

EL SISTEMA DE ADMINISTRACIÓN GENÉRICO DE BASE DE DATOS

FREDDY LEANDRO ANGARITA CASTELLANOS

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2004

EL SISTEMA DE ADMINISTRACIÓN GENÉRICO DE BASE DE DATOS

FREDDY LEANDRO ANGARITA CASTELLANOS

**Proyecto de Grado
para optar al título de Ingeniero de Sistemas**

Director

JORGE HERRERA CASTILLO

Ingeniero de Sistemas

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2004

AGRADECIMIENTOS

Alcanzar una meta, disfrutarla y luego buscar metas más altas es lo que se aprende tras alcanzar una tan importante, aprender y llegar al conocimiento a través de la práctica y de guías dedicados inspiran a seguir adelante. Gracias a quienes me apoyaron alcanzar este logro

A Dios quien siempre a nuestro lado

A mis padres, quienes con su apoyo y continuo impulso me ayudaron a desarrollo de este proyecto

Al Ing. Jorge Herrera, quien con sus aportes y consejos fue posible la realización de este proyecto

A todos los profesores involucrados en mi formación como Ingeniero de Sistemas

A Mireya, quien me apoyo y estuvo a mi lado la mayor parte de mi carrera

A mis compañeros de universidad quien con su amistad y conocimientos me apoyaron en el desarrollo de este proyecto y durante toda mi carrera

CONTENIDO

	Pág.
INTRODUCCIÓN	16
2. ASPECTOS GENERALES	18
2.1 OBJETIVOS	18
2.1.1 Objetivo general	18
2.1.2 Objetivos específicos	18
2.2 JUSTIFICACIÓN	19
2.2.1 Antecedentes del problema	19
2.2.2 Descripción de objetivos	21
2.3 IMPACTO Y VIABILIDAD	22
2.3.1 Impacto	22
2.3.2 Viabilidad	22
3. MARCO DE TRABAJO	24
3.1 Dominio del problema	24
3.1.1 Marco Social	25
3.1.2 Marco institucional	25
3.1.3 Marco Conceptual	25

3.1.3.1 Plataformas informáticas	26
3.1.3.2 Las aplicaciones Web	27
3.1.3.3 Desarrollo por capas	30
3.2 Dominio de la solución	32
3.2.1 Marco Social	32
3.2.2 Marco Conceptual	32
3.2.2.1 Metamodelado	32
3.2.2.2 Modelo de tres capas	33
3.2.2.3 Metodología de desarrollo	33
3.2.3 Marco tecnológico	37
3.2.3.1 Tecnologías de desarrollo	38
3.2.3.2 Manejador de Bases de datos	38
4. DESARROLLO	40
4.1 INTRODUCCIÓN	40
4.2 GESTIÓN DE MODELADO	41
4.2.1 Especificación de Requisitos	41
4.2.1.1 Definiciones, siglas y abreviaciones	42
4.2.1.2 Requisitos preliminares	47
4.2.1.3 Modelo de Casos de uso	48
4.2.1.4 Requisitos Adicionales	55
4.2.1.5 Identificación de riesgos	58

4.3 MODELADO DE DATOS	60
4.3.1 Modelo de Análisis	60
4.3.1.1 Análisis de los Caso de Uso	60
4.3.2 Descripción de datos	63
4.3.2.1 Entidad de datos	65
4.4 MODELADO DEL PROCESO	68
4.4.1 Modelo de Diseño	68
4.4.1.1 Modelo de Casos de Uso de diseño	68
4.4.2 Modelo de Despliegue	83
4.5 GENERACIÓN DE LA APLICACIÓN	85
4.5.1 Modelo de Interfases	86
4.5.1.1. Interfaces de la herramienta	87
4.6. PRUEBAS Y ENTREGA	93
4.6.1 Meta modelado	94
4.6.2 Sistema de anexos	94
4.6.3 Validación de datos	94
4.6.4 Multilenguaje	96
4.6.5 Capa de datos	96
4.6.6 Permisos y usuarios	96
4.6.7. Creación del sistema de enlaces	97
5. CONCLUSIONES	98

6. PERSPECTIVAS DE LA HERRAMIENTA	100
6.1 DESARROLLO ORIENTADO A OBJETOS	100
6.2 SEGURIDAD	100
6.3 HERRAMIENTA EDUCATIVA	101
BIBLIOGRAFIA	102
ANEXOS	104

LISTA DE TABLAS

	Pág.
Tabla 1. Tecnologías de desarrollo Web del lado del servidor.	28
Tabla 2. Tecnologías de desarrollo Web del lado del cliente mundialmente reconocidas.	29
Tabla 3. Descripción de las etapas de desarrollo de la metodología DRA.	35
Tabla 4. Resumen del manejo de validaciones del sistema	57
Tabla 5. Casos de Uso a Implementar	69
Tabla 6. Perspectivas de la herramienta en cuanto a metamodelado.	95

LISTA DE FIGURAS

	Pág.
Figura 1. Diseño tradicional de Aplicaciones Administrativas de base de datos	20
Figura 2 . Secuencia de envío de información hacia el servidor de base de datos.	27
Figura 3. Arquitectura de Tres capas. Caso más general de arquitectura multicapas.	30
Figura 4 . Modelo de Desarrollo de Aplicaciones Rápidas (DRA).	34
Figura 5. Flujos de trabajo que tienen lugar en las cuatro fases del Proceso Unificado y disciplinas de soporte asociadas.	37
Figura 6. Representación de la característica de la búsqueda en profundidad para la tabla Ciudad.	43
Figura 7. Representación de una relación muchos a muchos para un modelo relacional de bases de datos.	44
Figura 8. Representación de la relación entre las tablas de modo uno a muchos para un modelo relacional de bases de datos.	45
Figura 9. Casos de uso definidos para los actores Público, Cliente y Administrador en la etapa de Gestión del Modelado.	56
Figura 10. Diagrama de actividades para el CU- Definir Permisos.	61
Figura 11. Diagrama de actividades para el CU- Realizar Metamodelado.	62
Figura 12. Parte del Modelo Entidad Relación de la Herramienta para fines de Búsquedas Especializadas.	66
Figura 13. Parte del Modelo Entidad Relación de la Herramienta para la vista de análisis.	66
Figura 14. Parte del Modelo Entidad Relación de la Herramienta	66

para la agrupación de campos.

Figura 15. Parte del Modelo Entidad Relación de la Herramienta que permite la administración de Usuarios y Grupos.	67
Figura 16. Parte del Modelo Entidad Relación de la Herramienta para fines de modelado de datos.	67
Figura 17. Parte del Modelo Entidad Relación de la Herramienta que almacena las características de tablas y columnas (campos).	67
Figura 18. Modelo de casos de uso completo para la etapa de Modelado del Proceso.	84
Figura 19. Diagrama de despliegue del modelo de diseño elaborado para la implementación del sistema.	85
Figura 20. Página de ingreso al Módulo Administrativo GenApp	86
Figura 21. Menú principal del sistema.	87
Figura 22. Formulario del sistema en el cual se observa el listado de grupos de usuarios.	87
Figura 23. Formulario del sistema para la Administración de permisos para un usuario hipotético: Administrador.	87
Figura 24. Formulario del sistema para la Administración de permisos para un usuario hipotético: Cliente.	88
Figura 25. Formulario del sistema para crear, modificar o eliminar Grupos de Usuario.	89
Figura 26. Acceso directo para ver la información de los Usuarios del Grupo.	89
Figura 27. Ventana del sistema en donde se listan los Grupos de Usuario.	90
Figura 28. Ventana del sistema en donde se detallan la información personal y de ingreso de los Usuarios.	90
Figura 29. Ventana del sistema de Administración de Tablas.	91
Figura 30. Ventana del sistema correspondiente a la configuración	92

de los Nombres Amigables de las Tablas.

Figura 31. Ventana del sistema correspondiente a la configuración de los Nombres Amigables de los Campos. 93

Figura 32. Ventana del sistema correspondiente a la Administración de tipos. 94

LISTA DE ANEXOS

	Pág.
ANEXO A. Características de los datos de entrada al sistema	104
ANEXO B. Características hardware y software necesarias para el correcto funcionamiento del sistema	107
ANEXO C. Descripción de la configuración de los directorios del sistema	111
ANEXO D. Características de las consultas realizadas por el sistema.	115
ANEXO E. Generación de enlaces para la publicación final de la aplicación Web	118
ANEXO F. Detalles de la integración de la herramienta con la base de datos	120
ANEXO G. Metodología para la creación de búsquedas full text-search	125

TÍTULO:

SISTEMA DE ADMINISTRACIÓN GENÉRICO DE BASE DE DATOS*

AUTOR:

ANGARITA CASTELLANOS, Freddy Leandro**

PALABRAS CLAVES:

Sistemas de información, meta modelado, generación de formularios, SQLServer, Manejador de base de datos, catálogo del sistema, lenguaje de definición de datos, lenguaje de manipulación de datos, Active Server Pages, ASP, páginas Web, intranet, Internet, análisis de consultas, javascript, HTML, DHTML, bases de datos, modelos de datos, tablas, análisis de relaciones, diseño de tablas.

DESCRIPCIÓN O CONTENIDO:

El sistema desarrollado puede ser utilizado como herramienta de apoyo para el crear sitios Web. El sistema esta diseñado para generar dinámicamente formularios de entrada de datos y listados, basado en el diseño de la base de datos a la vez que cuenta con sistema especializado de búsquedas para generar los reportes deseados por el usuario final. El sistema cuenta con técnicas de metamodelado para aumentar la experiencia del usuario final, esta característica es la que hace posible crear sitios Web a partir del modelo de datos; Adicionalmente, el sistema cuenta con un sistema de permisos con el cual se puede establecer niveles de acceso de la información basado en permisos y orientado a tablas. Para disminuir la complejidad de los modelos se cuenta con un administrador de tipos para crear entradas sencillas asociadas a tipos de datos enteros, también se tienen agrupaciones de campos y de tablas asociadas (tablas intermedias) para presentar la información al usuario final de manera ordenada. El sistema funciona mediante el análisis de las tablas del sistema, las relaciones uno a muchos y muchos a muchos, las entradas de validación y las características de cada campo de la base de datos lo cual lo hace flexible para recibir cambios en el modelo cuando el administrador del sistema lo considere necesario.

El sistema fue desarrollado utilizando como base el manejador de base de datos SQLServer, el cual proporciona los servicios y la metodología de programación suficiente para elaborar un análisis sobre el modelo y permitir la generación de los elementos anteriormente mencionados. Se uso ASP 3.0 como lenguaje de programación.

* Proyecto de Grado

** Facultad de Ingenierías Físico-Mecánicas. Ingeniería de Sistemas e Informática. Director del proyecto: Ingeniero JORGE HERRERA CASTILLO

TITLE

GENERIC DATABASE ADMINISTRATION SYSTEM*

AUTHOR

ANGARITA CASTELLANOS, Freddy Leandro**

KEY WORDS

Information Systems, Metamodeling, forms generation, SqlServer, Database manager, system catalog, data definition language, data manipulation language, Active Server Pages, ASP, Web Pages, Intranet, Internet, Query analysis, Javascript, HTML, DHTML, Databases, data modeling, tables, relationship analysis, table design.

DESCRIPTION OR CONTENTS

The developed system can be used as support tool to develop Web sites. The system is designed to dynamically spawn data input forms and listings, based on the database design, simultaneously that count with a specialized system of searches to generate the reports wished by the end user. The system counts on techniques of meta modeling to increase the end user experience, this characteristic is the one that does possible to create Web sites from the data modeling; Additionally, the system counts on a system of permissions with which it is possible to be established access levels for the information based on profiles and it is oriented to tables. In order to diminish the complexity of the models it is has an administrator of types to create simple entrances to integer data types, also groupings of fields and associate tables (intermediate tables) to present the information to the end user of ordered way. The system works by means of the analysis of the system table, relations one to many and many to many, the check constraints and the characteristics of each field of the data base which makes flexible to receive changes in the model when the system administrator considers it necessary.

The system was developed using as it bases the SQLServer database manager system, which provides the services and the programming methodology sufficient to elaborate an analysis on the model and to allow the generation of the elements previously mentioned. ASP 3,0 was used as programming language.

* Graduation Project

** School of System Engineering and Computer science. Faculty of Sciences physicist-mechanics. Project Director: Engineer JORGE HERRERA CASTILLO

INTRODUCCIÓN

Una de las principales características de las empresas de desarrollo de software es la búsqueda del aumento de la competitividad en sus productos, sobre todo en un mundo cambiante, en donde los sistemas de información se vuelven cada vez más importantes en las empresas en todo el mundo. Por ello se requieren herramientas que aporten al desarrollo rápido, efectivo y eficiente de aplicaciones. Esto se logra a través de componentes de reutilización que permitan solucionar problemas comunes a la hora de acceder a la información; para este caso en particular, se hace referencia a las operaciones de administración de bases de datos vía Internet.

