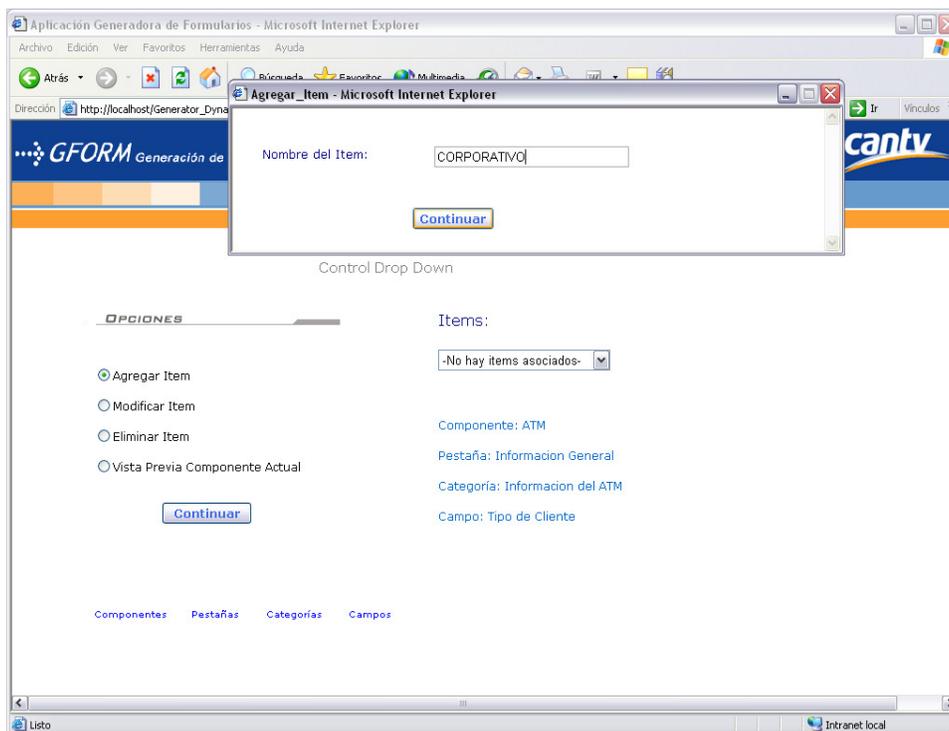
**Fig.58. Página para incluir las categorías**

6. Una vez agregadas las categorías se procede a incluir los campos que formarán parte de las mismas. Para ello se elige la opción *Editar Elementos de la Categoría* en cada una de ellas y luego *Agregar Campo*, indicando el nombre del mismo y el tipo de control asociado:



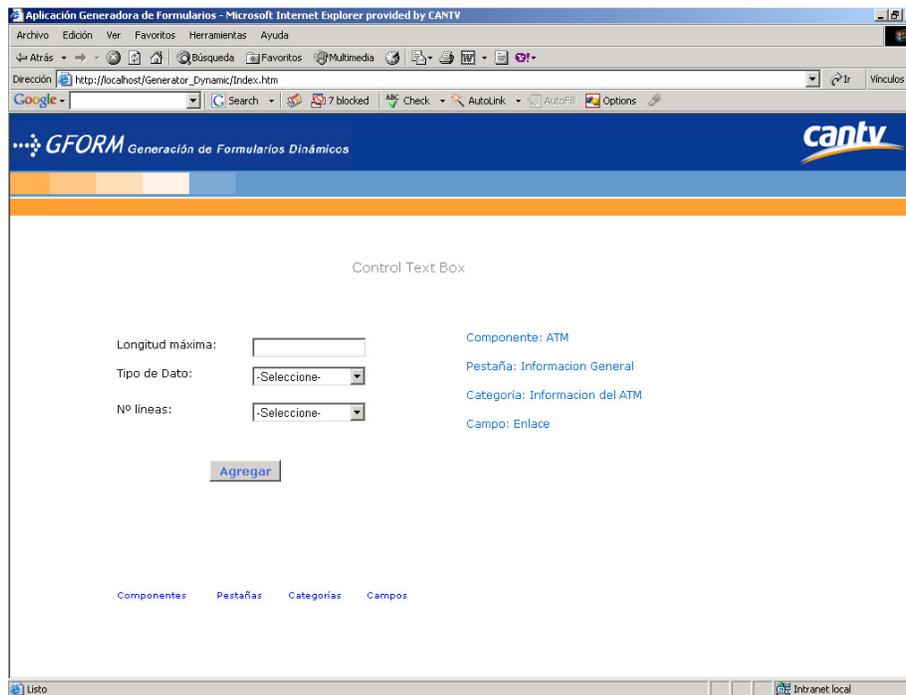
**Fig.59. Página para agregar campos**

7. Para agregar los controles de tipo Drop Down List (combos) se especifican cada uno de los items que formaran parte de dicho control a través de la opción *Agregar Item*



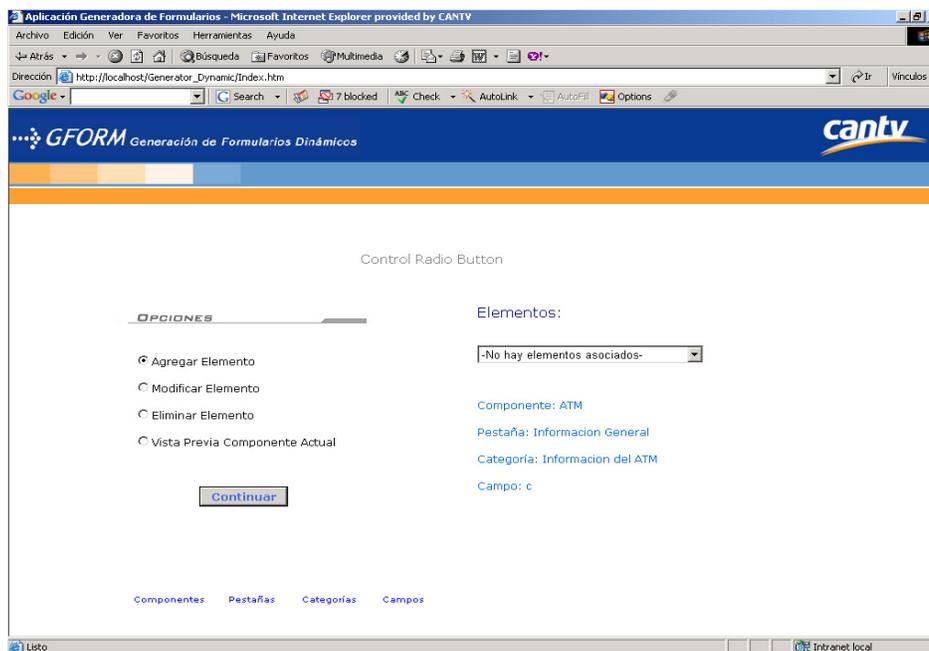
**Fig.60. Página para agregar un control Drop Down**

8. Se agregan los controles de tipo Text Box señalando los valores de longitud máxima, tipo de dato y número de líneas:



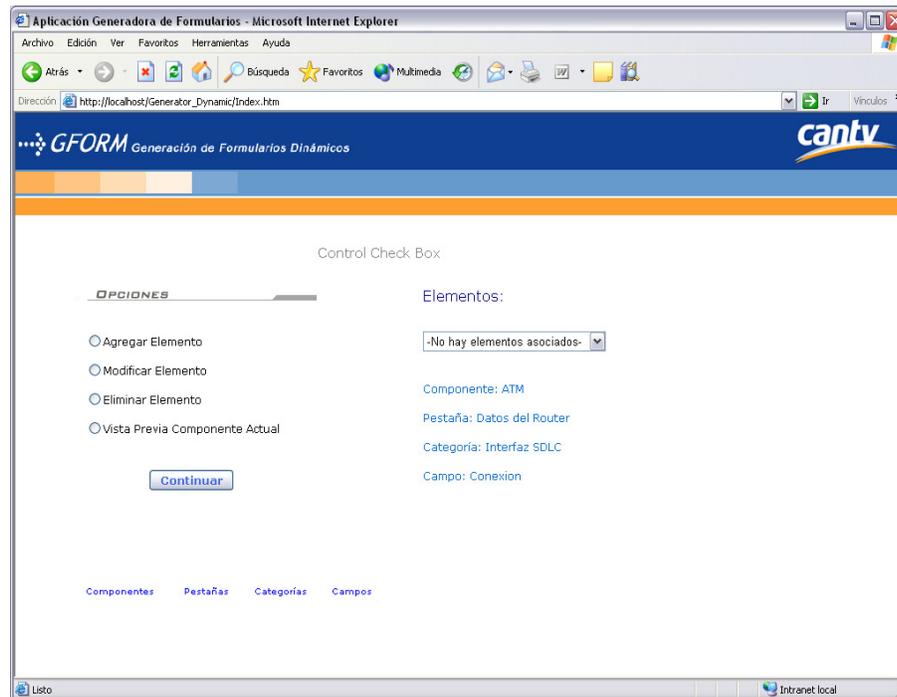
**Fig.61. Página para agregar un control Text Box**

9. Los controles Radio Button se agregan desde una página como la siguiente:



**Fig.62. Página para agregar un control Radio Button**

10. Los controles Check Box se agregan desde la siguiente página:

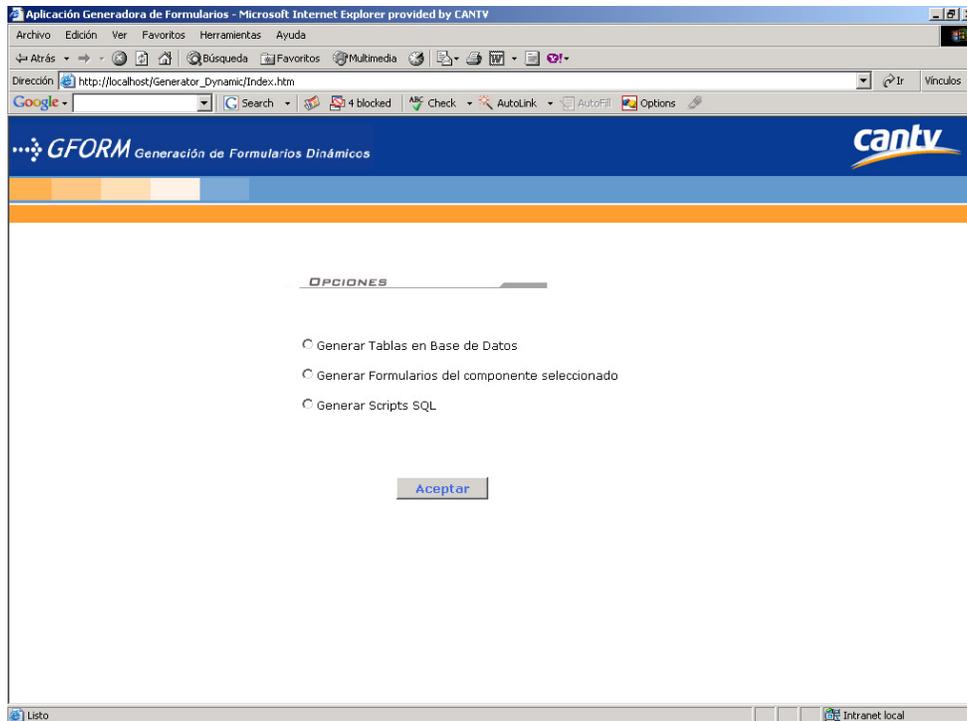


**Fig.63. Página para agregar un control check box**

Para los últimos dos tipos de controles se lleva a cabo una secuencia parecida a la del control Drop Down List para agregar los ítems.

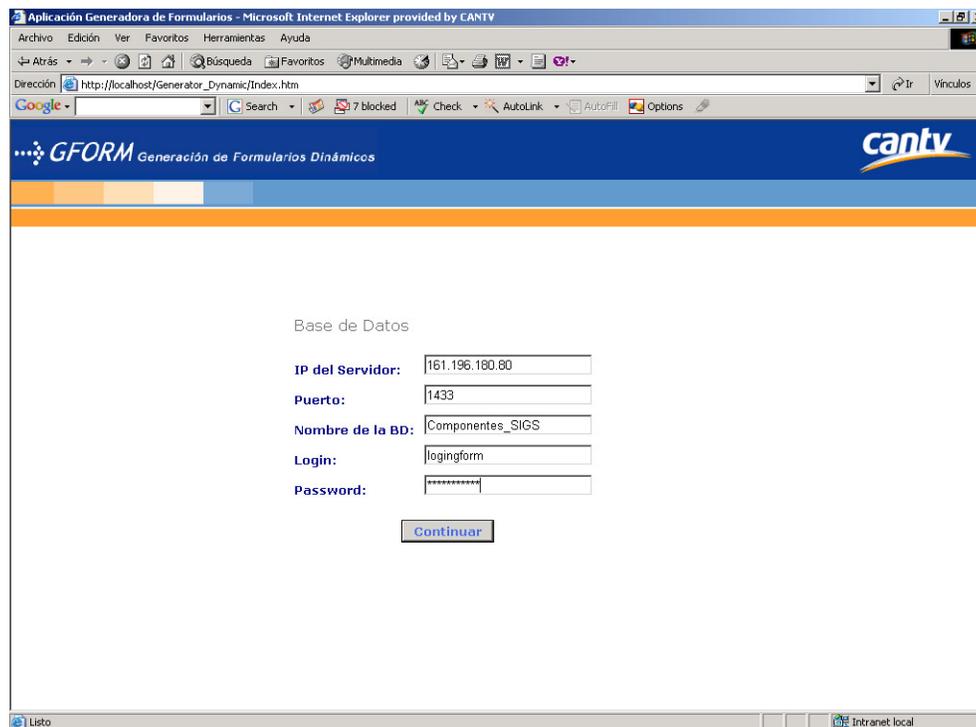
11. Al terminar de ingresar todos los datos de diseño del nuevo tipo de componente se procede a generar las tablas en la base de datos indicada por el usuario, los formularios y los scripts SQL de las tablas que soportan el almacenamiento de los mismos.

Para esto desde la página de Componentes se elige ATM y luego la opción *Generar* mostrándose la siguiente página:



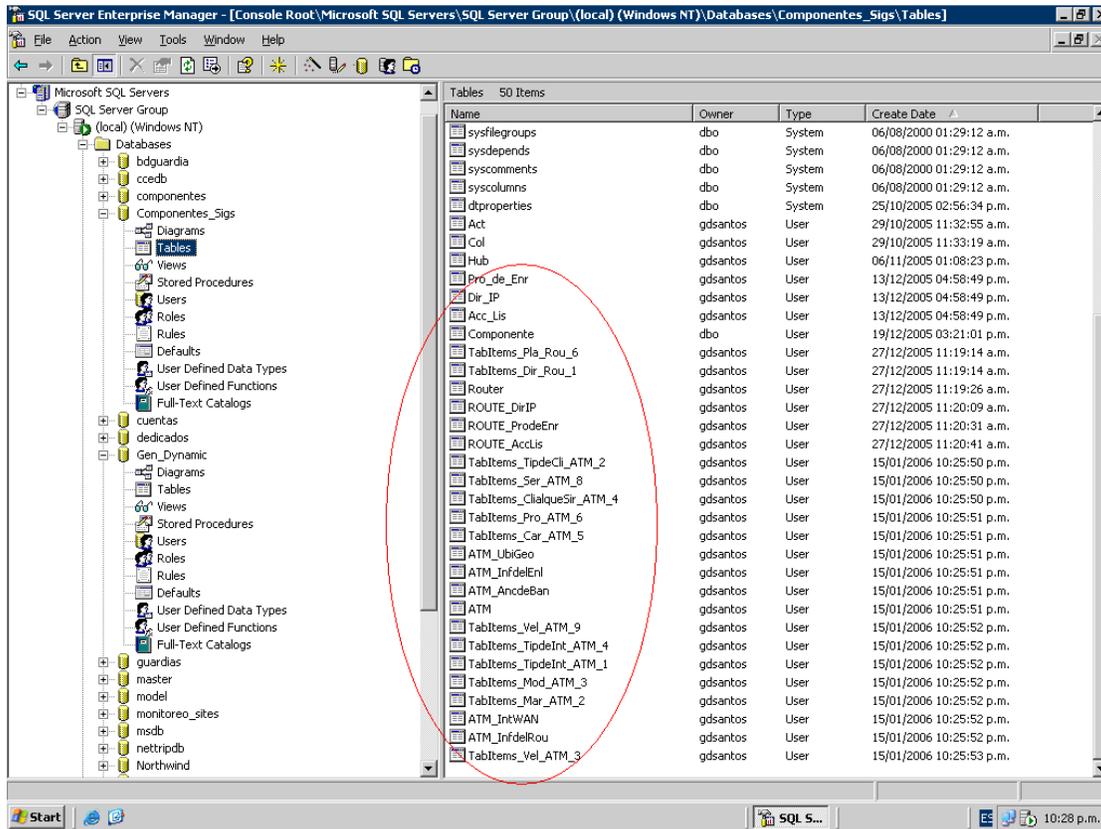
**Fig.64. Listado de opciones a generar para un componente**

12. Se selecciona la opción *Generar Tablas en Base de Datos* que luego lleva a una página en la que se deben señalar los datos de la misma:



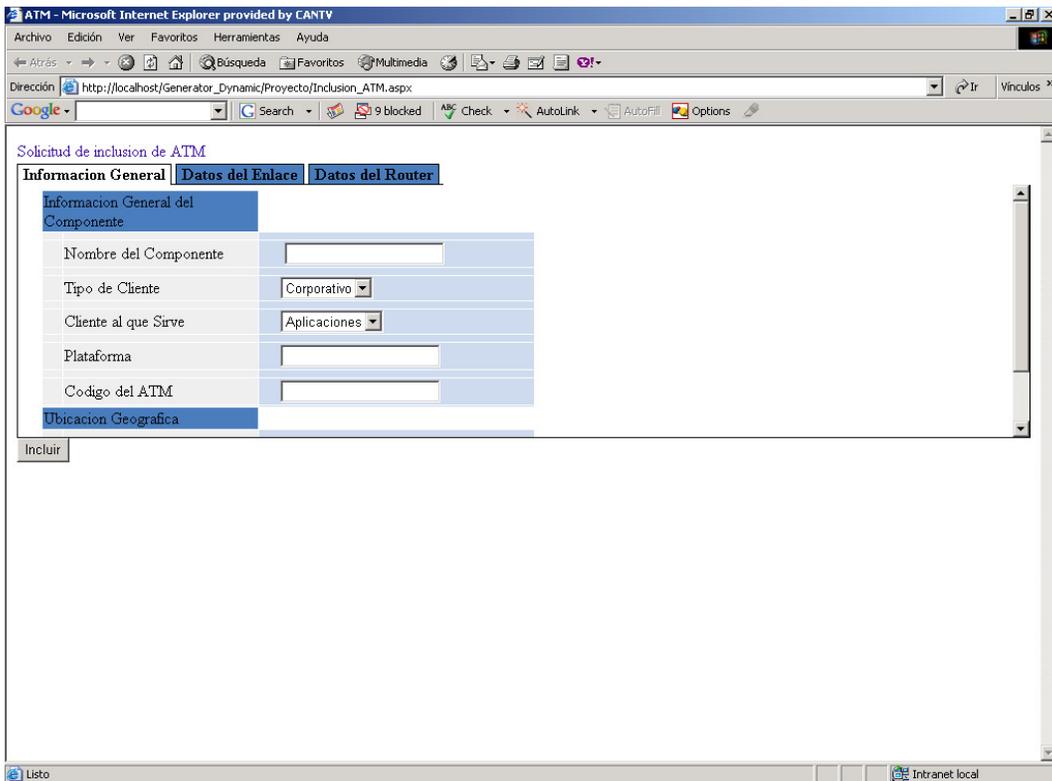
**Fig.65. Datos específicos para la base de datos**

De esta manera se crean las tablas en la base de datos que señale el usuario, en este caso son creadas en la base de datos **Componentes\_SIGS** ubicada en el servidor con IP 161.196.180.80. A continuación se muestra una vista de las tablas generadas:



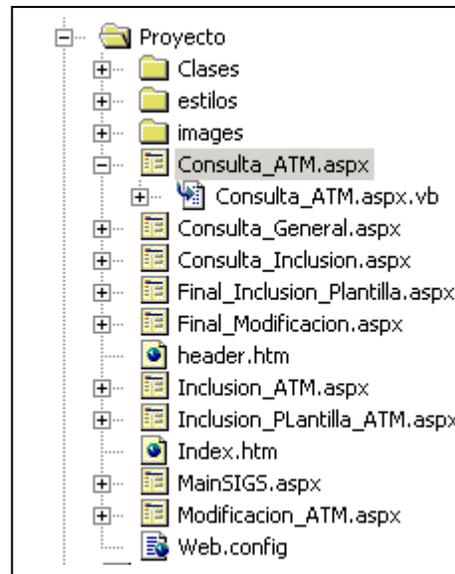
**Fig.66. Tablas generadas para el componente ATM**

13. Para crear los formularios se escoge la opción *Generar Formularios del componente seleccionado* creándose de esta manera los archivos para Inclusión, Inclusión desde Plantilla, Modificación y Consulta. A continuación se muestra la interfaz del formulario de inclusión generado para el componente ATM:



**Fig.67. Formulario de Inclusión generado para ATM**

Todos los formularios generados se almacenan dentro de la carpeta **Proyecto** de la aplicación Gform, la cuál presenta la siguiente estructura:



**Fig.68. Estructura de la carpeta Proyecto de la aplicación GForm**

Se observa la carpeta **Clases** que contiene la clase “**nombre.vb**” utilizada para crear dinámicamente el nombre de los archivos, la carpeta **estilos** con el archivo de hojas de estilo “**estiloexp.css**” y la carpeta **images** con las imágenes de la aplicación. En la raíz de la carpeta se encuentran ciertos archivos html y aspx estáticos, que forman parte del inicio de la aplicación: **header.htm**, **Index.htm**, **MainSIGs.aspx**, **Consulta\_General.aspx**, **Consulta\_Inclusión.aspx**, **Final\_Inclusión**, **Final\_Inclusion\_Plantilla.aspx**, **Final\_Modificación.aspx**, **Final\_Consulta.aspx**