En el presente documento se expone un prototipo de herramienta que puede llegar a convertirse en alternativa productiva en el desarrollo de sistemas de información a gran escala por dos razones principalmente: la primera, porque está orientada al uso de servicios Web y de datos que proporcionan los servidores de aplicaciones; y la segunda, porque permite a los desarrolladores reutilizar programación con facilidad para acelerar el proceso de desarrollo, así como también dar acceso a los desarrolladores novatos a la funcionalidad compleja. Estos componentes se integran imperceptiblemente, mediante un proceso sencillo de configuración, en el entorno de tiempo de diseño, con el fin de facilitar el desarrollo y el acceso a datos en una aplicación Web.

Las razones expuestas, de manera más amplia, se relacionan con la construcción de modelos de datos a gran escala, en donde se representa información compleja mediante una gran cantidad de tablas, relaciones, validaciones y control de usuarios, los cuales tienen que ser manejados directamente por el programador cada vez que se desarrolla una aplicación Web. El fin de la herramienta es, entonces, aligerar ese trabajo, que muchas veces se convierte en difícil, tedioso y repetitivo. También es importante anotar que durante las etapas de desarrollo los modelos de datos sufren transformaciones, debido a la inclusión de nuevos requerimientos o a errores en los diseños iniciales, lo cual muchas veces implica modificar grandes porciones de código, lo que conduce a aumentar los costos de un proyecto.

El Sistema de Administración Genérico de Base de datos, permite al administrador manejar una capa creada sobre el modelo de datos diseñado en el manejador de base de datos (SQLServer). Esta capa está orientada a la administración de los datos contenidos en las diferentes tablas de usuario, mediante el análisis de tablas, las relaciones entre ellas y las validaciones definidas. Adicionalmente, permite buscar, incluir anexos, incluir enlaces, manejar permisos y, en esencia, crear una interfase de manera rápida entre el usuario final, el modelo de la base de datos y la información del cliente.

Es de resaltar que los elementos utilizados en este proyecto son en su mayoría producto de la experiencia personal en el desarrollo de sistemas de información orientados a Internet y del estudio minucioso de las capacidades y beneficios del manejador de base de datos SQLServer, el cual es ampliamente usado en el medio colombiano, debido a su alto grado de confiabilidad, eficiencia, efectividad y además a los costos de licencia¹.

¹ Licencia Microsoft SQL Server 2000 Desktop Engine (MSDE 2000). Disponible en la Web: <http://search.microsoft.com/search/results.aspx?View=msdn&st=a&qu=msde&c=4&s=2>

2. ASPECTOS GENERALES

Los aspectos generales comprenden la presentación del proyecto de grado en términos de los objetivos perseguidos y del análisis del entorno que condujo a la problemática que se desea solucionar a partir de la futura implantación de la herramienta que se ha diseñado e implementado, denominada Sistema Genérico de Administración de Bases de Datos.

2.1 OBJETIVOS

2.1.1 Objetivo general. El objetivo general de la propuesta de grado es diseñar e implementar un prototipo de un sistema genérico de administración de bases de datos para el apoyo de los procesos de desarrollo de sistemas de información aplicando algunas técnicas de meta modelado.

2.1.2 Objetivos específicos. El sistema estará en capacidad de:

- Permitir la generación dinámica de formularios para la administración de registros y listados para la visualización de la información contenida en una tabla específica de las Bases de datos diseñadas en Microsoft SqlServer2000, utilizando tecnología ASP, vía Internet o Intranet.
- Incorporar meta información a un modelo de datos genérico, para permitir la generación dinámica de formularios y la administración de la base de datos por el usuario.
- Utilizar la arquitectura de tres capas para estructuras de datos, para incrementar la escalabilidad y hacer la aplicación más fácil de actualizar, sin afectar el correcto funcionamiento de la misma.

2.2 JUSTIFICACION

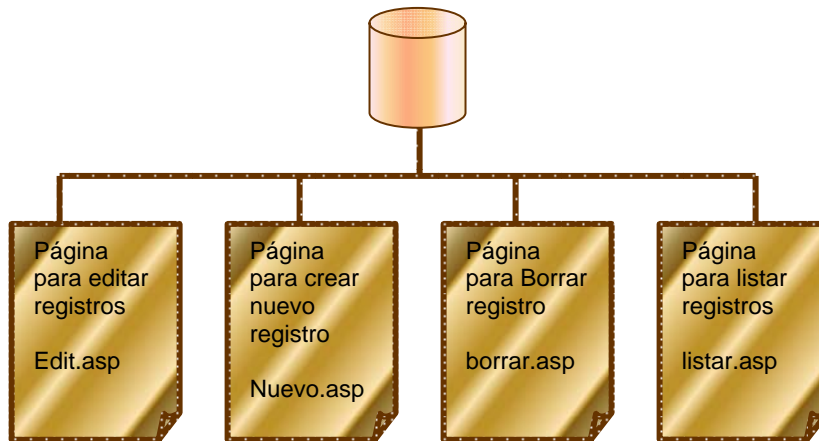
2.2.1 Antecedentes y descripción del problema. La experiencia en el diseño e implementación de sistemas de información y, en especial, aquellos a ser usados como aplicaciones para Internet, pone siempre al descubierto un problema que es común a todos los proyectos de este tipo. Normalmente las aplicaciones para Internet requieren el desarrollo de páginas individuales para la administración de la información contenida en determinada tabla de la base de datos; por lo tanto, el grupo desarrollador se enfrenta al hecho de tener que construir varias páginas que le permitan realizar operaciones básicas sobre los registros de una tabla, como lo son: crear un nuevo registro, editar uno existente, eliminarlo (ya sea en forma de cascada² o no); así como las operaciones de listado, paginación de resultados y generación de listados. Actualmente, la mayoría de los desarrollos de este tipo de sistemas de información, implementan de manera manual estas zonas administrativas.

El problema radica en el tiempo de desarrollo y en los cambios realizados en los requerimientos del sistema a mediados del proceso de construcción de la aplicación o, en ocasiones, en su finalización. Cuando existe la necesidad de incluir un cambio tan pequeño como el de modificar el color del fondo de una página, puede llevar al programador a tener que pasar horas innecesarias modificando pequeños detalles de un aplicación muy grande.

La figura 1 describe la situación típica para la administración de una sola tabla en una base de datos: un sistema con 100 tablas tiende a tener 400 archivos ASP y, como se mencionaba anteriormente, si existe un cambio, por pequeño que este fuera, y si el programador no ha hecho un buen diseño de su aplicación, tendrá que modificar las 400 páginas una a una. Esto también sucede cuando se cambian las políticas de acceso sobre las diferentes tablas o registros por parte del cliente. Esto sin duda ocasiona grandes trastornos el desarrollo normal de un proyecto y tiene como resultado el incremento en los costos de desarrollo.

² Eliminar el registro seleccionado y los registros asociados a este mediante una llave foránea.

Figura 1. Diseño tradicional de Aplicaciones Administrativas de base de datos



Cuando se diseña un sistema de Información para que sea consultado vía Internet, se debe tener en cuenta la presentación de la información al usuario final. Para esto, se establecen plantillas³ que permitan crear una imagen homogénea del sistema. Cuando se tiene un amplio número de desarrolladores, el proceso de integrar los diferentes módulos suele ser tedioso. La ventaja de la herramienta que se presenta en este documento es que genera una sola plantilla de diseño, la cual es utilizada en todas y cada una de las páginas de forma homogénea, de tal manera que los cambios que se apliquen serán tomados automáticamente por todos los elementos del sitio con sólo modificar una plantilla.

Otro inconveniente que se tiene cuando se desarrollan sitios que tienen gran cantidad de tablas es el manejo de permisos; este manejo debe realizarse de manera directa sobre cada página, modificando los permisos de acceso sobre la misma directamente en el código. La herramienta que se plantea tiene un sistema administrativo que permite la creación de perfiles⁴ de usuario (y los usuarios pertenecientes a cada perfil con un identificador y una contraseña) que tendrán la posibilidad de otorgar o negar permisos de lectura y/o escritura sobre cada tabla en el sistema.

³ Son esquemas de presentación que permitirán la navegación de los desarrolladores de un sitio que use como base el sistema genérico de administración de base de datos.

⁴ Conjunto de características comunes (permisos de administración de la información) para varios usuarios de un sistema.

2.2.2 Descripción de objetivos. Como apoyo al desarrollo de sistemas de información se diseñó e implementó una herramienta para la administración de bases de datos creadas en el motor de base de datos SQLServer versiones 2000 y 7, permitiendo a los desarrolladores la ejecución de operaciones rápidas y eficientes sobre las diferentes tablas de la Base de datos. Adicionalmente, el sistema provee a los usuarios finales una interfase amigable y fácil de entender de los elementos almacenados en el servidor de aplicaciones.

Además de las tareas básicas de administración de registros (crear, listar, editar y eliminar), el sistema cuenta con elementos adicionales que facilitan a un equipo de desarrollo la configuración del sistema a sus necesidades, tales como el análisis de la información contenida en las tablas del catálogo del sistema de las bases de datos SqlServer 2000 (en éstas se encuentra información detallada, creada automáticamente por el motor, de cada una de las tablas y de sus campos). Por otra parte, cuenta con herramientas administrativas que permiten configurar el manejo de permisos y usuarios y de los elementos de modelado.

A partir de la experiencia del análisis de casos de desarrollos de sistemas de información se observó que el desarrollo por capas es una excelente opción para diseñar un sistema flexible y cambiante en el tiempo. Para la implementación de la capa de datos se desarrollaron consultas y procedimientos almacenados que permiten la comunicación eficiente con Microsoft SqlServer; gran parte de esta capa estará ubicada dentro del mismo motor de Base de datos de Microsoft SqlServer y será instalada en cada base de datos a administrar. Lo anterior se hace con el objetivo de maximizar el desempeño de la aplicación. En esta capa de Datos también se incluye el manejo de la información de meta modelado, la cual está contenida en la base de datos a administrar a manera de tablas adicionales, con formatos de nombres especiales ya que estas tablas deben estar ocultas para los usuarios finales).

Respecto a la capa de presentación, se desarrollaron plantillas y esquemas de funcionamiento para ser adaptadas a los diferentes sistemas a desarrollar de manera sencilla. Estas plantillas podrán tener diseños gráficos variados, de acuerdo a las

necesidades del usuario; sin embargo, guardan ciertos parámetros de configuración para el momento de elaborar el enlace con la aplicación Web.

2.3 IMPACTO Y VIABILIDAD

2.3.1 Impacto. Esta investigación se traduce en beneficio económico, porque la herramienta podrá ser utilizada por desarrolladores de aplicaciones al interior de la Universidad, lo cual beneficiará los intereses de la misma al reducir los tiempos y los costos de desarrollo orientado a Internet e intranet. Adicionalmente, esta aplicación tendrá un impacto tecnológico de acuerdo a la orientación que se le dé al diseño e implementación de los sistemas de información, ya que el paradigma de meta modelado esta tomando fuerza a nivel mundial debido al poder de cómputo actual y a la alta velocidad de los sistemas manejadores de bases de datos. Muchos de los sistemas de información de las grandes compañías están basados en esta técnica⁵, ya que se trata de negocios cambiantes y de misión crítica, razones por las cuales deben permitir realizar cambios de forma rápida y eficiente, sin comprometer la integridad de los datos.

Por otra parte, el desarrollo de una versión de producción comercial podría ser de amplia aceptación entre las empresas desarrolladoras de software, ya que permite disminuir los tiempos de desarrollo y, a su vez, su costo, lo que las haría más competitivas en el mercado. Adicionalmente, el desarrollo de la aplicación en ASP y la utilización de bases de datos Microsoft SqlServer 2000 o 7 es ventajoso, porque gran parte de los desarrolladores en Colombia están familiarizados con herramientas Microsoft para la aplicación de sistemas de este tipo.

2.3.2 Viabilidad. Como primera medida, el motor de SqlServer es de libre distribución⁶, lo cual resulta en una disminución en los costos de elaboración del proyecto por concepto de licenciamiento de software.

⁵ Ejemplo concreto de ello es la aplicación SAP utilizada por varias empresas importantes del país. Para mayor información consultar el sitio Web: www.sap.com/colombia/

⁶ Los administradores corporativos si tienen valor monetario.

En segunda instancia, el reto impuesto por la arquitectura inherente en la Web es superar las desventajas y limitaciones considerables, sobre todo en lo que se refiere a la dimensión y capacidad de las interfaces de aplicaciones, los medios y el contenido, y la sofisticación general de las soluciones que se pueden construir y entregar. Desde esta perspectiva, la herramienta provee a los diseñadores y administradores de bases de datos una herramienta útil e ideal para llevar a cabo la administración de sistemas de información vía Internet desde cualquier parte, es decir, independiente de la plataforma operativa.