14. Por último se selecciona la opción *Generar Scripts SQL* a través de la cuál se construye un archivo de texto que contiene los scripts de creación de tablas así como los scripts de inserción de la información de contenido de los controles Drop Down List: El contenido del archivo generado para el componente ATM es el siguiente:

```

CREATE TABLE TabItems_TipdeCli_ATM_1(id smallint PRIMARY KEY CLUSTERED,
texto varchar(50) NOT NULL)
INSERT INTO TabItems_TipdeCli_ATM_1 Values(1,'Corporativo')
INSERT INTO TabItems_TipdeCli_ATM_1 Values(2,'Externo')
INSERT INTO TabItems_TipdeCli_ATM_1 Values(3,'ISP')
CREATE TABLE TabItems_ClialqueSir_ATM_1(id smallint PRIMARY KEY CLUSTERED,
texto varchar(50) NOT NULL)
INSERT INTO TabItems_ClialqueSir_ATM_1 Values(1,'Cantv')
INSERT INTO TabItems_ClialqueSir_ATM_1 Values(2,'Cantv.net')
INSERT INTO TabItems_ClialqueSir_ATM_1 Values(3,'Movilnet')
INSERT INTO TabItems_ClialqueSir_ATM_1 Values(4,'Aplicaciones')
CREATE TABLE ATM(id_Comp int primary key, NomdelCom varchar(100),TipdeCli
varchar(100),ClialqueSir varchar(100),Pla varchar(100),CodelATM
varchar(100))
CREATE TABLE ATM_UbiGeo(id_Categ int primary key, Ciu varchar(100),Est
varchar(100), id_comp int)
CREATE TABLE TabItems_Car_ATM_1(id smallint PRIMARY KEY CLUSTERED, texto
varchar(50) NOT NULL)
INSERT INTO TabItems_Car_ATM_1 Values(1,'Cantv')
INSERT INTO TabItems_Car_ATM_1 Values(2,'Movilnet')
INSERT INTO TabItems_Car_ATM_1 Values(3,'Bantel')
INSERT INTO TabItems_Car_ATM_1 Values(4,'COMSAT')
CREATE TABLE ATM_InfdelEnl(id_Categ int primary key, Car varchar(100),Posres
varchar(100), id_comp int)

CREATE TABLE ATM_AncdeBan(id_Categ int primary key, AncdeBan varchar(100),CIR
varchar(100),EIR varchar(100), id_comp int)
CREATE TABLE ATM_InfdelRou(id_Categ int primary key, Mar varchar(100),Mod
varchar(100),PosPorSha varchar(100), id_comp int)

```

**Fig.69. Archivo con scripts para ATM (SQLs\_ATM.txt)**

Estos Scripts podrán ser ejecutados en la base de datos SQL Server requerida por el usuario.

Para poder generar cualquiera de las tres opciones la aplicación construye un documento XML con toda la información del nuevo componente. A continuación se muestra dicho documento para el componente ATM:

```

<?xml version="1.0" encoding="utf-8"?>
<componente nombre="ATM">
  <form nombre="Informacion General" archivo="Incluir_ATM_form1.aspx" id="form1" ide="1">
    <categoria nombre="Informacion General del Componente" abv="ATM_InfGendelCom" ide="1">
      <campo nombre="Nombre del Componente" abv="NomdelCom" tipo="varchar(100)" ide="1" categ="1">
        <control type="asp:TextBox" id="txt_Pred_NombreC" maxlength="50" tipo_dato="Caracter" textmode="SingleLine"
        />
      </campo>
      <campo nombre="Tipo de Cliente" abv="TipdeCli" tipo="varchar(100)" ide="2" categ="1">
        <control type="asp:DropDownList" id="dropdown_TipdeCliATM1" table="TabItems_TipdeCli_ATM_1"
        query="myQuery_TipdeCliATM1" comm="myCommand_TipdeCliATM1" read="myReader_TipdeCliATM1" />
      </campo>
      <campo nombre="Cliente al que Sirve" abv="ClialqueSir" tipo="varchar(100)" ide="3" categ="1">
        <control type="asp:DropDownList" id="dropdown_ClialqueSirATM1" table="TabItems_ClialqueSir_ATM_1"
        query="myQuery_ClialqueSirATM1" comm="myCommand_ClialqueSirATM1" read="myReader_ClialqueSirATM1" />
      </campo>
      <campo nombre="Plataforma" abv="Pla" tipo="varchar(100)" ide="4" categ="1">
        <control type="asp:TextBox" id="txt_PlaATM1" maxlength="30" tipo_dato="Caracter" textmode="SingleLine" />
      </campo>
      <campo nombre="Codigo del ATM" abv="CodelATM" tipo="varchar(100)" ide="ult" categ="1">
        <control type="asp:TextBox" id="txt_CodelATMATM10" maxlength="50" tipo_dato="Caracter"
        textmode="SingleLine" />
      </campo>
    </categoria>
    <categoria nombre="Ubicacion Geografica" abv="ATM_UbiGeo" ide="2">
      <campo nombre="Ciudad" abv="Ciu" tipo="varchar(100)" ide="1" categ="2">
        <control type="asp:TextBox" id="txt_CiuATM4" maxlength="30" tipo_dato="Caracter" textmode="SingleLine" />
      </campo>
      <campo nombre="Estado" abv="Est" tipo="varchar(100)" ide="ult" categ="2">
        <control type="asp:TextBox" id="txt_EstATM10" maxlength="60" tipo_dato="Caracter" textmode="SingleLine" />
      </campo>
    </categoria>
  </form>
  <form nombre="Datos del Enlace" archivo="Incluir_ATM_form2.aspx" id="form2" ide="2">
    <categoria nombre="Informacion del Enlace" abv="ATM_InfdelEnl" ide="3">
      <campo nombre="Carrier" abv="Car" tipo="varchar(100)" ide="1" categ="3">
        <control type="asp:DropDownList" id="dropdown_CarATM1" table="TabItems_Car_ATM_1" att="texto"
        query="myQuery_CarATM1" comm="myCommand_CarATM1" read="myReader_CarATM1" />
      </campo>
      <campo nombre="Posee respaldo" abv="Posres" tipo="varchar(100)" ide="ult" categ="3">
        <control type="asp:RadioButton" id="radio_PosresATM101151" GroupName="RadioGroup115" Text="Si" ide="1" />
        <control type="asp:RadioButton" id="radio_PosresATM41152" GroupName="RadioGroup115" Text="No" ide="ult" />
      </campo>
    </categoria>
  </form>

```

```

<form nombre="Datos del Enlace" archivo="Incluir_ATM_form2.aspx" id="form2" ide="2">
  <categoria nombre="Informacion del Enlace" abv="ATM_InfdelEnl" ide="3">
    <campo nombre="Carrier" abv="Car" tipo="varchar(100)" ide="1" categ="3">
      <control type="asp:DropDownList" id="dropdown_CarATM1" table="TabItems_Car_ATM_1" att="texto"
query="myQuery_CarATM1" comm="myCommand_CarATM1" read="myReader_CarATM1" />
    </campo>
    <campo nombre="Posee respaldo" abv="Posres" tipo="varchar(100)" ide="ult" categ="3">
      <control type="asp:RadioButton" id="radio_PosresATM101151" GroupName="RadioGroup115" Text="Si" ide="1" />
      <control type="asp:RadioButton" id="radio_PosresATM41152" GroupName="RadioGroup115" Text="No" ide="ult" />
    </campo>
  </categoria>
  <categoria nombre="Ancho de Banda" abv="ATM_AncdeBan" ide="4">
    <campo nombre="Ancho de Banda" abv="AncdeBan" tipo="varchar(100)" ide="1" categ="4">
      <control type="asp:TextBox" id="txt_AncdeBanATM4" maxlength="30" tipo_dato="Numérico" textmode="SingleLine"
/>
    </campo>
    <campo nombre="CIR" abv="CIR" tipo="varchar(100)" ide="2" categ="4">
      <control type="asp:TextBox" id="txt_CIRATM9" maxlength="30" tipo_dato="Caracter" textmode="SingleLine" />
    </campo>
    <campo nombre="EIR" abv="EIR" tipo="varchar(100)" ide="ult" categ="4">
      <control type="asp:TextBox" id="txt_EIRATM6" maxlength="20" tipo_dato="Numérico" textmode="SingleLine" />
    </campo>
  </categoria>
</form>
</componente>

```

**Fig.70. XML generado para el componente ATM (XML\_Generado\_ATM.xml)**

---

Mas adelante, en la sección 2.5.2 se desarrollara un caso de estudio con los formularios generados anteriormente para el componente ATM. Se incluirán pantallas del sistema para el manejo de componentes así como las pruebas que permitan verificar las funcionalidades del mismo.

## **FASE 6: Prueba y Evaluación (GForm)**

Las pruebas realizadas corresponden a la evaluación del comportamiento de la aplicación con respecto a los requisitos funcionales (Pruebas Funcionales). Se utiliza el mismo formato de pruebas de la Coordinación de Mantenimiento de Aplicaciones de Cantv.

### **Prueba 1: Crear Diseño de los Formularios**

Validar que el usuario pueda crear de manera satisfactoria el diseño deseado de los formularios para un componente.

#### **Pasos de la Prueba:**

1. El usuario se autentica en la aplicación.
2. Una vez en la página de Componentes selecciona la opción Agregar Componente especificando el nombre del mismo.
3. Selecciona el nuevo componente agregado y luego escoge la opción Editar elementos del Componente.
4. Desde la página de pestañas agrega las pestañas que desea que tenga el formulario para ese componente, indicando el nombre de las mismas.
5. Para cada pestaña agregada selecciona la opción Editar elementos de la Pestaña.
6. Desde la página de Categorías agrega todas las categorías que desee para esa pestaña y para cada una selecciona la opción Editar elementos de la Categoría.
7. Ubicado en la página de Campos agrega los campos asociados a cada categoría específica y para cada uno de los campos seleccionados escoge la opción Editar elementos del Campo.
8. Selecciona el control que desea para el campo específico, escogiendo entre Text, DropDownList, Check o Radio Button y especifica para cada uno de ellos las características que desea cumpla el control.

**Resultados Esperados:**

Se debe crear un formulario que contenga las pestañas, categorías, campos y controles especificados por el usuario.

**Resultados Obtenidos:**

Se crea un formulario con las pestañas, categorías, campos y controles especificados por el usuario.

**Prueba 2: Modificar Diseño de los Formularios**

Validar que el usuario pueda modificar satisfactoriamente el diseño de los formularios para un componente específico.

**Pasos de la Prueba:**

1. El usuario se autentica en la aplicación.
2. Desde la página de Componentes selecciona el Componente al cuál desea modificar el diseño y escoge la opción Editar elementos del Componente.
3. Una vez en la página de Pestañas selecciona la pestaña que desea modificar y luego la opción Modificar Nombre de Pestaña indicando el nuevo nombre que desee para la misma.
4. Selecciona la opción Editar elementos de la Pestaña.
5. Selecciona la categoría a la cuál desea modificar el nombre y luego la opción Modificar Nombre de Categoría especificando el nuevo nombre que desea tenga la misma.
6. Selecciona la opción Editar elementos de la Categoría.
7. Selecciona el campo al cuál desea modificar el nombre y luego la opción Modificar Nombre del Campo especificando el nuevo nombre que desea tenga el mismo.

**Resultados Esperados:**

El formulario debe ser modificado según el nuevo diseño que indica el usuario.

**Resultados Obtenidos:**

Se modifica el formulario según el nuevo diseño que indica el usuario.

---

### **Prueba 3: Eliminar entidades del diseño de los Formularios**

Validar que el usuario pueda eliminar de manera satisfactoria entidades en el diseño de los formularios para un componente o eliminar un componente como tal.

#### **Pasos de la Prueba:**

1. El usuario se autentica en la aplicación.
2. Desde la página de Componentes selecciona el Componente sobre el cuál quiere eliminar entidades y escoge la opción Editar Elementos del Componente.
3. Si es una pestaña la que desea eliminar la selecciona del combo de pestañas y escoge la opción Eliminar Pestaña.
4. Si desea eliminar una categoría perteneciente a esa pestaña selecciona la opción Editar elementos de la Pestaña y una vez en la página de categorías selecciona del combo la que desea eliminar y escoge la opción Eliminar Categoría.
5. Si desea eliminar un campo perteneciente a esa categoría selecciona la opción Editar elementos de la Categoría y una vez en la página de campos selecciona del combo el que desea eliminar y escoge la opción Eliminar Campo.

#### **Resultados Esperados:**

Se deben eliminar de los formularios del Componente escogido las entidades señaladas por el usuario o bien si es seleccionado así el componente completo.

#### **Resultados Obtenidos:**

Se eliminan de los formularios del Componente escogido las entidades señaladas por el usuario.

### **Prueba 4: Generar Formularios del componente seleccionado**

Validar que se puedan generar satisfactoriamente los formularios de Inclusión, Inclusión desde Plantilla, Modificación y Consulta con el diseño especificado por el usuario.

#### **Pasos de la Prueba:**

1. El usuario se autentica en la aplicación.
2. Una vez en la página de Componentes selecciona el componente de la lista de componentes creados y luego la opción "Generar".

3. Ubicado en la página de opciones selecciona la opción Generar Formularios del componente seleccionado y luego presiona el botón “Aceptar”.

**Resultados Esperados:**

Se deben crear los formularios de Inclusión, Inclusión desde Plantilla, Modificación y Consulta para el componente señalado.

**Resultados Obtenidos:**

Se crean los formularios de Inclusión, Inclusión desde Plantilla, Modificación y Consulta para el componente señalado.

**Prueba 5: Generar Scripts del componente seleccionado**

Validar que se pueda generar satisfactoriamente un archivo de texto que contenga los scripts SQL para creación de las tablas que soporten el almacenamiento de los datos para los formularios.

**Pasos de la Prueba:**

1. El usuario se autentica en la aplicación.
2. Una vez en la página de Componentes selecciona el componente de la lista de componentes creados y luego la opción “Generar”.
3. Ubicado en la página de opciones selecciona la opción Generar Scripts SQL del componente seleccionado y luego presiona el botón “Aceptar”.

**Resultados Esperados:**

Se debe crear un archivo de texto con el nombre “**SQLS\_<nombre\_componente>.txt**” que contenga todos los scripts para la creación de tablas para almacenamiento de datos.

**Resultados Obtenidos:**

Es creado un archivo de texto con el nombre “**SQLS\_<nombre\_componente>.txt**” que contiene todos los scripts para la creación de tablas para almacenamiento de datos.

**Prueba 6: Generar Tablas en la Base de Datos**

Validar que se puedan crear de manera satisfactoria las tablas que soporten el almacenamiento de los datos para los formularios de un componente específico.

**Pasos de la Prueba:**

1. El usuario se autentica en la aplicación.
2. Una vez en la página de Componentes selecciona el componente de la lista de componentes creados y luego la opción “Generar”.
3. Ubicado en la página de opciones selecciona la opción Generar Tablas en Base de Datos y luego presiona el botón “Aceptar”.
4. Debe completar el formulario indicando los datos de la base de datos en donde desea crear las tablas: IP del Servidor, Puerto, Nombre de la BD, Login y Password y presiona el botón Continuar.

**Resultados Esperados:**

Se deben crear las tablas que soportan el almacenamiento de datos en la base de datos especificada por el usuario.

**Resultados Obtenidos:**

Se crean las tablas que soportan el almacenamiento de datos en la base de datos especificada por el usuario.

En la sección 2.5 se presenta el análisis de la aplicación GenSigs la cual se encuentra constituida por los formularios que genere la aplicación GForm. Se incluye el estudio de las fases de WebML así como un caso de estudio aplicado a los formularios del componente ATM que fueron generados en la sección 2.4.1.

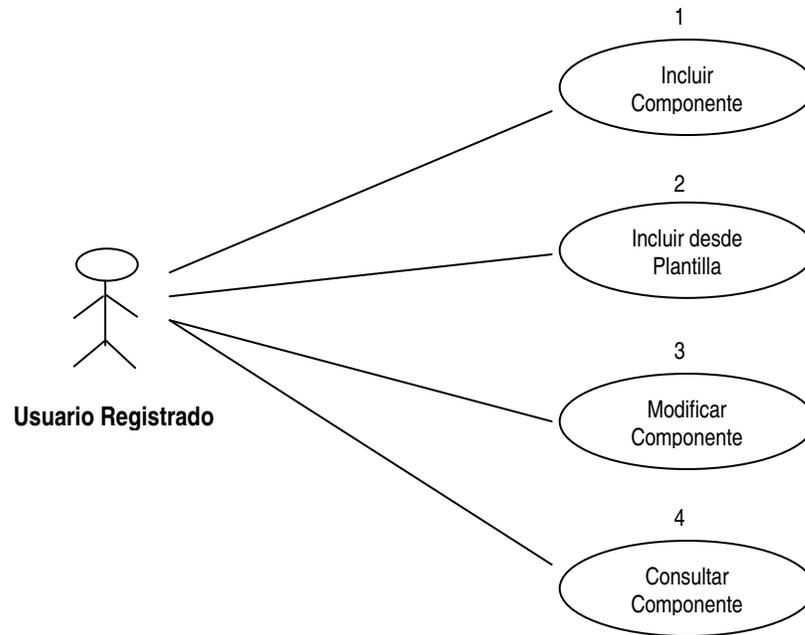
## **2.5 GenSigs. Sistema para el manejo de Componentes Cantv**

A través de este sistema el usuario puede realizar las tareas de inclusión, modificación y consulta de componentes de red. Los formularios que permiten realizar dichas tareas se generan como producto de la aplicación generadora de formularios.

### **FASE 1: Especificación de Requerimientos (GenSigs)**

#### **1. *Requerimientos Funcionales (GenSigs)***

Los procesos soportados por la aplicación se representan a través del siguiente Diagrama de Casos de Uso:



**Fig.71. Diagrama de Casos de Uso para los procesos soportados por el sistema para el manejo de componentes Cantv**

### **2.1 Especificación de los Casos de Uso (GenSigs)**

A continuación se especifican con detalle cada uno de los casos de uso del diagrama anterior:

#### **1) Incluir Componente**

Permite a un usuario agregar un componente de un tipo determinado a la base de datos.

#### **2) Incluir desde Plantilla**

Permite a un usuario agregar un componente basándose en otro que ha sido registrado anteriormente.

#### **3) Modificar Componente**

Permite a un usuario modificar los datos de un componente que se ha registrado anteriormente.

#### **4) Consultar Componente**

---

Permite al usuario visualizar los valores de un componente determinado que se encuentra registrado.

### **3. Requerimientos de Datos (GenSigs)**

El sistema realizará el manejo de datos correspondientes al nuevo componente, dichos datos dependen exclusivamente de los campos señalados por el usuario al momento de crear los formularios del mismo.

### **4. Requerimientos de Personalización (GenSigs)**

Como se indicó anteriormente los usuarios del sistema constituyen un solo grupo por lo que no existen preferencias de contenido, servicios, ni derechos de acceso.

### **5. Requerimientos de Dispositivos Específicos (GenSigs)**

No aplica ya que el sistema no será visible en múltiples dispositivos.

### **6. Requerimientos No Funcionales (GenSigs)**

- El sistema cuenta con facilidad de uso al presentar una interfaz simple con objetos distribuidos eficazmente que contribuyen al sencillo aprendizaje del uso de la misma.
- Se cuenta con un nivel de rendimiento adecuado, escalabilidad y disponibilidad, características provistas por el diseño de la arquitectura de la aplicación.

## **FASE 2: Diseño de Datos (GenSigs)**

El modelo de datos depende exclusivamente de la información que el usuario requiera almacenar para cada componente y las tablas que soporten este modelo se crearán de forma dinámica. Por esta razón no se puede definir un modelo de datos específico ya que el mismo será impredecible.

## **FASE 3: Diseño de Hipertexto (GenSigs)**

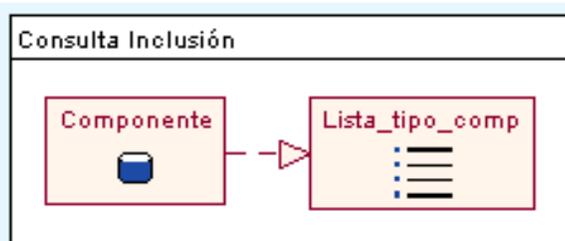
**Modelo de Composición:** las páginas que conforman el hipertexto contenido en el sitio Web son las siguientes:

- **Página Home:** contiene una unidad índice **Lista\_Oper** que presenta un listado de las operaciones sobre los componentes que se encuentran disponibles al usuario: Incluir, Incluir desde Plantilla, Modificar y Consultar. Es de tipo Home y Landmark.



*Fig.72. Página Home del Sistema para el manejo de componentes Cantv*

- **Página de Consulta para la inclusión:** es presentada al usuario luego de que el mismo escoge la opción de Incluir un nuevo Componente. Es de tipo Default ya que es la página que se carga por defecto en el Área de Inclusión y contiene una unidad índice **Lista\_tipo\_Comp** para mostrar el listado de los componentes disponibles.



*Fig.73. Página de Consulta para la inclusión del Sistema para el manejo de componentes Cantv*

- **Página de Inclusión de un Componente:** en esta se lleva a cabo la inclusión de un nuevo componente por parte del usuario. Se encuentra formada por las siguientes unidades: unidad índice **List\_datos\_comp** que contiene un listado con diferentes datos acerca del componente que podrán ser seleccionados por el usuario. Una unidad de entrada **Datos\_comp** en la que se deberán completar otros datos del componente y por último una unidad multidata **Inf\_comp** en donde se almacenarán todos los datos para el componente específico que el usuario determine en las unidades anteriores.

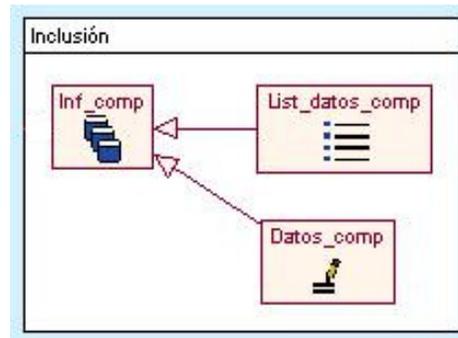


Fig.74. Página de Inclusión de un Componente del Sistema para el manejo de componentes Cantv

- **Página de Consulta General:** en esta página se realiza la búsqueda de un componente en la base de datos partiendo de la información que especifique el usuario. Es de tipo Default ya que es la página que se presenta por defecto en el Área de Inclusión desde Plantilla y Modificación. Presenta una unidad índice **Lista\_tipo\_Comp** para mostrar el listado de los componentes disponibles y una unidad de entrada **Nombre\_componente** para que el usuario indique el nombre del componente que desea buscar.