3. MARCO DE TRABAJO

Los avances en la tecnología de la información guían a las empresas en las transformaciones requeridas para enfrentar un mundo competitivo, reformando el modo en el cual los servicios son ofrecidos, mejorando la eficiencia, efectividad y control de los servicios. Esta breve perspectiva del manejo de la información en la actualidad muestra una idea de las necesidades que deben cubrir las empresas desarrolladoras a la hora de emprender un proyecto, necesidades que requieren de soluciones rápidas y bajos costos frente a la competencia.

Con lo anterior se presenta un problema que abre las puertas a ofrecer una solución en cuanto al manejo de bases de datos, cuya propuesta se desglosará a continuación, y que se traduce en una herramienta de trabajo que permite a los desarrolladores de software obtener dos beneficios fundamentales a la hora de apoyar a las empresas en el manejo de la información:

- Permitir administrar la información contenida en las tablas de las Bases de Datos de las empresas, utilizando un sistema de Administración de Base de Datos Relacional elaborado en SQL Server.
- Permitir a las empresas desarrolladoras convertirse en líderes de proyectos en los cuales se deban usar las técnicas de desarrollo rápido de aplicaciones de Bases de Datos (RAD⁷).

3.1 DOMINIO DEL PROBLEMA

A continuación se describirán brevemente algunos conceptos y técnicas usadas para el desarrollo del sistema y la concepción de la idea general del mismo. Para ello se ha delimitado el contexto en donde la aplicación sería utilizada; además, se definieron marcos que engloban el dominio del problema:

⁷ Del inglés: Rapid Application Development.

3.1.1 Marco social. El marco social está definido por las personas o entidades que utilizarían la herramienta a desarrollar.

La experiencia personal frente al manejo de las bases de datos, el desarrollo orientado a plataformas con tecnología Web y el contacto directo con las necesidades de diferentes empresas, permitió identificar gran parte de los requisitos de la aplicación, cuyo marco social lo conforman los desarrolladores de páginas Web dinámicas orientadas a la administración de grandes cantidades de información y de modelos relacionales complejos, facilitándoles las tareas de crear, listar, editar y eliminar registros de las bases de datos de manera homogénea para toda una aplicación en conjunto y no de manera aislada.

3.1.2 Marco institucional. El marco institucional se define como el lugar o el espacio en el cual se realizan los procesos que se desean optimizar. En cuanto a este aspecto, se pueden definir dos marcos institucionales en los cuales se da la aplicabilidad de la herramienta:

- La universidad como ente educativo, el cual utilizará la herramienta para reducir los tiempos y costos de desarrollo Web al interior de la misma haciéndola aún más competitiva en el manejo de la información. Además, como aplicación orientada a la capacitación de temas referente a las estructuras de datos en el área de Ingeniería de Sistemas.
- Las empresas desarrolladoras de software orientado a Internet e intranet.

El sistema se ha desarrollado en la Escuela de Ingeniería de Sistemas en el marco de la Universidad Industrial de Santander, específicamente para el área de diseño y desarrollo de estructuras de datos.

3.1.3 Marco conceptual. El marco conceptual está conformado por aquellos temas relacionados con el desarrollo orientado a Internet, tales como: las plataformas informáticas basadas en redes (Internet), las tecnologías de desarrollo para estas plataformas y las capas de información en las bases de datos. Estos conceptos se describen más ampliamente a continuación.

3.1.3.1 Plataformas informáticas. Las plataformas informáticas son estructuras coordinadas de software y hardware que proveen la base para la construcción de los sistemas de información. Para evitar abordar demasiados temas alrededor de las plataformas informáticas únicamente se mencionarán los principales componentes de estas (las más utilizadas).

Los Servidores

Los equipos que actúan en la función de “servidor” pueden ser incluidos en dos grandes categorías: Servidores de Aplicaciones (y de Base de Datos) y Servidores de Archivos.

- Servidores de Aplicaciones (y de Base de Datos): almacenan la información de acuerdo a la función de la empresa y ejecutan las aplicaciones, en nivel regional o central.
- Servidores de Archivos (y de Impresión): actúan como centralizadores de archivos (de datos o de programas) y de recursos de impresión en una red local, al cual los demás micros de una red tienen acceso para obtener servicios.

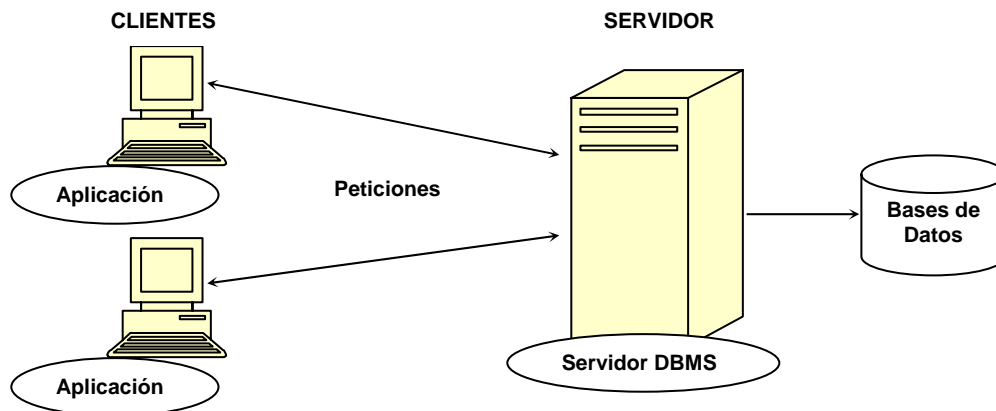
Un servidor Web, es un servidor de aplicaciones que se encarga de manejar la memoria dada a cada proceso, de proporcionar un método seguro para el intercambio interno de datos por los diferentes métodos de petición del cliente y de proporcionar un manejo de nombres y administración de los diferentes sitios Web creados y ubicados dentro del dominio del servidor.

La figura 2 muestra la secuencia en la que el cliente envía mensajes o solicitudes hacia el servidor de bases de datos. Los resultados de cada orden de son devueltos al cliente. El DBMS⁸ se encarga de recolectar los datos desde su base de datos, no envía los registros completos, teniéndose un uso mucho más eficiente de la capacidad de procesamiento distribuida. Es usual que se generen aplicaciones en el cliente y en el servidor. Los servidores de bases de datos constituyen un elemento fundamental en la arquitectura de aplicaciones Web y dadas características, permiten el desarrollo de aplicaciones de gran

⁸ El sistema manejador de bases de datos es la porción más importante del software de un sistema de base de datos. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

importancia para las empresas ya que permiten visualizar información de manera clara, lo cual apoya las decisiones corporativas.

Figura 2⁹. Secuencia de envío de información hacia el servidor de base de datos.



Las Estaciones de Trabajo (clientes)

Las estaciones de trabajo son microcomputadores conectados en red destinados, básicamente, a dos funciones: estaciones de desarrollo y estaciones de oficina.

Las estaciones de desarrollo son utilizadas por programadores y/o analistas para desarrollar los sistemas aplicativos de la organización, manejando lenguajes de programación, archivos de prueba, etc.

Las estaciones de oficina utilizan básicamente herramientas de productividad (procesadores de texto, hojas de cálculo, etc.), correo electrónico y los aplicativos desarrollados para la institución

3.1.3.2 Las aplicaciones Web. Una aplicación Web es el típico ejemplo de un aplicativo cliente servidor, en los cuales los clientes realizan peticiones al servidor mediante el uso

⁹ MÉNDEZ, D. Néstor Darío. Elementos a tener en cuenta en el diseño e implementación de bases de datos. Facultad de Ciencias y Administración. Universidad Nacional de Colombia. Manizales: 2003. Disponible en el sitio Web: <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060029>

de protocolos tales como el HTTP (Subconjunto del protocolo TCP/IP). Estas peticiones lanzan procesos en el servidor, el cual realiza cálculos u operaciones sobre bases de datos con el objeto de responder al cliente con un formato entendible y agradable.

Tabla 1¹⁰. Tecnologías de desarrollo Web del lado del servidor.

LENGUAJE	DESCRIPCIÓN
<i>Perl</i>	Practical Extraction and Report Language. Lenguaje interpretado creado con el objetivo principal de simplificar las tareas de administración de un sistema UNIX. Hoy en día se ha convertido en un lenguaje de propósito general.
<i>Python</i>	Es un lenguaje interpretado que permite escribir programas pequeños, utilizado en desarrollo Web para la creación de CGI
<i>C, C++</i>	Utilizado para la creación de CGI
<i>PHP</i>	PHP es un lenguaje interpretado diseñado para el desarrollo de sitios dinámicos. La distribución más popular de PHP es como módulo para el servidor Apache, aunque puede funcionar como un interprete para ejecutar aplicaciones CGI.
<i>ASP</i>	Active Server Pages. Tecnología creada por Microsoft destinada a la creación de sitios Web. Incluida dentro de los servidor Windows (Internet Information Server IIS). Esta tecnología ha sido ampliamente usada desde 1997
<i>ASP .Net</i>	Nuevo lenguaje de programación de Microsoft para la generación de aplicaciones Web, este nuevo lenguaje introduce todo un nuevo paradigma en programación.
<i>JSP</i>	Java Server Pages (JSP). Es un lenguaje interpretado insertado en páginas Web y basada en Java para el desarrollo de sitios dinámicos.

Las aplicaciones Web han tenido gran auge debido a su gran flexibilidad y la posibilidad de poder acceder a datos en cualquier parte del mundo, es por esto que es la elección ideal para empresas en un mundo cambiante y globalizado.

Tecnologías de desarrollo Web

Existen diferentes tecnologías para el desarrollo Web, cada una orientada a una tarea específica y con ventajas y desventajas propias.

¹⁰ MORA V. Andrea. Sistema intranet de información para el apoyo de la actividad académica en el programa de especialización en docencia universitaria del centro de desarrollo para la docencia universitaria - cededuis - sieduis 1.0 -. Propuesta de Grado. Colombia: Escuela de Ingeniería de Sistemas e Informática U.I.S, 2001.

- *Tecnologías del lado del servidor:* En la tabla 1 se presenta un resumen de las características más importantes de las tecnologías más utilizadas.

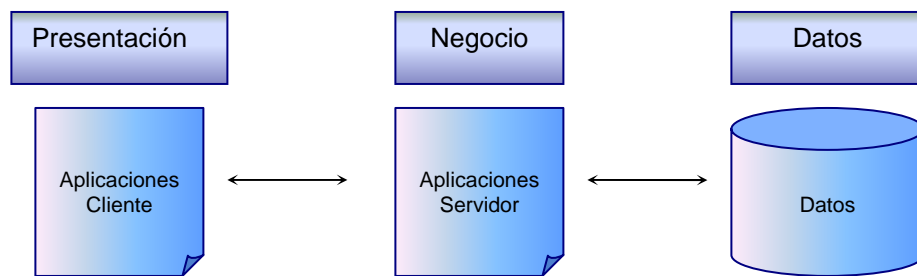
Tabla 2. Tecnologías de desarrollo Web del lado del cliente mundialmente reconocidas.

LENGUAJE	DESCRIPCIÓN
<i>HTML</i>	Hipertext Markup Language. Lenguaje que se utiliza para la creación de páginas Web. Consta de un conjunto de elementos, denominados etiquetas o marcas, que se utilizan para incluir texto, dibujos, tablas, cabeceras, etc.
<i>DHTML</i>	Extensiones de HTML que cubren el uso de hojas de estilo en cascada (CSS), uso de layers o capas para el posicionamiento y visibilidad de elementos en el navegador, programación con lenguajes de scripts y modelado de objetos de documentos (DOM).
<i>JavaScript</i>	Lenguaje interpretado incrustado en las páginas Web para permitir ejecutar código en el cliente a través del manejo de objetos y la captura de eventos producidos en el navegador.
<i>VBScript</i>	Lenguaje interpretado similar a JavaScript proporcionado por Microsoft.
<i>Applets de Java</i>	Pequeña aplicación accesible en un servidor Internet, que se transporta por la red, se instala automáticamente y se ejecuta como parte de un documento Web.
<i>FLASH</i>	Funciona como un plug-in del lado del cliente; sirve para presentar información dinámica y contenido animado al cliente lo cual enriquece la experiencia de la aplicación, haciendo más agradable la experiencia de por parte del usuario al incluir texto, sonido y animación.
<i>ActiveX</i>	Tecnología de Microsoft que permite ejecutar un programa que siga este estándar en los equipos de los usuarios al ser vinculado como objeto en una página Web. Los programas ActiveX dependen del sistema operativo Windows, por lo tanto no funcionan fuera de este ambiente.