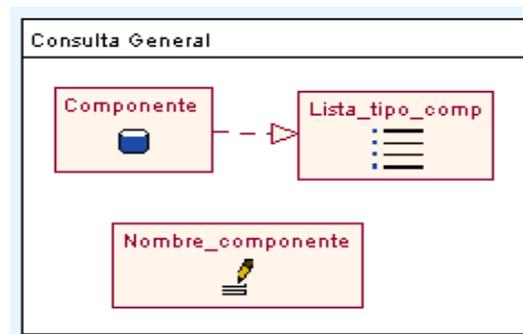


Fig.75. Página de Consulta General del Sistema para el manejo de componentes Cantv

- **Página de Selección de Componente:** página que muestra los nombres de los componentes registrados en la base de datos que cumplen con las especificaciones indicadas por el usuario en la página de Consulta General. Estos nombres de componentes son presentados a través de una unidad índice **List\_nombres\_comp**.



Fig.76. Página de Selección de un Componente del Sistema para el manejo de componentes Cantv

- **Página de Inclusión desde Plantilla y Modificación:** página mostrada al usuario una vez que ha seleccionado el componente que desea utilizar como plantilla para incluir uno nuevo o bien el que desea modificar. En ella se presenta una unidad índice **Lista\_inf\_comp** con diversas instancias pertenecientes al componente seleccionado y una unidad de entrada **Otra\_inform** en cuyos campos se presenta el resto de la información del componente. Los datos presentados por ambas unidades son obtenidos a través de una unidad multidatos **Datos\_comp** y una vez que son modificados se envían nuevamente a dicha unidad para ser almacenados.



**Fig.77. Página de Inclusión desde Plantilla y Modificación del Sistema para el manejo de componentes Cantv**

- **Página de Consulta de Componente:** esta página sirve para consultar la información de un componente específico previamente seleccionado por el usuario. En ella se presenta la información completa del componente proveniente de una unidad multidatos **Componente** la cuál podrá ser consultada por el usuario pero no modificada. Es de tipo Default ya que es la página que se carga por defecto en el área de Consulta de un Componente.



**Fig.78. Página de Consulta de Componente**

**Modelo de Navegación:** a continuación se presentan las vistas de sitio de la aplicación que permiten especificar el flujo de información necesario para procesar las unidades y operaciones:

**1) Vista de Sitio Público:** al igual que la aplicación generadora de formularios esta vista de sitio está formada por la página Login y Login Fallido. Adicionalmente contiene una unidad de operación predefinida llamada Login Unit 1 la cuál toma los valores de login y password proporcionados por el usuario y verifica su validez.

**2) Vista de Sitio Usuario Registrado:** es la vista presentada a los usuarios registrados una vez que se autentifican en el sistema, esta dividida en tres áreas fundamentales compuestas a su vez por las páginas especificadas anteriormente. Estas áreas son:

- Área de Inclusión (1)
- Área de Inclusión desde Plantilla y Modificación (2)
- Área de Consulta de un Componente (3)

En la siguiente figura se muestran gráficamente dichas áreas. Para una mejor comprensión de la navegación entre las páginas fue incluida la página Home en cada una de ellas pero esta no pertenece a un área en particular sino que forma parte de la vista de sitio para los usuarios registrados.



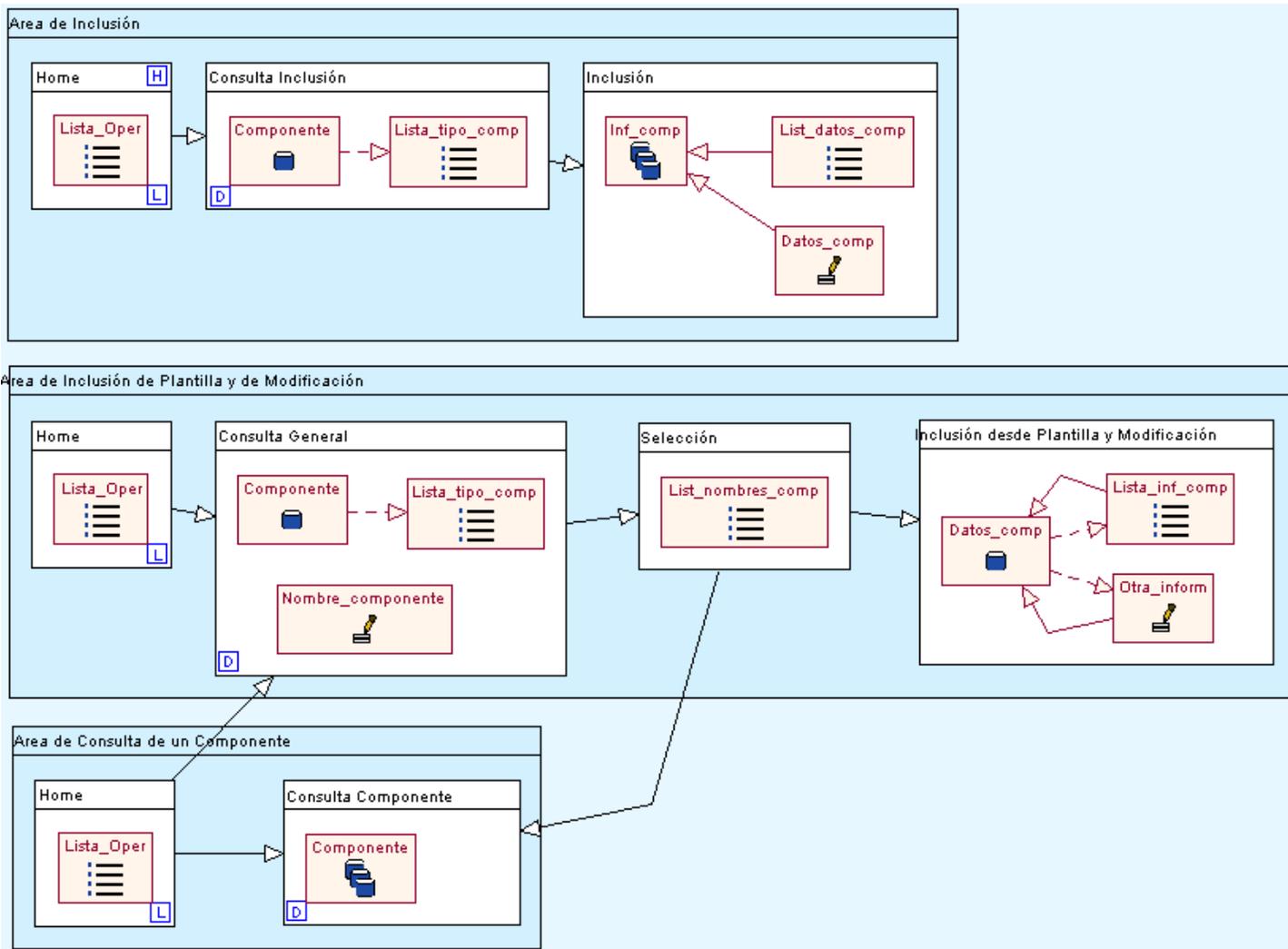
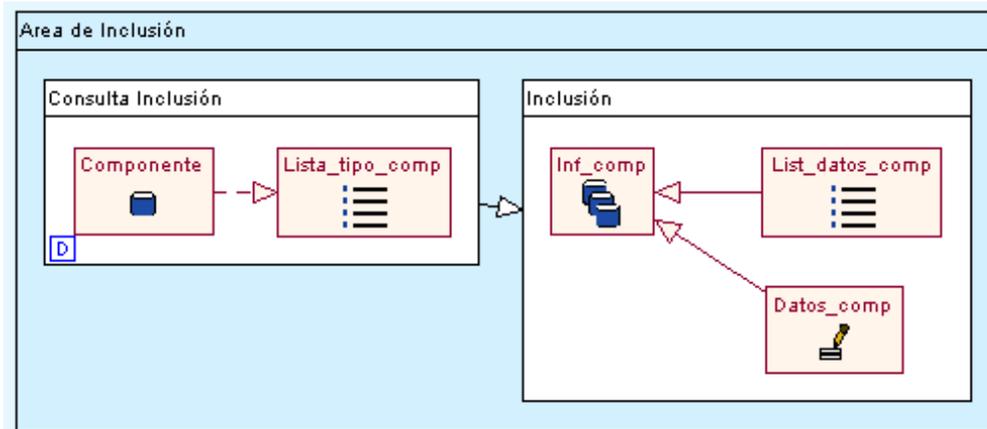


Fig.79. Vista de usuario registrado

Por su parte, las Áreas contienen páginas o sub-áreas utilizadas para organizar jerárquicamente las vistas de sitio. En este caso las áreas diseñadas están formadas únicamente por páginas. De forma individual se muestran cada una de las áreas anteriores:

- 1) **Área de Inclusión:** utilizada para organizar el hipertexto referente a la inclusión de un nuevo componente, está formada por las páginas **Consulta** e **Inclusión**. Una vez que el usuario selecciona la operación Incluir se muestra la página de Consulta y posteriormente la página de Inclusión.



**Fig.80. Área de Inclusión del Sistema para el manejo de componentes CANTV**

2) **Área de Inclusión desde Plantilla y Modificación:** en esta área se organiza el hipertexto referente a la Inclusión desde plantilla de un componente o modificación del mismo. Esta formada por las páginas **Consulta**, **Selección** e **Inclusión desde Plantilla o Modificación**.

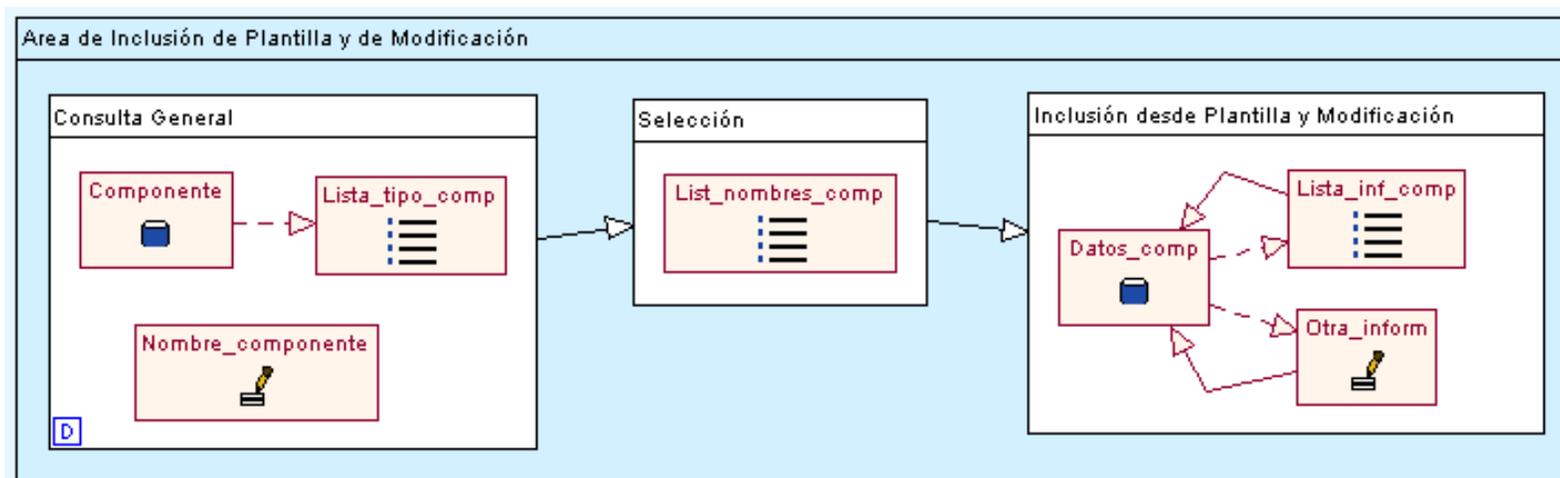
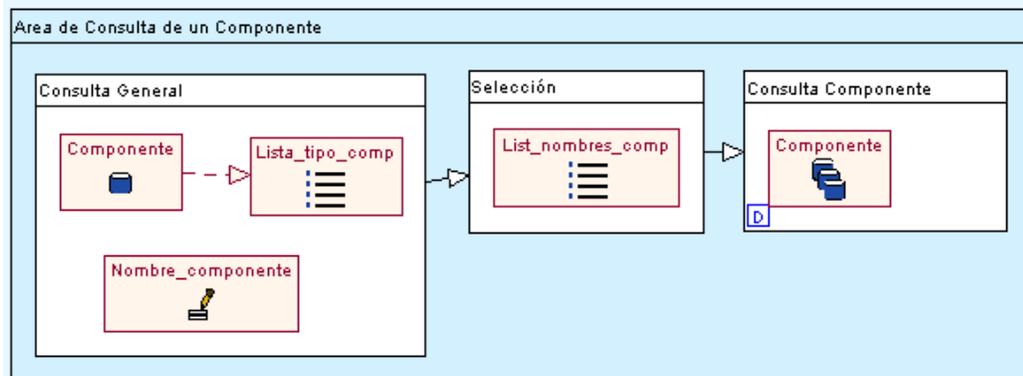


Fig.81. Área de Inclusión desde Plantilla y Modificación

3) **Área de Consulta de un Componente:** formada por la página Consulta\_componente es usada para organizar el hipertexto referente a la consulta de los datos de un componente específico.



**Fig.82. Área de Consulta de un Componente**

#### FASE 4: Diseño de Arquitectura

Este sistema presenta las mismas metas, requerimientos de diseño, escenarios de despliegue y diseño de arquitectura de la aplicación generadora de formularios, inclusive ambas aplicaciones se encuentran alojadas en los mismos servidores.

#### FASE 5: Implementación

La fase de implementación no será desarrollada debido a que gran parte de las páginas que conforman esta aplicación es generada de manera dinámica por la aplicación GForm. Por lo tanto las mismas varían y dependen de manera directa del diseño de los formularios que indique el usuario.

Es importante señalar que para poder ejecutar la aplicación generada junto con los formularios es necesario seguir los siguientes pasos:

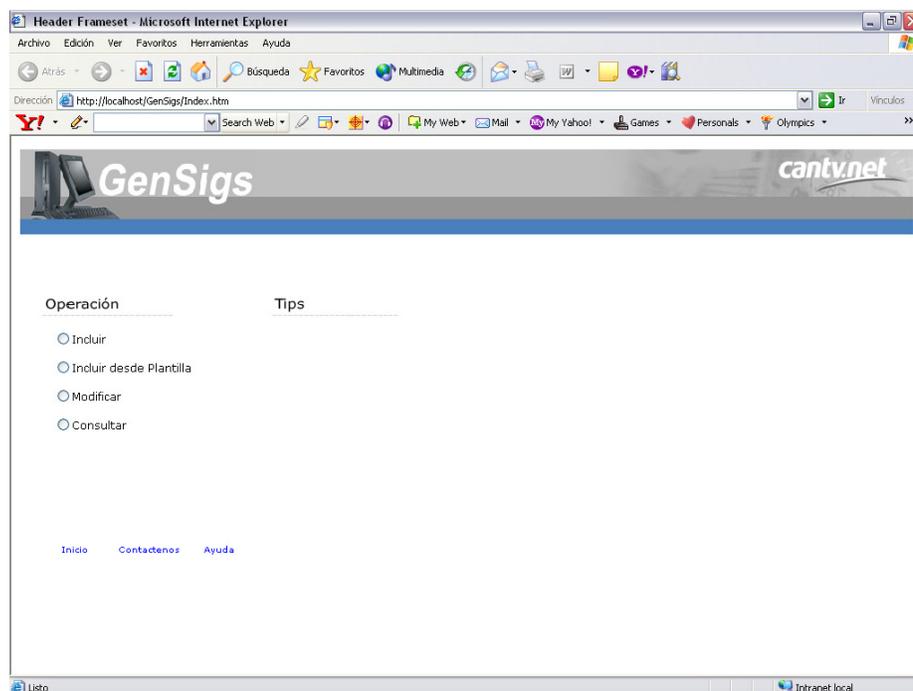
- 1) Crear un nuevo proyecto en Visual Studio.NET con el nombre GenSigs y copiar el contenido de toda la carpeta Proyecto (ya señalado en la sección 2.5) en la nueva carpeta, la cuál se encontrará en el directorio C:\inetpub\wwwroot.
- 2) Desde Visual Studio.NET incluir todos los archivos copiados al nuevo proyecto.
- 3) Se debe agregar la referencia de los WebControls **Microsoft.Web.UI.WebControls.dll**, desde la siguiente ruta: C:\Archivos de programa\IE Web Controls\build. [8]

- 4) Establecer el archivo ***Index.htm*** como Página de inicio.
- 5) Por último una vez aplicados los pasos anteriormente descritos se podrá ejecutar la aplicación.

A continuación se presenta un caso de estudio de la aplicación GenSigs en el cual se muestran los formularios generados en la sección 2.4.1 para el componente ATM. Se presenta en detalle la inclusión, modificación y consulta de un nuevo componente de este tipo.

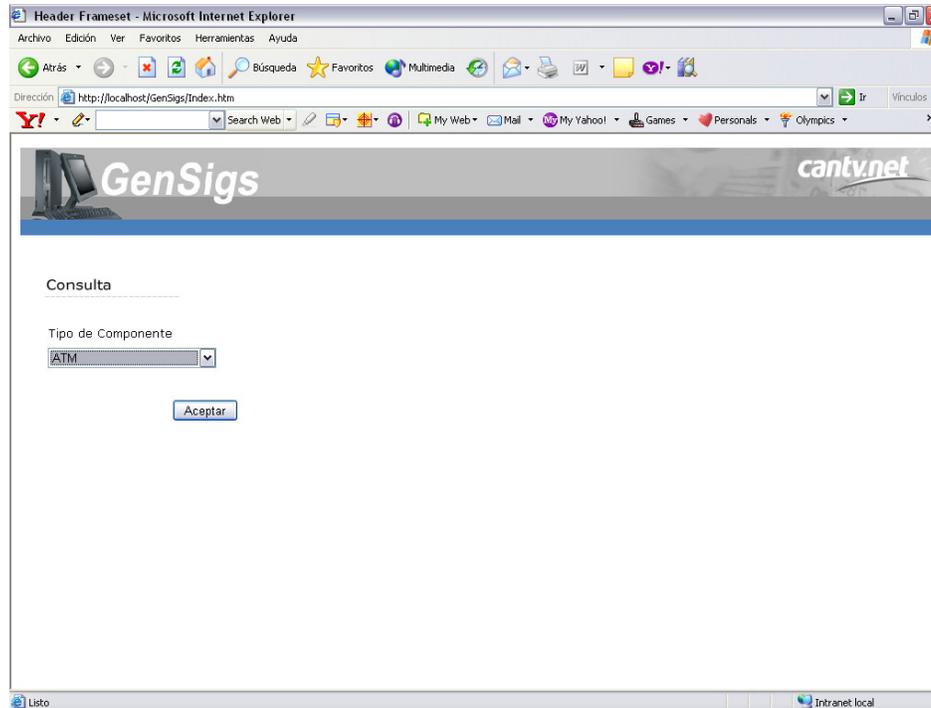
### 2.5.1 Caso de Estudio del Sistema para manejo de Componentes Cantv: GenSigs

Desde la página principal del sistema el usuario puede acceder a cualquiera de las siguientes opciones: Incluir, Incluir desde Plantilla, Modificar y Consultar. Tal como se muestra en la pantalla siguiente:



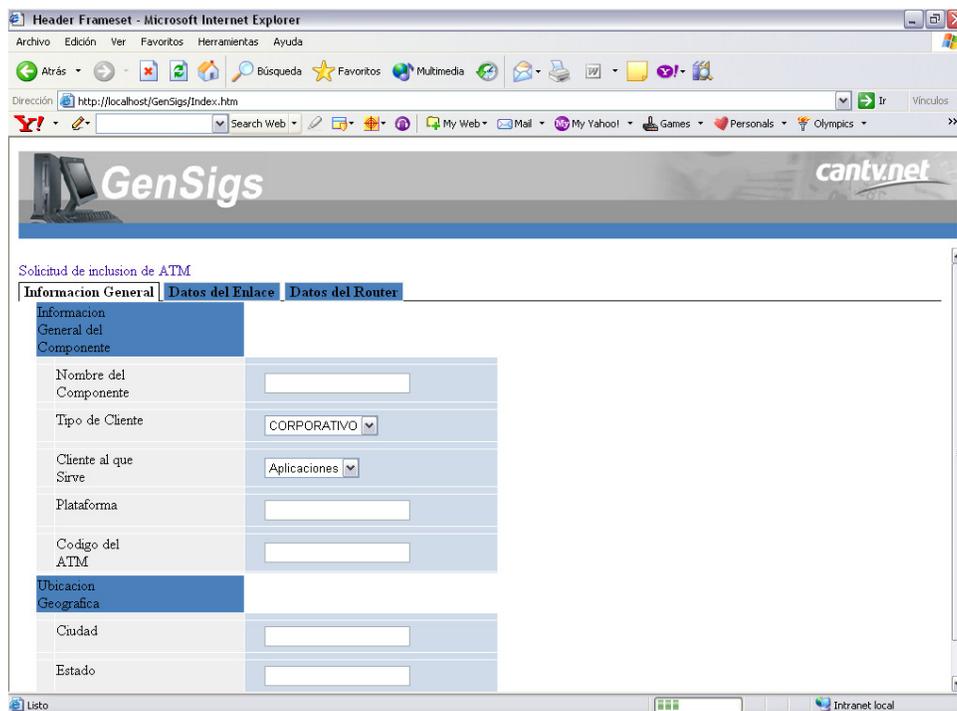
**Fig.83. Página principal del sistema GenSigs**

Al seleccionar la opción “Incluir” el sistema lleva al usuario a la siguiente página, en donde debe escoger el Tipo de Componente que desea incluir:



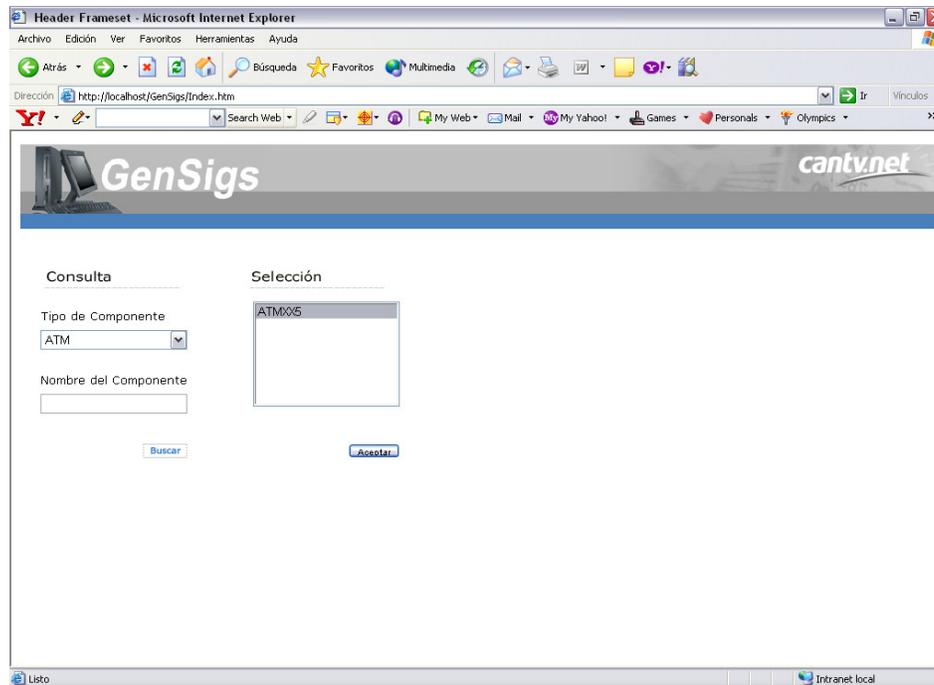
**Fig.84. Página principal de consulta del sistema GenSigs**

Al escoger ATM el sistema muestra una página que contiene un formulario para la Solicitud de Inclusión de ATM, en el cual el usuario debe llenar todos los campos y luego hacer clic sobre el botón Incluir. De esta manera se incluye un registro para el nuevo componente de tipo ATM.



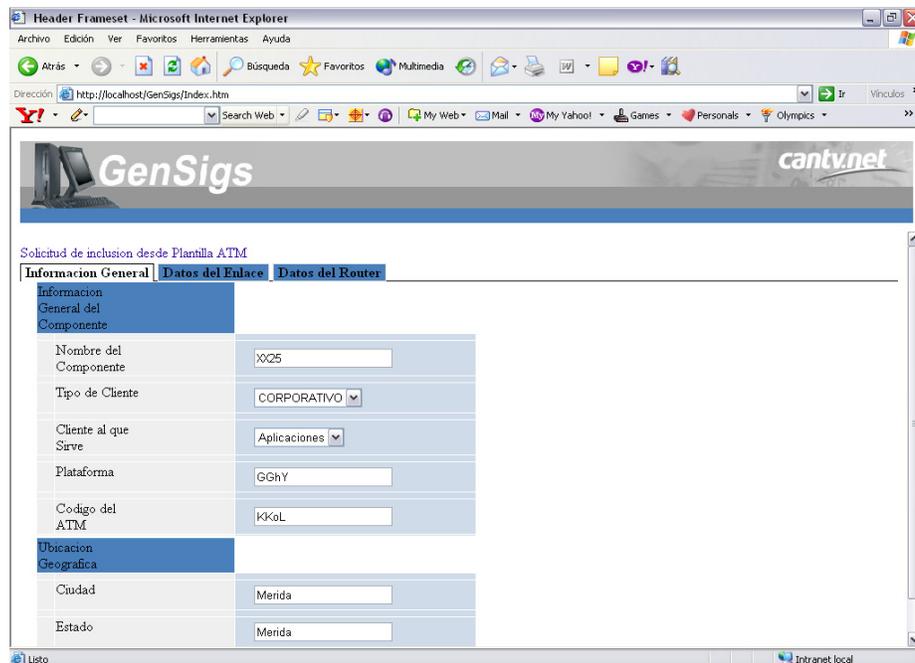
**Fig.85. Página de solicitud de inclusión del sistema GenSigs**

Si ya se han incluido otros componentes ATM anteriormente y el usuario desea tomar como plantilla uno de estos para incluir uno nuevo debido a que poseen características similares escoge la opción “Incluir desde Plantilla” del menú principal. Luego en la pantalla de Consulta que se muestra a continuación debe escoger el Tipo de Componente ATM y colocar el nombre del componente que desea utilizar como plantilla.



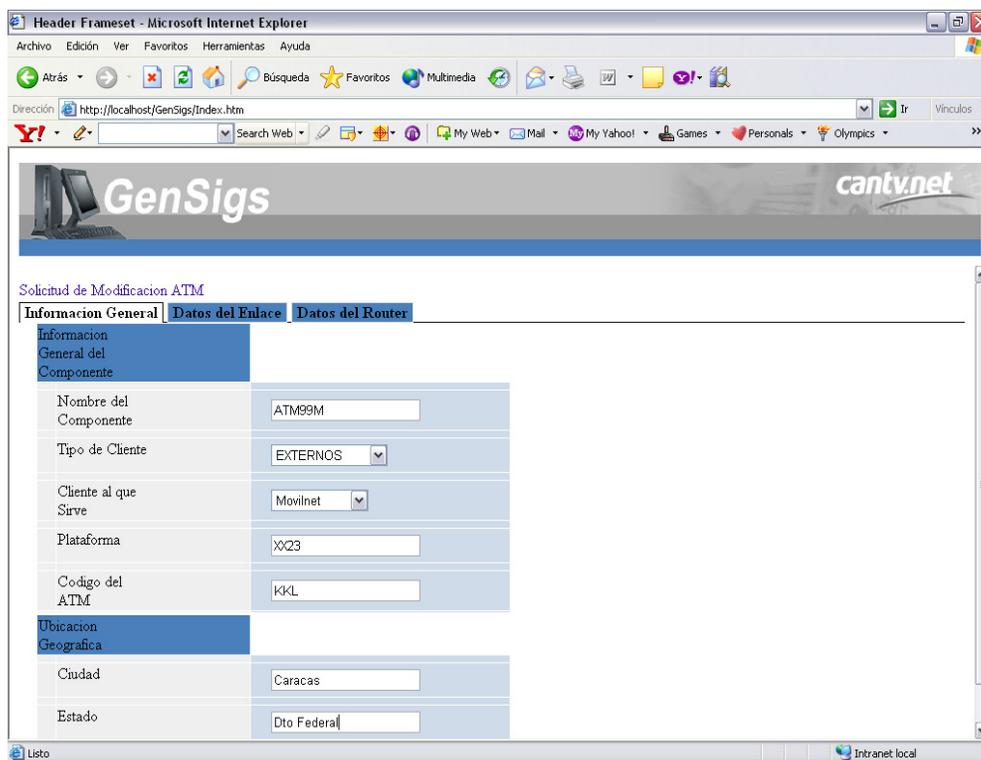
**Fig.86. Página de consulta para inclusión desde plantilla del sistema GenSigs**

Al realizar esta selección se cargara el formulario de inclusión con los valores de los campos del componente escogido como plantilla. Luego, el usuario podrá seguir incluyendo el nuevo componente cambiando el valor de los campos que sea necesario cambiar y dejando iguales los que no. Al terminar, dándole clic al botón Incluir se añadirá el nuevo componente.



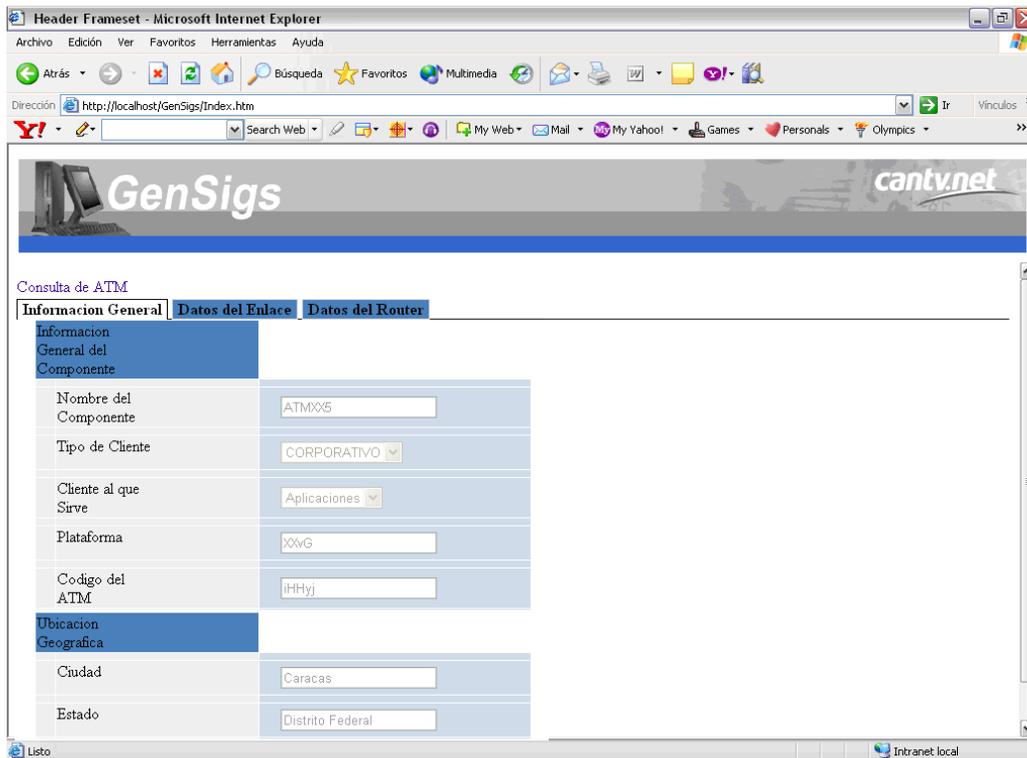
**Fig.87. Página de solicitud de inclusión de un componente ya registrado en el sistema GenSigs**

Para modificar los registros de un componente ya incluido luego de escoger la opción Modificar y de seleccionar el componente se cambian los registros que desee el usuario.



**Fig.88. Página de modificación de un componente específico del sistema GenSigs**

Finalmente, para consultar los datos de un componente ya incluido luego de escoger la opción Consultar y de seleccionar el nombre del componente se muestran al usuario los formularios con los datos del mismo.



**Fig.89. Página de consulta de un componente específico del sistema GenSigs**

---

## **FASE 6: Prueba y Evaluación (GenSigs)**

Para evaluar los requisitos funcionales de la aplicación, en base al caso de estudio del componente ATM se realizaron las siguientes pruebas:

### **Prueba 1: Incluir Componente**

Validar que el usuario pueda incluir de manera satisfactoria un componente de tipo ATM.

#### **Pasos de la Prueba:**

1. El usuario selecciona la opción Incluir desde la pagina principal del sistema.
2. Desde la página de consulta que se muestra a continuación selecciona el Tipo de Componente ATM.
3. Una vez que se muestra el formulario de inclusión llena todos los campos del mismo con los datos del nuevo componente que desea incluir.