- *Tecnologías del lado del cliente:* Del lado del servidor se desarrollan programas que ejecutan tareas especiales en el servidor y devuelven los resultados de dichos programas a los clientes en formatos especiales. La tabla 2 muestra un listado y las características fundamentales de los diferentes formatos para la presentación de información del lado del cliente.

3.1.3.3 Desarrollo por capas. Una arquitectura multicapa consiste en dividir la funcionalidad del sistema total en capas lógicas que pueden ser encapsuladas como componentes, entre los cuales existe una interacción a alto nivel, lo que supone una mayor escalabilidad de la aplicación, un mantenimiento menor y un incremento en la reutilización de componentes.

Figura 3. Arquitectura de Tres capas. Caso más general de arquitectura multicapas.



Típicamente, se utilizan tres capas en esta arquitectura: capa de datos, de negocio y de presentación. Así, en la capa de datos encapsulamos todas las funciones de base de datos; las funciones de interfaz de usuario en la capa de "Presentación" y, finalmente, toda la lógica de operación en la capa de "Negocios".

Esta encapsulación tiene como objeto ofrecer una colección más simple de funciones que permiten desarrollar aplicaciones a alta velocidad y bajo costo. El costo de mantenimiento del sistema disminuye notablemente, ya que una modificación en una capa no debe afectar a las demás.

Modelo de tres capas

(a) *Capa de Datos:* El nivel de servicios de datos es responsable de: Almacenar, recuperar y mantener los datos, así como verificar la integridad de los mismos.

Los servicios de datos tienen una variedad de formas y tamaños, incluyendo los sistemas de administración de bases de datos relacionales, servidores de correo electrónico y sistemas de archivos.

(b) Capa de Aplicación o de Negocios: La capa de aplicación es el “puente” entre un usuario, representado en la capa de presentación, y los servicios de datos, que se encuentran en la capa de datos. Los servicios de esta capa responden a peticiones del usuario (u otros servicios de negocios) para ejecutar una tarea de este tipo. Cumplen con esto aplicando procedimientos formales y reglas de negocio a los datos relevantes. Esto aísla al usuario de la interacción directa con la base de datos.

Esta capa contiene los algoritmos utilizados en el meta modelado y ejecuta las operaciones de análisis del modelo de datos, lo cual permitirá comunicar a la capa de presentación cómo realizar la presentación de la información de datos. En otras palabras, configura la capa de presentación para enviar datos al cliente final

(c) Capa de Presentación: Los servicios de presentación proporcionan la interfaz necesaria para presentar información y reunir datos. También aseguran los servicios de negocio necesarios para ofrecer las capacidades de transacciones requeridas e integrar al usuario con la aplicación, con el fin de ejecutar un proceso de negocios.

Los servicios de presentación generalmente son identificados con la interfaz de usuario y normalmente residen en un programa ejecutable localizado en la estación de trabajo del usuario final. El cliente proporciona el contexto de presentación, generalmente un navegador, como Microsoft Internet Explorer o Netscape, que permite ver los datos remotos a través de una capa de presentación HTML generada por la capa de Negocio.

Mediante el uso de capas se separa la programación que da acceso a los datos de las bases de datos de otros contenidos del sistema. Esto ayuda a asegurar que los desarrolladores estén libres para enfocarse en escribir su lógica de negocios en componentes sin preocuparse acerca de cómo se muestra la salida. Recíprocamente, esto da libertad a los diseñadores de usar herramientas familiares para modificar la interfaz.

3.2 DOMINIO DE LA SOLUCIÓN

El dominio de la solución se refiere al contexto en el que se desarrolla la herramienta; es decir, se mencionan aquellas tecnologías utilizadas que contribuyeron a la solución del problema planteado permitiendo un desarrollo rápido y óptimo de los procesos.

El dominio de la solución incluye los siguientes marcos:

3.2.1 Marco social. El marco social a nivel del dominio de la solución es el grupo de personas que participaron en el proyecto. Para la propuesta son: el diseñador y desarrollador de la aplicación y el asesor del proyecto quien, desde la normatividad de la Escuela de Ingeniería de Sistemas (U.I.S), se denomina el Director de la propuesta de grado.

Indirectamente, también participaron del proyecto las empresas de desarrollo que hicieron manifiesta la necesidad de la creación de la herramienta y que, a partir de la experiencia con relación a proyectos realizados, facilitaron la comprensión del dominio del problema y proporcionaron orientación en la búsqueda de la solución.

3.2.2 Marco conceptual. El marco conceptual, desde el punto de vista del dominio de la solución, reúne las características de todos los elementos que orientaron el desarrollo de la herramienta.

De acuerdo al tema del presente proyecto, se tuvieron en cuenta conceptos que conducen al método de modelado utilizado y las metodologías usadas para definir la especificación de requisitos, el diseño e implementación de la herramienta, que fue una combinación de Desarrollo Rápido de Aplicaciones con el Proceso Unificado de Desarrollo software (RUP¹¹).

3.2.2.1 Meta modelado. Para entrar a hablar de meta modelado, primero se tiene que definir qué es un meta dato. Meta dato, es un dato acerca de un dato. Considerando lo anterior tenemos que el meta modelado es información enriquecedora de un modelo, es

¹¹ Tomado del inglés: Rational Unified Process.

decir que para el caso, se enriquece un modelo entidad relación con información adicional para poder presentarlo de manera automática al usuario final.

Es de notar que el sistema sin meta información acerca del modelo de datos generaría los formularios, teniendo en cuenta los permisos de acceso dados por el sistema, pero para el usuario final no serían fácilmente asimilables, por cual se convierte en una parte esencial en la administración del sistema.

El Meta modelado es un concepto, no una metodología, el cual fue extraído desde su etimología y no de una bibliografía, pero como concepto aporta al diseño de la funcionalidad deseada en pro del usuario final, para el cual el manejo de la información, su consulta y manipulación deben realizarse de la manera más transparente posible.

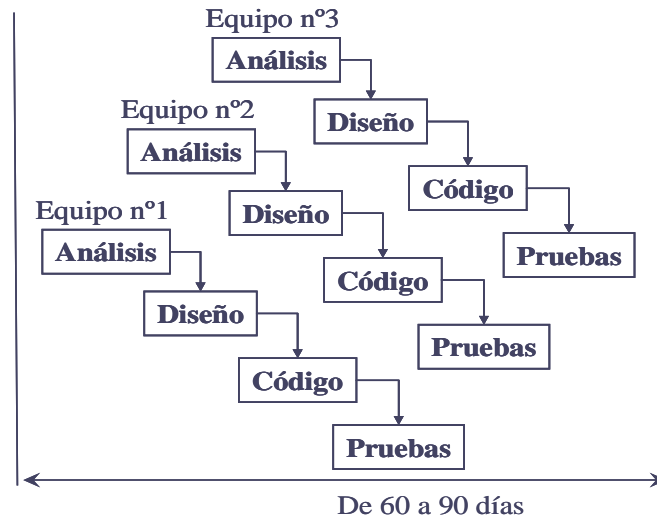
3.2.2.2 Modelo de tres capas. Dado que en un modelo de 3 capas la base de datos está separada de las aplicaciones, y que esta característica conlleva una mayor facilidad para hacer ajustes al software y al sistema, en términos de la herramienta se definieron las operaciones de meta modelado y de análisis de bajo nivel de los componentes de la base de datos (tablas, relaciones y datos). También se incluyó la programación en el motor de la base de datos para proporcionar mayor rendimiento de la aplicación.

3.2.2.3 Metodología de desarrollo. Para la realización de este proyecto se utilizó el Desarrollo Rápido de Aplicaciones basado en una segunda metodología como lo es el Proceso Unificado. Esta última se tomó como guía de desarrollo en la conducción de las actividades de administración y de ingeniería, más específicamente fue utilizado para capturar los requerimientos funcionales que influenciaron el diseño, la implementación y las pruebas de la herramienta; sin embargo, se han combinado algunas de sus fases y se han omitido ciertos flujos de trabajo precisamente por la velocidad de desarrollo que parte de la experiencia del autor de la propuesta.

A continuación se presenta una breve descripción de las metodologías de desarrollo software utilizado.

Desarrollo Rápido de Aplicaciones (DRA)

Figura 4¹². Modelo de Desarrollo de Aplicaciones Rápidas (DRA).



El modelo DRA es una adaptación a alta velocidad del modelo en cascada en el que se logra el desarrollo rápido utilizando un enfoque de construcción basado en componentes reutilizables y utilizando técnicas de cuarta generación en lugar de software con lenguajes de programación de tercera generación. En la siguiente figura se muestran las fases que comprende este modelo.

Es de anotar que la metodología DRA, apoya los desarrollos cuando se tiene experiencia en el desarrollo de las aplicaciones que se desea generar, pero no proporciona una metodología estándar para la representación de los diferentes elementos del sistema, por lo cual, para efectos de la comprensión y documentación se utilizan algunas características de proceso unificado.

¹² SANCHEZ, Pérez Javier. El Proceso de software: Metodología de desarrollo software. España: Facultad de Informática. Universidad de Las Palmas de Gran Canaria, 2002.

Tabla 3¹³. Descripción de las etapas de desarrollo de la metodología DRA.

ETAPAS	DESCRIPCIÓN	ACTIVIDADES	RESULTADOS
Gestión de Modelado	El flujo de información entre las funciones de gestión se modela de forma que responda interrogantes alrededor de la gestión de la información	Estudio de las fuentes, flujos y destino de la Información	Definición del flujo de información
Modelado de Datos	El flujo de información se refina como un conjunto de objetos de datos necesarios para apoyar la empresa. Se definen los atributos de los objetos y las relaciones entre objetos	Definición de los atributos de los objetos y las relaciones entre ellos	Atributos de los objetos y relaciones entre ellos
Modelado del Proceso	Se logra el flujo de información necesario para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir o recuperar un objeto de una base de datos		
Generación de Aplicaciones	En esta fase se trabaja para utilizar componentes de programas ya existentes o para crear componentes reutilizables	Generar Código	Código
Pruebas y Entrega	En esta etapa se prueban todos los componentes nuevos y se deben ejercitar todas las interfases a fondo		Listado de Errores y cambios sugeridos

En proyectos donde se comprenden bien los requisitos, se limita correctamente el ámbito del proyecto, y el sistema se puede dividir en módulos, el proceso DRA permite al equipo de desarrollo crear un sistema completamente funcional dentro de periodos cortos de tiempo (60-90 días).

La figura 4 muestra el modelo de desarrollo rápido de aplicaciones mediante módulos de procesos que resumen la topología de la metodología.

¹³ GOMEZ, F. Luis Carlos. Ciclos de vida de desarrollo software. Grupo de Investigación en sistemas y tecnologías de la información – STI - . Universidad Industrial de Santander. Colombia: 1993.

En la tabla 3 se describen cada una de las etapas que se deben seguir en un modelo de desarrollo DRA, así como las actividades y los resultados esperados en cada una de ellas.

El Proceso Unificado

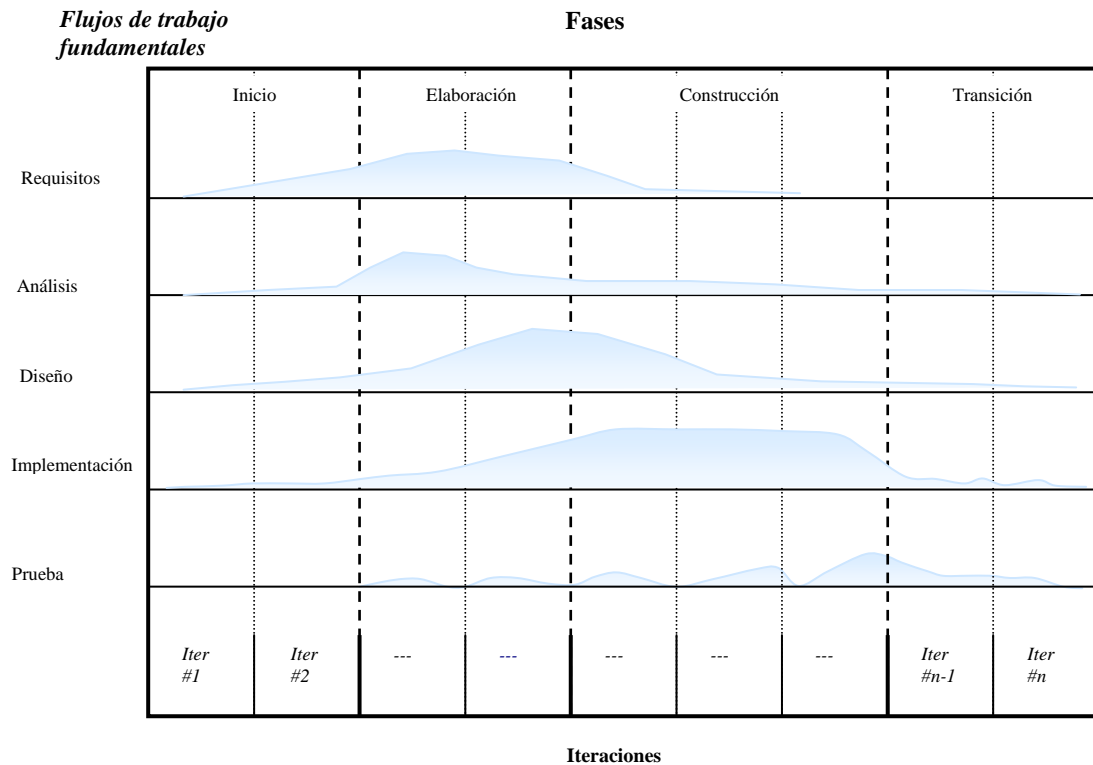
El proceso unificado está basado en componentes y se sostiene sobre tres ideas básicas: casos de uso, arquitectura y desarrollo iterativo e incremental. Para hacer que estas ideas funcionen, se necesita un proceso, que tenga en cuenta ciclos, fases, flujos de trabajo, gestión del riesgo, control de calidad, gestión de proyecto y control de la configuración.

Un desarrollo iterativo, guiado por los casos de uso y centrado en la arquitectura, construye un software mediante pequeños incrementos; añade cada incremento a la acumulación previa de incrementos, de tal forma que siempre se tenga una construcción ejecutable. La arquitectura proporciona la estructura sobre la cual guiar las iteraciones, mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración.

De esta manera el proceso reduce el riesgo de grandes retrasos en la entrega de un producto, se fijan metas más inmediatas, por lo cual se puede controlar mejor el avance del proyecto.

El Proceso Unificado divide el proceso de desarrollo en ciclos (figura 5), donde se obtiene una nueva versión del producto al final de cada ciclo. Cada ciclo se divide en cuatro Fases: Inicio, Elaboración, Construcción, y Transición. Cada una de estas fases concluye con un hito bien definido donde deben tomarse decisiones respecto al proyecto como la reestructuración del cronograma de trabajo. Cada una de estas fases se divide a su vez en iteraciones.

Figura 5¹⁴. Flujos de trabajo que tienen lugar en las cuatro fases del Proceso Unificado y disciplinas de soporte asociadas.



Cada iteración sigue la estructura de un pequeño ciclo de vida en cascada, pasando a través de los cinco flujos de trabajo fundamentales: requisitos, análisis, diseño, implementación y prueba. En la iteración también incluye la planificación que precede a los flujos de trabajo y la evaluación que va detrás de ellos.

3.2.3 Marco tecnológico. El marco tecnológico está constituido por las tecnologías que pasaron a formar parte del sistema y las herramientas que se utilizaron para desarrollar la herramienta.

¹⁴ JACOBSON, Ivar; RUMBAUGH, James y BOOCH Grady. El Proceso Unificado de Desarrollo de Software. Rational Software Corporation. España: Editorial Addison Wesley autorizado por Pearson Educación S.A, 1999.

Se describen a continuación el conjunto de tecnologías de desarrollo Web utilizadas, de acuerdo a la clasificación presentada en el numeral 3.1.2, además se realiza una breve descripción del manejador de bases de datos utilizado.

3.2.3.1 Tecnologías de desarrollo. La tecnología del lado del servidor seleccionada para el desarrollo de la herramienta fue ASP, dado el conocimiento previo del autor de la presente propuesta. Los principales motivos de su selección fueron: su estabilidad, facilidad de manejo y depuración y, sobre todo, la amplia documentación y ejemplos encontrados.

En un sentido más amplio, una aplicación ASP es un ambiente del lado del servidor para crear Scripts para desarrollar aplicaciones dinámicas e interactivas. Con ASP se pueden combinar páginas HTML, comandos de Script del lado del cliente, y componentes COM¹⁵, lo cual posibilita al desarrollador crear potentes aplicaciones fáciles de implementar y mantener.

Con ASP se manipula directamente el código HTML y se genera sobre el explorador cliente; este aspecto proporciona un grado de flexibilidad más alto en el diseño y creación de páginas dinámicas, pero tiene un precio en cuanto al desempeño general de la aplicación; dicho inconveniente se suple con el diseño basado en tres capas de datos.

Las páginas ASP, pueden ser programadas en JavaScript¹⁶ o en VisualBasic Script¹⁷, siendo este último el de mayor utilización a nivel mundial. La tecnología utilizada para la presentación de la información del lado del cliente para efectos del presente proyecto fueron: HTML, DHTML¹⁸ y Javascript.

3.2.3.2 Manejador de Bases de datos (DBMS). El manejador de datos SQLServer 2000 de la compañía Microsoft, permite la creación y la administración de bases de datos

¹⁵ Tecnología de Componentes del Modelo de Objetos de Microsoft. Información disponible en el sitio Web: http://searchwin2000.techtarget.com/sDefinition/0,,sid1_gci211823,00.html

¹⁶ Lenguaje de desarrollo Web de Sun Microsystems. Información disponible en el sitio Web: www.sun.com

¹⁷ Lenguaje de desarrollo Web de Microsoft Corporation. Información disponible en el sitio Web: www.microsoft.com

¹⁸ Mayor información disponible en el sitio Web: www.w3c.org

relacionales, maneja recursos de disco, memoria y almacenamiento. Además, permite la interacción con el usuario mediante el uso de un administrador corporativo que proporciona una utilidad para administrar las tablas y los diferentes elementos de las bases de datos relacionales, así como las características propias del SQLServer, como son los procedimientos almacenados, los usuarios, entre otras.

También, y aun más importante, permite la ejecución de comandos SQL estándar permitiendo así la creación de consultas y la modificación de la estructura del modelo de la base de datos

El SQLServer cuenta con consultas estándar para traer la información del catálogo del sistema, conocidas con el prefijo **INFORMATION_SCHEMA**¹⁹; estas consultas extraen la información del catálogo de la base de datos. Con estas consultas se obtiene la información del modelo de cada una de las tablas, columnas, relaciones, condiciones, validaciones, información de llaves (primarias y foráneas), las cuales se combinan con las tablas de metamodelado para la generación de formularios y listados y, en conjunto con el sistema de permisos, el manejo del sistema en general.

Una de las características que se tuvo en cuenta para seleccionar el motor de base de datos de SQLServer es que es de libre distribución²⁰, ello hace posible la instalación de la aplicación en los servidores destino sin tener preocupaciones de licencia. Asimismo se recomienda el uso de administradores corporativos gratuitos, como el **Aqua Data Studio**²¹, el cual administra bases de datos de diferentes motores y es de libre distribución

¹⁹ Tomado de la ayuda en línea de SQLServer.

²⁰ Se entiende la licencia en virtud de la cual cualquier persona está facultada para hacer copias de un programa de computador, distribuirlo, explotarlo económicamente y modificarlo.

²¹ Mayor información en el sitio Web: <http://www.aquafold.com/>

4. DESARROLLO

4.1 INTRODUCCIÓN

Para el desarrollo de este proyecto se empleará el modelo de ciclo de vida DRA (Desarrollo Rápido de Aplicaciones) que fue elegido teniendo en cuenta los siguientes elementos:

- Los requerimientos del sistema son simples y no cambiarán a través del desarrollo del proyecto.
- La tecnología ASP y las bases de datos SqlServer han sido previamente estudiadas para iniciar el desarrollo de este proyecto.
- Ya se han realizado estudios previos de la tecnología a utilizar y se tiene una arquitectura basada en cliente-servidor en tres capas.
- La planificación del desarrollo está basado en reutilización de código con lo cual se disminuyen las estimaciones de tiempo que normalmente tomaría la implementación de la herramienta.

Cabe anotar que en el proyecto se utilizará la metodología Desarrollo Rápido de Aplicaciones de desarrollo de software (modelo tradicional²²) pero con algunas modificaciones ya que se emplearán algunas herramientas de otro ciclo de vida (Proceso Unificado – modelo para sistemas orientados a objetos²³) para complementar el desarrollo de este proyecto y hacer más fácil su comprensión y su desarrollo.

Las etapas de modelado DRA consideradas son cinco: la gestión de modelado, el modelado de datos, el modelado del proceso, la generación de aplicaciones y se concluirá con la etapa de pruebas y entrega.

²² Modelo formados por un conjunto de fases o actividades en las que no tienen en cuenta la naturaleza evolutiva del software.

²³ Modelos con un alto grado de iteratividad y solapamiento entre las fases que los componen.

4.2 GESTIÓN DE MODELADO

El objetivo principal de la etapa de gestión de modelado es revisar el flujo de información entre las funciones de gestión que se modela, de manera que se responda a los interrogantes alrededor de la gestión de la información.

El objetivo a cumplir en esta fase es revisar las fuentes, flujos y destino de la información, con el fin de responder a preguntas esenciales como:

- ¿Cuáles son las principales funciones del sistema para sus usuarios más importantes?
- ¿Cómo podría ser la arquitectura del sistema?
- ¿Cuál es el plan de proyecto y cuánto costará desarrollar el producto?

En esta etapa se realizará la investigación pertinente que permita establecer los elementos necesarios para desarrollar exitosamente el proyecto; se consultarán las referencias bibliográficas y se reunirá toda la información posible en cuanto al uso de las tablas del catálogo del sistema.

Al combinar el ciclo de vida DRA con las técnicas de modelado del PUR se podría obtener la respuesta a la primera pregunta, realizando un modelo de caso de uso simplificado que contenga los casos de uso más críticos. Cuando ya estén listos los casos de uso, la arquitectura es provisional y únicamente se obtiene un bosquejo de ella. En esta fase se identifican y priorizan los riesgos más importantes, se planifica en detalle la fase de elaboración y se estima el proyecto de manera aproximada. Además, se definen los elementos críticos sobre los cuales operará todo el sistema.

4.2.1 Especificación de Requisitos. La especificación de requisitos software está relacionada con la funcionalidad, el desarrollo, las restricciones del diseño, los atributos y las interfaces externas. “En particular debe reconocerse cualquier requisito externo impuesto por una especificación del sistema y debe tratarse. Además deben definirse las

respuestas del software a todos los posibles datos de la entrada del sistema y a toda clase de situaciones”²⁴.

Una nota que es importante especificar son las respuestas a las entradas válidas e inválidas a ciertos valores que puedan poner en riesgo la fiabilidad del sistema.

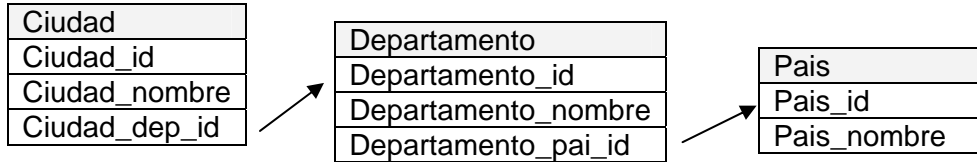
4.2.1.1 Definiciones, siglas y abreviaciones. La definición de los términos que se utilizaron en la propuesta permite ampliar la comprensión del tema tratado.

- **ACTORES:** Un actor abstrae y caracteriza a uno o varios usuarios externos que interactúan con el sistema. El objeto usuario puede ser una persona, un sistema informático, otro subsistema u otro tipo de objeto. El conjunto completo de actores describe todas las diferentes formas de comunicación entre los usuarios externos y el sistema.
- **ANEXOS:** Son archivos asociados a cada registro, los cuales extienden la información contenida en el mismo, se pueden tratar de imágenes, documentos, etc.
- **BÚSQUEDA EN PROFUNDIDAD:** Es la característica del sistema que permite la presentación de la información al usuario final de manera filtrada para la generación de formularios. Si se tiene el caso en que una tabla tenga una relación a una ciudad, sin configurar ningún nivel de profundidad, mostraría todas las ciudades registradas en la tabla. La intención de la búsqueda es proporcionar un sistema de selección más sencillo.

La figura 6 muestra el caso presentado solucionado a través de la búsqueda de profundidad, en donde la ciudad es la que esta directamente relacionada con el campo de la tabla en la que actualmente se está trabajando; adicionalmente, se muestra la información del departamento, lo cual proporciona un filtro que hace de la selección un proceso más sencillo.

²⁴ ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE. Estándar IEEE: 830 de 1993. Disponible en el sitio Web: <http://www.ewh.ieee.org/r9/guadalajara/boletin/sep01/>

Figura 6. Representación de la característica de la búsqueda en profundidad para la tabla Ciudad.