#### **Resultados Esperados:**

Se debe registrar en el sistema el nuevo componente con todos los datos señalados por el usuario.

#### **Resultados Obtenidos:**

Es registrado en el sistema el nuevo componente con todos los datos señalados por el usuario.

### **Prueba 2: Incluir desde Plantilla**

Validar que se pueda insertar la solicitud de inclusión de un ATM tomando como punto de partida los datos de otro componente ATM que ya haya sido incluido.

#### **Pasos de la Prueba:**

1. El usuario selecciona la opción Incluir desde Plantilla de la pagina principal del sistema.
2. Desde la página de consulta que se muestra a continuación selecciona el Tipo de Componente ATM y llena el campo siguiente con el nombre del componente que desea utilizar como plantilla.

3. Presiona el botón “buscar”.
4. En la página de Selección debe aparecer el nombre del componente que cumple con el criterio de búsqueda anterior.
5. Hacer clic sobre el nombre del componente que desea tomar como plantilla para la inclusión del nuevo ATM.
6. Debe aparecer la página donde aparece la información general del ATM, en la parte superior deben aparecer un conjunto de pestañas que son enlaces directos a cada una de las partes que contienen toda la información del ATM.
7. Modificar los campos en los cuales desea cambiar la información.

**Resultados Esperados:**

Se debe registrar en el sistema el nuevo componente con todos los datos señalados por el usuario.

**Resultados Obtenidos:**

Se registra en el sistema el nuevo componente con todos los datos señalados por el usuario.

**Prueba 3: Modificar Componente**

Validar que el usuario pueda modificar satisfactoriamente los datos de un componente ATM específico ya incluido en el sistema.

**Pasos de la Prueba:**

1. El usuario selecciona la opción Modificar desde la página principal del sistema.
2. Desde la página de consulta que se muestra a continuación selecciona el Tipo de Componente ATM y llena el campo siguiente con el nombre del componente que desea modificar. Presiona el botón “buscar”.
3. Una vez que se muestran los formularios con los datos específicos del componente correspondiente modifica los que desee y presiona el botón “Modificar”.

**Resultados Esperados:**

Los datos del componente ATM específico deben ser modificados según como lo haya señalado el usuario.

**Resultados Obtenidos:**

Se modifican los datos del componente ATM específico según como lo señaló el usuario.

**Prueba 4: Consultar Componente**

Validar que el usuario pueda consultar los datos de un componente ATM específico ya incluido en el sistema.

**Pasos de la Prueba:**

1. El usuario selecciona la opción Consultar desde la página principal del sistema.
2. Desde la página de consulta que se muestra a continuación selecciona el Tipo de Componente ATM y llena el campo siguiente con el nombre del componente que desea modificar. Presiona el botón “buscar”.
3. En la página de Selección debe aparecer el nombre del componente que cumple con el criterio de búsqueda.
4. Hacer clic sobre el nombre del componente que se desea consultar.
5. Debe aparecer la página donde aparece la Información General del Componente, en la parte superior deben aparecer un conjunto de pestañas que son enlaces directos a cada una de las partes en que ha sido clasificada toda la información del ATM.
6. Se debe verificar que la información que se está presentando es la misma a la incluida por el usuario.

**Resultados Esperados:**

Presentar la información completa del ATM, la cual debe ser igual a la insertada por el usuario.

**Resultados Obtenidos:**

Se muestra la información completa del ATM, la cual es igual a la insertada por el usuario.

## **CONCLUSIONES, RECOMENDACIONES Y APORTES DE LA INVESTIGACIÓN**

El presente Trabajo Especial de Grado cumplió con el objetivo de crear un sistema que permite a los usuarios generar de manera dinámica formularios orientados específicamente a la gestión de componentes de red, así como las estructuras de datos que permiten el procesamiento de los mismos.

Para la creación de este sistema se siguió el proceso de desarrollo WebML el cual incluye varias fases, lo que conllevó a un desarrollo incremental e iterativo, donde se construyeron los diagramas o modelos de diseño que ayudaron en la implementación de la solución planteada. WebML a diferencia de otros procesos de desarrollo de software esta orientado específicamente a la creación de sitios Web lo cual proporciona un conjunto de ventajas que otras metodologías no aportarían en este escenario, entre las cuales se pueden mencionar:

- Expresión de la estructura de una aplicación web a un alto nivel de descripción lo cuál facilita la evolución y el mantenimiento.
- Provee múltiples vistas del mismo contenido.
- Almacenamiento de la información generada durante el proceso de diseño, la cuál puede ser usada durante el tiempo de vida de la aplicación para la generación dinámica de las páginas que componen el sitio.
- Separación de la información de contenido y de la presentación.
- Tiene su propio software de construcción de los diagramas llamado WebRatio, lo cual facilita y agiliza la elaboración del diseño.

El uso de documentos XML para estructurar toda la información concerniente al diseño de los formularios creados por el usuario proporcionó la flexibilidad y facilidad necesarias al momento de la interpretación de los mismos. Dicha interpretación se llevó a cabo a través de XSLTs que permitieron la creación de las estructuras planteadas.

A partir de los documentos XML contruidos y modificando los XSLTs se pueden generar nuevas aplicaciones.

### **Recomendaciones**

Se recomienda incluir mejoras o nuevas funcionalidades al sistema, entre las que se sugieren:

- Incluir en los formularios de manera obligatoria otros campos adicionales al nombre del componente que permitan una mayor precisión en la búsqueda del mismo.
- Permitir el almacenamiento de la información en cada pestaña del formulario generado para evitar su pérdida en caso de fallas en el sistema al momento de incluir o modificar un componente.
- Realizar una plantilla de la estructura de diseño del tipo de componente del cual se desean crear los formularios Web, para de esta forma facilitar esta tarea.
- Mejorar la interfaz y navegación de la aplicación a través de la cual el usuario diseña los formularios deseados por una interfaz que permita hacer este trabajo de una manera más fácil.
- Incluir en los formularios generados las validaciones respecto a la entrada de datos en los controles.
- Realizar un tratamiento para que sean tomadas en cuenta letras acentuadas, ya que por el momento en las palabras acentuadas no se muestran los acentos.

### **Aportes de la Investigación**

- Con la utilización de WebML como proceso de desarrollo y lenguaje de modelación se contribuye a dar a conocer más ampliamente el mismo así como las ventajas de utilizarlo en el diseño y modelación de sitios web, ya que por sus características ofrece mayor precisión y un alto nivel de expresión de aplicaciones web con respecto a otros.
- Con el desarrollo de este trabajo especial de grado, se contribuye con respecto a los generadores dinámicos de código .aspx, pero más que eso a los generadores de aplicaciones, ya que este generador permite la creación de una aplicación conjuntamente con las estructuras físicas que soportan el almacenamiento de sus datos.
- Se contribuye a la utilización de la tecnología ASP.NET para la elaboración de generadores dinámicos de código .aspx.
- La utilización de XSLTs para la transformación de XMLs, incluyendo en los mismos código HTML así como también código Visual Basic y consultas SQLs que conforman los archivos a generar.

## BIBLIOGRAFIA Y REFERENCIAS

[1]	Ambler, Scott W. <b>"The Design of Robust Persistence Layer For Relational Databases"</b> 2003.
[2]	Alonso Pulido, Esther M <sup>a</sup> , Arteaga Domínguez, Octavio Peñate y Jiménez, Cristina <b>"Tutorial de HTML"</b> Facultad de Informática de la Universidad de Las Palmas de Gran Canaria. <a href="http://gias720.dis.ulpgc.es/Gias/Cursos/Tutorial_html/indice.htm#inicio">http://gias720.dis.ulpgc.es/Gias/Cursos/Tutorial_html/indice.htm#inicio</a>
[3]	<b>"Aplicaciones ASP.NET"</b> <a href="http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/cpguide/html/cpconASPNETApplications.asp">http://msdn.microsoft.com/library/spa/default.asp?url=/library/SPA/cpguide/html/cpconASPNETApplications.asp</a>
[4]	Ceri Stefano, Piero Fraternali, Bongio Aldo, Brambilla Marco, Comai Sara, Matera Maristella <b>"Designing Data- Intensive Web Applications"</b> San Francisco, USA 2003
[5]	<b>"Diccionario de Términos Informáticos"</b> <a href="http://www.moheweb.galeon.com/diccinformatic.htm">http://www.moheweb.galeon.com/diccinformatic.htm</a>
[6]	<b>"Guía Breve de Tecnologías XML"</b> <a href="http://www.w3c.es/divulgacion/guiasbreves/TecnologiasXML">http://www.w3c.es/divulgacion/guiasbreves/TecnologiasXML</a>
[7]	<b>"JavaScript"</b> <a href="http://members.lycos.co.uk/morocho00/java/queesjava.htm">http://members.lycos.co.uk/morocho00/java/queesjava.htm</a>
[8]	<b>"Internet Explorer Web Controls"</b> <a href="http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art178.asp">http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art178.asp</a>
[9]	<b>Java y Aplicaciones sobre Internet</b> 2004 <a href="http://www.linti.unlp.edu.ar/catedras/Java%20y%20Aplicaciones%20sobre%20internet/2004/Teorias/CLASE%2010-jsp+javabeans.pdf">http://www.linti.unlp.edu.ar/catedras/Java%20y%20Aplicaciones%20sobre%20internet/2004/Teorias/CLASE%2010-jsp+javabeans.pdf</a>
[10]	Hista de Venezuela <b>Administración de Requerimientos con Casos De Uso</b> Hista de Venezuela, Venezuela 2002
[11]	<b>Pauta de presentación de Tesis</b> Sistema de servicio de información Universidad de Chile Junio 2003 <a href="http://www.uchile.cl/bibliotecas/servicios/tesis.pdf">http://www.uchile.cl/bibliotecas/servicios/tesis.pdf</a>
[12]	Perez, Argenis <b>"Análisis diseño y desarrollo del sistema de gestión de las solicitudes de monitoreo de los componentes en la plataforma de monitoreo de Cantv.net(SIGS-WEB)".</b> UCV, Facultad de Ingeniería, Escuela de Ingeniería Eléctrica Caracas 2003.
[13]	<b>"¿Que es Asp.net ?"</b> <a href="http://www.intermtty.com/html/que_es_asp_net.html">http://www.intermtty.com/html/que_es_asp_net.html</a>
[14]	<b>"¿Qué es ASP.NET?"</b>

---

---

	<a href="http://es.gotdotnet.com/quickstart/aspplus/doc/whatisaspx.aspx">http://es.gotdotnet.com/quickstart/aspplus/doc/whatisaspx.aspx</a>
[15]	<b>"The Source for Developers"</b> 2004 <a href="http://java.sun.com/products/jta/">http://java.sun.com/products/jta/</a>
[16]	Tito, Miguel <b>"Información general de ADO.NET"</b> <a href="http://www.monografias.com/trabajos14/informe-ado-net/informe-ado-net.shtml">http://www.monografias.com/trabajos14/informe-ado-net/informe-ado-net.shtml</a>
[17]	<b>"WebML User Guide version 3.0"</b> Politécnico de Milano 2001 <a href="http://webml.org">http://webml.org</a>
[18]	<b>"WebRatio"</b> <a href="http://www.webratio.com/sv1.do">http://www.webratio.com/sv1.do</a>