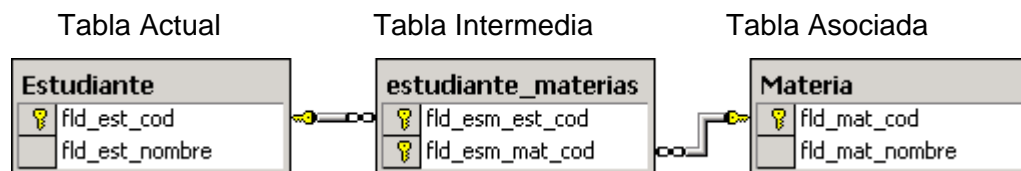


Si se está alimentando la tabla Ciudad, entonces para seleccionar el departamento a la que pertenece, se muestra un combo adicional que permite seleccionar el país al cual pertenece ese departamento.

- **CASOS DE USO:** Conjunto de roles que los usuarios de un sistema asumen cuando interactúan con el mismo. Es la especificación de las secuencias de acciones, incluyendo secuencias variantes y secuencias de error, que pueden ser efectuadas por el sistema, subsistema o clase por interacción con autores externos.
- **DBMS: DataBase Management System,** Sistema de Gestión de Bases de Datos es un software específico que gestiona y organiza bases de datos. Un DBMS es sustancialmente un software que se coloca entre el usuario y los datos como tales.
- **DIAGRAMA DE COLABORACIÓN.** Es aquel que enfatiza la organización estructural de los objetos que envían y reciben mensajes; muestra las interacciones organizadas alrededor de instancias y de los enlaces entre ellas. Por otra parte, muestra cómo el control pasa de un objeto a otro a medida que se lleva a cabo el caso de uso, de igual manera, envía mensajes a través de los objetos.
- **ENLACES WEB:** Es la información que se le anexa o relaciona a cada registro de cada tabla de manera que complementen la información prevista para los registros definidos en la base de datos. Estos enlaces pueden ser de tipo http, ftp o mail.

- **INFORMACIÓN ASOCIADA:** Es la información que se extrae de las relaciones muchos a muchos de las tablas. Por ejemplo: si actualmente se observa una tabla denominada “Estudiante” y otra llamada “Materias”, se realiza el siguiente análisis: un estudiante puede cursar muchas materias en un determinado semestre; por tanto, esta información se almacenaría en una tabla intermedia creada por el diseñador del sistema (figura 7). El sistema de modelado genérico provee un método semiautomático para manejar este tipo de relaciones. Se presenta una página al administrador en donde se muestra la página actual, la cual es intermedia. Se debe seleccionar el tipo de tabla para que el sistema pueda determinar si se trata de una tabla intermedia o de una tabla en una relación en profundidad (por ejemplo: Ciudad, departamento, País). Si se hiciera de manera automática, el sistema asumiría a Departamento como tabla intermedia lo cual no siempre es deseado.
- **INFORMACIÓN RELACIONADA:** Es la información que se trae con relación a la tabla que actualmente se está trabajando. El sistema maneja automáticamente aquellas relaciones uno a muchos creadas dentro del diseño de la base de datos del usuario. (Ver Figura 7)

Figura 7. Representación de una relación muchos a muchos para un modelo relacional de bases de datos.



En este caso, (figura 8) la tabla personas tiene una tabla relacionada a ella de modo uno a muchos, lo cual genera un elemento que permite seleccionar un dato de la tabla relacionada (ciudad).

Figura 8. Representación de la relación entre las tablas de modo uno a muchos para un modelo relacional de bases de datos.

Personas	
Nombre	
<...>	
Ciudad	
<input type="text" value=" "/>	
Bucaramanga	

- **JAVASCRIPT:** Lenguaje de secuencia de comandos para el Web que es usado tanto en el navegador como en el servidor Web desarrollado por Netscape Communicators. Como todos los lenguajes de script, es usado principalmente para unir componentes o para aceptar entrada de datos del usuario.
- **METADATOS:** Un Metadato es la información que describe el contenido, calidad, condición y otras características de un dato.
- **ODBC:** Open Database Connectivity, es un programa de interfase de aplicaciones (API) para acceder a datos en sistemas manejadores de bases de datos tanto relacionales como no relacionales, utilizando para ello SQL (**Structured Query Language**- Lenguaje de consulta estructurado).
- **PERMISO PROVEEDOR:** Permite a los usuarios del sistema la adición de nuevos registros a las tablas de la base de datos definida por el desarrollador, únicamente para aquellos que él mismo haya creado. Este permiso también incluye un acceso de lectura de la información.
- **TABLA BASE:** Son las tablas a partir de las cuales se almacena información en las tablas de metamodelado. En estas tablas se almacena información específica que provee el usuario con relación al sistema, la cual será modificada a través de las herramientas provistas por el sistema de meta modelado.

- **TABLAS DE METAMODELADO:** Las tablas de meta modelado almacenan información que enriquecen el modelo entidad relación que se define para soportar el sistema de información en general. Estas tablas tienen el prefijo `ult_`, lo cual permite distinguirlas del resto de las tablas del sistema. Este prefijo se utiliza para todos los elementos relacionados con el sistema de modelado: Tablas, vistas, procedimientos almacenados, etc.
- **TABLAS DEL SISTEMA:** Son tablas provistas por SQLServer para almacenar información del modelo de datos creado por parte del usuario final²⁵.
- **TABLAS DE USUARIO:** Son tablas creadas por un usuario desarrollador de SQL Server cuando está creando un modelo de datos propio para su uso; sobre estas tablas opera el sistema de modelado y estas son la base para presentar la información a los usuarios de la aplicación Web que utilice como base el sistema desarrollado en este proceso.
- **TÉCNICA DESCONECTADA:** Al realizar pruebas de desempeño generales del sistema, se determinó que algunas consultas sobre las tablas del catálogo del sistema suelen ser lentas, por lo cual el resultado de las consultas se graba en una tabla de usuario para combinarlo luego con las tablas de meta modelado. Esto aumenta considerablemente el desempeño en el momento de la generación de formularios y reportes.
- **VISTAS DE ANÁLISIS DEL CATÁLOGO DEL SISTEMA:** Estas vistas permiten conocer y generar los elementos contenidos dentro de la base de datos; además, traen los esquemas de información de la base de datos a través de vistas estándar del sistema, sin tener que modificar las tablas del sistema en SQLServer. Estas consultas se utilizan de esta manera para que sean funcionales en futuras versiones del SQLServer.

²⁵ Se puede consultar un listado completo de las tablas del sistema en los libros en línea de SQL Server, buscando por la palabra clave “system tables, all databases”

- **VISTAS DE METAMODELADO:** Estas vistas combinan la información de meta modelado con las vistas de análisis del catálogo del sistema. De estas vistas se extrae la información final para la generación de formularios, reportes y administración del sistema.

4.2.1.2. Requisitos preliminares. A continuación se presentan las características preliminares que debe de tener el sistema., las cuales serán desarrolladas mas adelante en el documento y durante la implementación del prototipo, estas características serán los lineamientos que seguirán durante todo el proceso.

El sistema tendrá la capacidad de:

- Reconocer, validar y generar los elementos contenidos dentro del modelo de base de datos relacional entregada por el usuario, a través de vistas de análisis del catálogo del sistema o consultas que no requieran la alteración de las tablas diseñadas.
- Permitir la integración de la herramienta con la base de datos creada por el usuario en SQL Server 2000 o 7 2 mediante la ejecución del script de instalación del sistema.
- Proveer un método de optimización semiautomático en la extracción de las relaciones muchos a muchos de las tablas que componen la base de datos del sistema a través del uso de la técnica desconectada.
- Almacenar información adicional que enriquezca el modelo entidad relación definido por el usuario, mediante el uso de tablas de metamodelado que soportan el sistema de información en general.
- Permitir la búsqueda de parámetros y expresiones dentro de las tablas de la base de datos previamente reconocida por la herramienta, basados en algoritmos de análisis de consultas parametrizadas, siempre y cuando se utilice la versión de SqlServer 2000.
- Generar formularios y reportes de la información almacenada en la base de datos mediante la combinación de las consultas de las vistas de meta modelado y las vistas de análisis del catálogo del sistema.
- Permitir la presentación de la información al usuario final de manera filtrada mediante la configuración de los enlaces entre la página de reportes y la aplicación Web, que proporcionan un sistema de selección de combos de datos predefinidos.

- Mantener automáticamente las fechas de actualización de cada registro para cada tabla, con el fin de informar a los visitantes las últimas modificaciones realizadas, y para los administradores del sistema se constituye en una herramienta de auditoría.

4.2.1.3. Modelo de Casos de uso. Para la descripción del modelo de casos de uso se hace necesario describir las entidades (actores) que interactúan con el sistema y que fueron identificadas en la primera etapa del DRA.

Actores. Los posibles usuarios encontrados en el análisis realizado son: Público, Cliente y Administrador. De manera más amplia se describen las características de cada uno de estos actores a continuación:

Público: El *Público* representa a las personas que utilizan el sistema para la visualización de la información contenida en una página. Este actor es aquella persona que navega por la Web y que accede al sistema (conocimiento previo de la dirección Web o a través de motores de búsqueda) y que solo ejerce actividades de observación de la información.

Cliente: El actor *Cliente* es aquel que posee un nombre de usuario y una contraseña para acceder a la página Web. El *Cliente* utiliza el sistema para actualizar la información de la base de datos; sin embargo, esta actividad no se realiza de manera inmediata, cualquier acción de acceso a datos será validada por el administrador antes de ser publicada en el sitio Web.

Administrador: Este actor representa a un profesional con conocimientos en desarrollo Web que tiene como principal función diseñar la base de datos y ejecutar la interacción de la herramienta de administración con la página Web. Tiene la capacidad de definir perfiles de usuario y otorgar permisos de acceso; revisar, publicar o rechazar la información proveniente del actor *Cliente* y, en general, manejar los datos del sistema.

Casos de uso. Los casos de uso son los requisitos funcionales del sistema orientado al uso que harán del mismo los actores. A continuación se describen los casos de uso más significativos para los actores que se definieron anteriormente y que se consideran los

más importantes en el desarrollo del Sistema Genérico de Administración de Bases de Datos.

Es importante anotar que la identificación y descripción de los casos de uso no es una actividad que se realiza de una vez, sino que a lo largo de las etapas del proceso los casos de uso van refinándose y añadiéndole detalles. Además pueden ir surgiendo nuevos casos de uso no detectados en un principio o eliminándose de acuerdo a la funcionalidad del sistema.

Caso de uso: Autenticar Usuario

Todos los actores utilizan este caso de uso para que el sistema verifique la entrada de los usuarios al sistema, asignando el perfil de seguridad necesario para manipular la información de la página.

Precondición

Abrir la aplicación Web.

Flujo de sucesos

- Camino básico
 1. El actor especifica su nombre de usuario y contraseña.
 2. El actor selecciona de un combo de opciones su perfil de usuario.
 3. El sistema despliega la página con la aprobación ó negación de acceso respectivo.
 4. La instancia de caso de uso termina.
- Caminos alternativos
 - El actor puede cancelar el proceso de autenticación de usuario en los pasos 1 y 2.
 - El actor puede recibir el mensaje de introducir nuevamente el nombre de usuario y/o la contraseña si estas no corresponden a las almacenadas en el sistema.

Poscondición

Termina la instancia del caso de uso con la presentación de las plantillas que les permitirá a los diferentes actores acceder a la información para la cual se han asignado los permisos correspondientes.

Para el caso del usuario *Administrador* el sistema le permitirá manejar la información de los usuarios, permisos y modelado.

Requisitos especiales

El sistema debe tener configurados los perfiles de usuario. Esta actividad es realizada por el actor *Administrador*.

Caso de uso: Realizar inscripción

El *Público* y *Cliente* utiliza este caso de uso para realizar la inscripción a la página Web ingresando sus datos personales para ser reconocido y poseer una identificación de usuario y una contraseña válidos en el sistema abrir la aplicación que permitirá realizar la simulación.

Precondición

Abrir la aplicación Web.

Flujo de sucesos

- Camino básico
 1. El actor ingresar al sistema de inscripción pulsando la opción de registrarse.
 2. Diligencia el formulario de inscripción al sistema, donde se captura la información de contacto con el usuario.
 3. Ingresa un login (nombre de usuario) y una contraseña.
 4. Da clic sobre el botón Aceptar y concluye la instancia de caso de uso.
- Caminos alternativos

En el paso 2 se puede cancelar el proceso de inscripción.

Poscondición

Se han establecido las características del actor necesarias para acceder a la información de la base de datos.

Caso de uso: Realizar Integración

El *Administrador* utiliza este caso de uso para realizar la integración de la herramienta con la base de datos diseñada por el desarrollador de la aplicación web.

Precondición

El *Administrador* debe haber realizado el modelo entidad relación de la base de datos en SQL Server 2000 ó 7.

Flujo de sucesos

1. El actor ejecuta el script de instalación del sistema.

2. El actor copia los archivos de la aplicación a la carpeta destinada para el sitio Web que se desea crear.
3. El actor crea un directorio virtual en el Internet Information Server.
4. El actor configura los archivos del sistema.
5. La instancia del caso de uso termina.

Poscondición

El administrador podrá continuar con el caso de uso *Realizar Metamodelado* o *Definir Permisos*.

Requisitos Especiales

El actor debe conocer los pasos necesarios para la creación del directorio para Windows 2000.

Caso de uso: Definir Permisos

El *Administrador* utiliza este caso de uso para otorgar permisos de acceso a la información a todos los actores que interactúan con el sistema con el fin de protegerla de modificaciones no deseadas.

Precondición

El sistema debe haber autorizado el acceso del actor *Administrador* mediante el CU *Autenticar Usuario*.

Flujo de sucesos

- Camino básico
 1. El *Administrador* ingresa a la sección de otorgar permisos.
 2. Selecciona de una lista el tipo de usuario al cual le otorgará los permisos.
 3. Selecciona de una lista el nombre de la tabla sobre la cual se le otorgarán los permisos.
 4. Elige los campos de la tabla y los tipos de permiso (lectura, escritura, proveedor).
 5. Pulsa el botón *Actualizar Permisos*.
 6. La instancia de caso de uso termina

Poscondición

Los usuarios del sistema podrán realizar las inscripciones a la página y acceder a la información de acuerdo a los permisos otorgados sobre las tablas de la base de datos.

Caso de uso: Realizar Metamodelado

Este caso de uso es utilizado por el *Administrador* con el fin de realizar la configuración de las tablas de la base de datos y así, desplegar la información para los demás actores del sistema.

Precondición

El Administrador debe haber ejecutado los casos de uso *Realizar Integración y Autenticar Usuario*.

Flujo de sucesos

- Camino básico
 1. El *Administrador* ingresa al módulo de Administración del Modelo.
 2. El sistema muestra un listado de las tablas identificadas de la base de datos
 3. El actor selecciona una de las tablas listadas.
 4. El sistema despliega los campos de la tabla.
 5. El actor asigna un nombre amigable tanto a la tabla como a cada uno de los campos.
 6. El actor asigna un tipo de dato (Fecha, Texto, **Byte**) a cada campo para su posterior despliegue en los formularios.
 7. El actor selecciona los campos que van a ser mostrados a los usuarios *Público* y *Cliente*.
 8. El actor oprime el botón "Actualizar"
- Caminos alternativos

El *Administrador* puede cancelar el proceso de metamodelado en el paso 3.

Poscondición

Los usuarios del sistema podrán ver la información contenida en la base de datos y actualizarla a través de la aplicación Web.

Variación

Si el actor omite el paso 5 el sistema desplegará el contenido de los registros precedidos con los identificadores que el diseñador del modelo relacional ha asignado tanto a las tablas como a los campos.

Caso de uso: Actualizar información

Los actores *Cliente* y *Administrador* utilizan este caso para modificar la información contenida en la página referente a los productos y/o servicios que esta ofrece.

Precondición

El sistema debe haber autorizado el acceso de los actores mediante el CU *Autenticar Usuario*

Flujo de sucesos

- Camino básico
 1. El actor se ubica en la página de reporte en la cual se encuentra la información a modificar.
 2. Pulsa el botón “Ver”.
 3. En el formulario que se presenta debe escribir el nuevo contenido.
 4. Pulsa el botón Actualizar para que los cambios se hagan efectivos.
 5. La instancia de caso de uso termina
- Caminos alternativos

Los actores pueden cancelar el proceso de actualización en los pasos 2 y 3.

Poscondición

Termina la instancia del caso de uso con la presentación de un mensaje de aprobación de la actualización.

Requisitos especiales

Toda información que intente actualizar el usuario *Cliente* no tiene efecto inmediato en la base de datos del sistema; la publicación de esta en la red es autorizada por el *Administrador*.

Caso de uso: Autorizar Actualización

El *Administrador* utiliza este caso de uso para autorizar la actualización de la información que ha sido modificada por el actor *Cliente*.

Precondición

El sistema debe haber autorizado el acceso al sistema al actor mediante el CU *Autenticar Usuario*.

Flujo de sucesos

- Camino básico
 1. El sistema despliega un mensaje en el que le indica al actor que la información de una tabla ha sido actualizada o eliminada por el actor *Cliente*.

2. El actor selecciona el botón “Ver Propiedades de la Tabla”.
 3. El sistema destacará la información de los campos que ha sido modificada.
 4. El actor activa aquellas acciones de modificación que aprueba y oprime el botón Actualizar.
 5. El sistema le muestra un mensaje de confirmación de la acción.
 6. El actor da clic en el botón “Aceptar”.
- Caminos alternativos

El *Administrador* puede cancelar el proceso de actualización en el paso 6.

Poscondición

Los datos serán actualizados en la base de datos.

Caso de uso: Consultar información

El *Público* y *Cliente* utilizan este caso de uso para visualizar la información correspondiente a los productos y/o servicios ofrecidos en la página.

Precondición

Abrir la aplicación Web.

Flujo de sucesos

- Camino básico
 1. El actor recorre los accesos directos dispuestos en la página.
 2. Selecciona el ítem sobre el cual desea desplegar la información.
 3. La instancia de caso de uso termina
- Caminos alternativos

Los actores pueden cerrar la aplicación en los pasos 1 y 2.

Poscondición

La instancia de caso de uso termina cuando se han consultado la información.

La figura 9 muestra el diagrama de casos de uso completo de la etapa de Gestión de Modelado. Este modelo de casos de uso es un bosquejo preliminar de la recolección de los requisitos funcionales de la herramienta a desarrollar, sin embargo, engloban las actividades principales a realizar por los actores. Las acciones directas sobre la base de

datos son únicamente llevadas a cabo por el actor Administrador quién es la persona con los permisos para crear, modificar y eliminar.

En las siguientes etapas DRA se obtendrá una visión más amplia de los casos de uso y de los actores que interactúan con el sistema.

4.2.1.4 Requisitos Adicionales. Los requisitos adicionales engloban las demás categorías de clasificación de requerimientos software (además de los funcionales expuestos en el numeral 4.2.1.3) y son los requisitos de datos y los no funcionales.

Requisitos de datos (Anexo A)

Para los datos de entrada al sistema deberán considerarse los siguientes requerimientos:

- Los datos entregados por el usuario respecto al diseño de la base de datos deberán adaptarse al modelo de datos del sistema y deben tener los formatos de entrada apropiados, aunque en la mayoría de casos el sistema proporciona una validación automática de esos campos. Los cuales están representados en la figura 9
- Es responsabilidad del usuario final ingresar correctamente la información del enlace entre la aplicación Web y el sistema, ya que la herramienta a desarrollar no provee un módulo para validar ese tipo de entradas.
- La creación de los modelos de datos que analiza la aplicación deben ser creados de manera normalizada²⁶ de acuerdo al estándar ISO8601 para el manejo de fechas.

²⁶ CÁRCAMO, S. José. Oracle a su alcance. Escuela de Ingeniería de Sistemas e Informática. Colombia: Universidad Industrial de Santander, 2000.

Figura 9. Casos de uso definidos para los actores Público, Cliente y Administrador en la etapa de Gestión del Modelado.



Requisitos no funcionales

Restricciones de diseño (Anexo B)

- El sistema requerirá de tres tipos de hardware para su correcto funcionamiento en una red Intranet o para su publicación en Internet: Un servidor de aplicaciones y de Base de datos, la infraestructura de comunicaciones correspondiente y los clientes.

- El sistema requerirá de Microsoft Windows Server (Cualquier versión mayor o igual a Windows 2000 Server), IIS, Microsoft Internet Information Server y Microsoft SQLServer 2000 o 7 ó MSDES para la ejecución de sus funciones de la manera esperada.

Manejo de excepciones y validaciones

- El sistema provee una metodología para la creación de validaciones de manera automática, de manera que se puedan evitar en su mayoría las situaciones de error generadas con la interacción del usuario.

Las validaciones generadas por el sistema de manera automática se muestran en la tabla 4 y se describe brevemente la forma en que se genera la validación en el mismo.

En algunos casos, cuando las entradas al sistema sean incorrectas, el sistema puede presentar errores al usuario y no realizar las operaciones requeridas. Estos errores se deben principalmente a que los datos entrados por el usuario no cumplen con las características expresadas en el Anexo A.

Restricciones de Rendimiento

- El rendimiento del producto final dependerá en la medida en que la configuración realizada al sistema de meta información y el diseño del modelo de datos se ajuste a las capacidades finales del sistema, de su complejidad, del número de tablas creadas y de la información de meta modelado ingresada.

Tabla 4. Resumen del manejo de validaciones del sistema

VALIDACIÓN	CÓMO SE GENERA
Campo requerido	Se genera de manera automática cuando el campo se declara como no nulo
Campo numérico	De manera automática el sistema genera validaciones para los rangos de los números que se generan en el sistema
Validación Check	En SQLServer se pueden crear

Validación fechas	validaciones de tipo Check Constraint, son especialmente usadas para números y es usada para validar rangos. Ya que frecuentemente el uso de fechas representa un problema, el sistema provee un sistema para manejarlo de manera estándar ²⁷ ,
SQL Injection²⁸	Este término se refiere a cuando una persona de manera intencional o no, incluye instrucciones SQL dentro de los textos que ingresa al sistema, haciéndolo fallar. Estas entradas son validadas y el sistema no falla cuando estos valores se ingresan

Requisitos de seguridad

- El sistema no tiene un límite aparente de clientes que soportar; sin embargo, esta limitante puede depender de la configuración del servidor en donde se encuentre instalado y de la configuración del servidor Web, es decir, la capacidad de mantener clientes concurrentes. Adicionalmente, se tiene como restricción el número de clientes concurrentes que soporte SQLserver.
- El sistema maneja una clasificación de permisos de acceso que dependerán de la configuración de seguridad que el actor *Administrador* desarrolle para la aplicación Web.
- El sistema requerirá de una configuración de manejo de seguridad adecuado del lado del servidor; es decir, donde están ubicados los archivos ASP del sistema y la base de datos, para restringir el acceso físico a estos elementos mediante una correcta administración de las cuentas de Windows NT.

4.2.1.5 Identificación de riesgos. Uno de los aspectos más destacables del PUR es el tratamiento que hace de los riesgos involucrados en los proyectos. Los riesgos influyen en la planificación de una propuesta, por esto el PUR es un método de desarrollo dirigido, entre otras cosas, por los riesgos.

²⁷ Estándar ISO8601, Se encuentra en los libros en línea de SqlServer, palabra clave “cast”

²⁸ Es una forma muy común de romper claves en sitios Web

Teniendo en cuenta lo anterior, se debe llevar a cabo la identificación y reducción de los riesgos a lo largo de todo el proyecto para evitar construir una arquitectura poco robusta. Los riesgos identificados se presentan a continuación:

Riesgos. El comportamiento del sistema, las capacidades, las interfases de usuario y el manejo de la información, han sido pensados de manera que permitan enlazar rápidamente todos los elementos del sistema con diferentes modelos relacionales de bases de datos que los usuarios definan. Los riesgos radican en una deficiente configuración de la meta información que impida la correcta conexión entre las capas del sistema y de un pobre modelado del lado del actor que desarrolla la aplicación Web con la ayuda de la herramienta.

En cuanto a este aspecto, se ha declarado como requisito de datos el cumplimiento de una serie de características de los datos de entrada como estandarización de parámetros, que permitan que los componentes del sistema y la página Web diseñada puedan enlazarse correctamente y eviten la generación de listados u operaciones entre registros erróneos.

Respecto a los riesgos de ampliación y mantenimiento. Cuando se actualiza el modelo de datos la aplicación toma los cambios de manera automática²⁹; sin embargo, el sistema no maneja la integridad de los datos ya que este proceso es responsabilidad del diseñador del modelo; por consiguiente, no se garantiza la no pérdida de datos cuando se realicen operaciones sobre el modelo de datos; de hecho, ni siquiera es garantizado por SQL Server.

Ampliar y complementar el modelo de datos de la aplicación en general tiene un efecto negativo sobre el desempeño general de la aplicación. Cuando se amplía el modelo se deben crear los nuevos enlaces en el sitio Web³⁰ para poder visualizar la información añadida al modelo.

²⁹ Al darle clic sobre la opción “Actualizar Motor”

³⁰ Ver la sección “Creación de enlaces entre la aplicación de modelado y el sitio Web anfitrión”

4.3. MODELADO DE DATOS.

En esta etapa de modelado se obtiene el flujo de información necesario para implementar la función de gestión.

El objetivo fundamental de esta etapa es obtener las descripciones del proceso, refinando las acciones declaradas para los casos de uso, con el objeto de cumplir con los requisitos funcionales y así estructurar una arquitectura candidata.

En esta etapa también se incluirá la descripción de los datos del sistema; es decir, el modelo entidad relación del sistema.

4.3.1. Modelo de Análisis. El modelo de análisis se construye a partir de los casos de uso que se han identificados en la etapa anterior.

El modelo de análisis permite esbozar la arquitectura candidata del sistema. Su estructura está guiada por la especificación de las secuencias de acciones a efectuar por el sistema, incluyendo variaciones y validaciones de los datos.

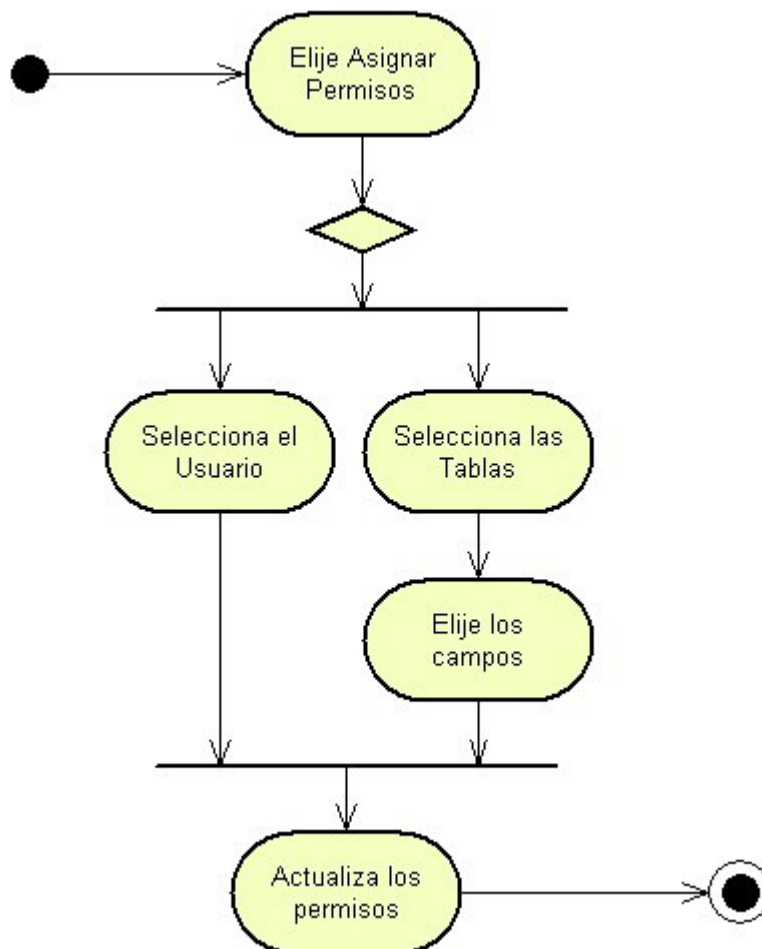
Para realizar el modelo de análisis se realizaron diagramas que corresponden a las estructuras utilizadas en el Proceso Unificado; sin embargo, dado que la tecnología utilizada (ASP 3.0) no soporta el desarrollo de clases, este no será orientado a objetos y será procedimental. Por ésta razón, para esta etapa se omitirán los diagramas de colaboración y únicamente se definirán los diagramas de actividades que mostrarán los procedimientos o flujo de sucesos de la herramienta.

4.3.1.1 Análisis de los Casos de Uso. Los casos de uso que se analizarán corresponden a aquellos que representan mayores riesgos de implementación y son los procesos de asignación de permisos y la realización del meta modelado.

Es importante recordar que los casos de uso se irán refinando a medida que se avance en las etapas de modelado del DRA hasta alcanzar el modelo de diseño necesario para la implementación del sistema en la etapa de Generación de la Aplicación.

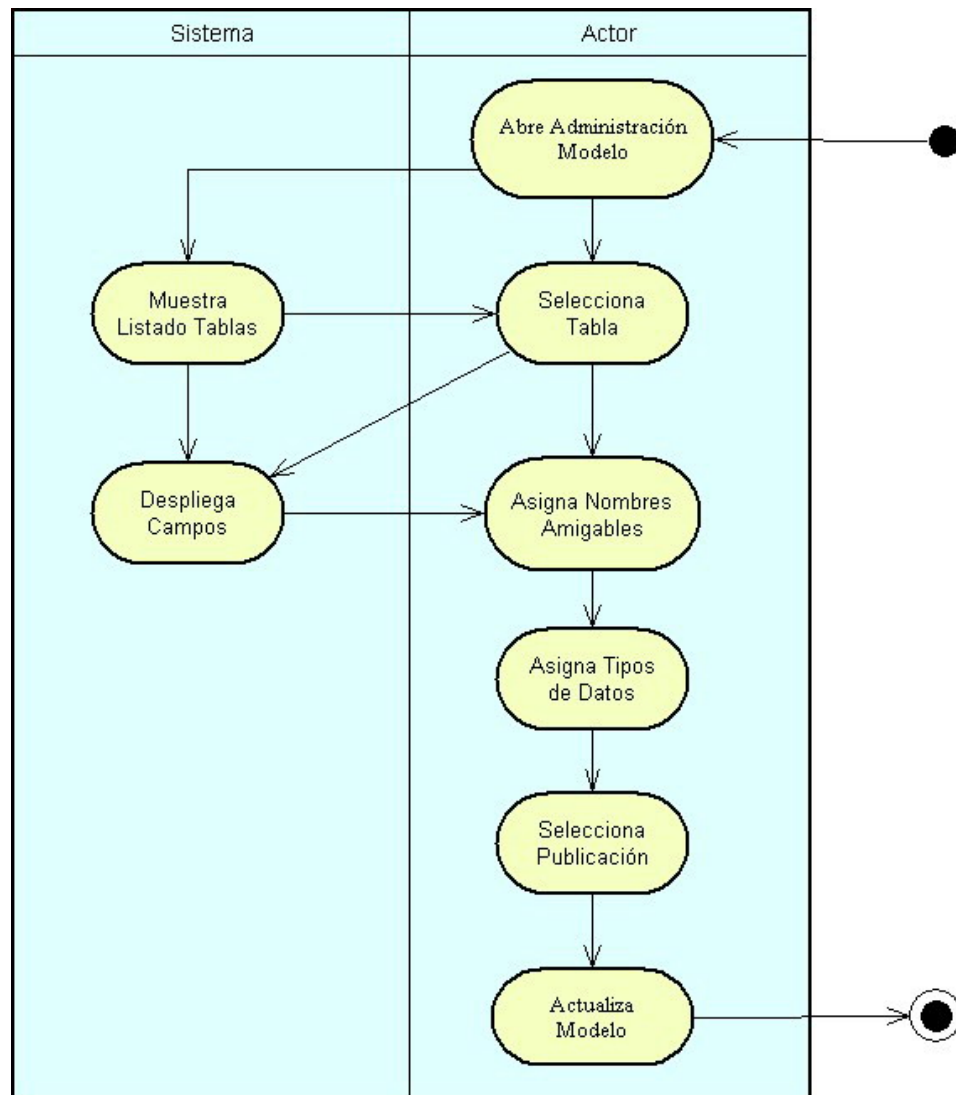
La Figura 10 muestra el diagrama de actividades para el caso de uso *Definir Permisos*. La actividad inicia con la selección de la opción “Asignar Permisos”. Seguidamente, el flujo sufre una bifurcación en la cual el actor *Administrador* deberá seleccionar tanto el usuario (*Cliente* o *Público*) como las tablas a las cuales este tendrá acceso. La selección de las tablas viene precedida por la acción de elegir los campos que se mostrarán. Una unión de control se efectuará para continuar con la acción de actualización de la información y concluir con el proceso de asignación de permisos.

Figura 10. Diagrama de actividades para el CU- Definir Permisos.



La Figura 11 muestra el diagrama de actividades para el caso de uso *Realizar Metamodelado*. Inicia cuando el *Administrador* selecciona la opción “Administrar Modelo”, el sistema desplegará un listado de las tablas que ha obtenido de la integración con la base de datos que ha sido diseñada en SQL Server; a continuación, el actor selecciona una tabla del listado y el sistema procederá a mostrarle la información de los campos que contiene. El actor puede realizar tres actividades con dichos campos: Asignar nombres amigables, Asignar Tipos de datos y elegir si desea que sean publicados o no. La actividad concluye cuando el actor oprime el botón “Actualizar”.

Figura 11. Diagrama de actividades para el CU- Realizar Metamodelado.



4.3.2 Descripción de datos. El sistema constará de los siguientes elementos, tomando como base la distribución basada en capas expuesta en el numeral 3.2.2.2.

Capa de Acceso a datos

En esta capa se construirán los elementos respectivos al análisis de las tablas del catálogo del sistema de las bases de datos Microsoft SqlServer 2000. Allí residen las consultas y procedimientos almacenados que permiten la comunicación eficiente con Microsoft SqlServer. Gran parte de esta capa estará ubicada dentro del mismo motor de Base de datos de Microsoft Sqlserver y será instalado en cada base de datos a administrar; esto se hace con el objetivo de maximizar el desempeño de la aplicación.

En esta capa de datos también se incluye el manejo de la información de meta modelado, la cual está en la base de datos a administrar como tablas adicionales con formatos de nombres especiales (ya que estas tablas no deben ser visualizadas directamente por los usuarios finales).

Para construir esta capa se crearán algoritmos de análisis que manejen la siguiente información:

- *Condiciones de Chequeo:* las cuales mostrarán las condiciones que establece el *Administrador* en cada campo cada vez que se quiere insertar o modificar un valor en un registro de una determinada tabla; estas condiciones sirven para generar validaciones del lado del cliente
- *Llaves primarias:* El sistema analizará las llaves primarias para determinar el campo sobre el cual se realizarán las búsquedas y creará los enlaces a las diferentes tablas (página para generar listados).
- *Estructura de Campos:* El sistema analizará internamente la composición de cada tabla, sus nombres de campos, sus tipos de datos, su longitud máxima, etc. Esto permitirá generar dinámicamente los formularios de edición y la creación de nuevos registros.

- Llaves foráneas: El sistema analizará las llaves foráneas de la base de datos para incluir los datos de una tabla relacionada en el formulario de la tabla actual; por ejemplo, si se tiene la tabla ciudad y departamento, en el momento de crear o de editar una entrada de la tabla ciudad, automáticamente aparecerá un selector con la información de los departamentos asociados por la relación a la tabla departamentos.
- Tablas intermedias: El sistema utilizará el análisis de las llaves foráneas enriquecido con la meta información para determinar dinámicamente las relaciones muchos a muchos encontradas en las diferentes tablas.
- En la capa lógica se encapsulará todo el procesamiento necesario para llevar a cabo las tareas de:
 - Listado de registro de una tabla, con opciones de paginación y filtro: Si se lista una tabla y el campo dado como amigable es una llave foránea, el valor asociado a esta en la tabla asociada será traído automáticamente.
 - Generación dinámica de formularios de edición y creación de registros: Se generarán a partir de la información obtenida en la capa anterior.
 - Manejo automático de relaciones muchos a muchos: una vez hecho el análisis de la información para determinar las tablas intermedias y las tablas del otro lado de la relación muchos a muchos³¹, se generarán los enlaces para ver la información asociada y crear registros en la tabla intermedia, lo que permitirá asociar directamente los elementos de una tabla con la otra. El proceso anteriormente descrito garantiza integridad referencial³².
 - Manejar los diferentes tipos de datos y su presentación a los actores que interactúan con el sistema: esto permitirá mostrar información en formato “solo lectura” para usuarios autorizados solo para dicho proceso, y formularios para usuarios con permisos de escritura sobre la tabla.
 - Manejo de grupos de campos³³ para presentar la información a los actores de manera congruente y más fácil de entender: esto se incluye pensando en tablas

³¹ En el modelo entidad relación siempre se emplea una tabla intermedia para representar una relación muchos a muchos.

³² La integridad referencial es un sistema de reglas que utiliza el motor de Base de datos para garantizar que las relaciones entre los registros de tablas relacionadas son válidas y que no se eliminan ni modifican accidentalmente datos relacionados.

³³ Esto permitirá la paginación de la información a desplegar en la página Web.

con gran cantidad de campos. El proceso de ver todos estos datos en una sola presentación es muy tedioso para el usuario.

Metamodelado

Para enriquecer y mostrar los datos a los usuarios se incluirá la siguiente información de meta modelado:

- Tabla de perfiles de usuario.
- Tabla de usuarios del sistema.
- Información acerca de los permisos para cada grupo y cada tabla.
- Tabla de configuración y parámetros generales del sistema.
- Información de los campos especiales. Por ejemplo, para el campo “fecha” se mostrará un calendario (generado mediante JavaScript) que facilite la selección de este parámetro.
- Metainformación acerca de cada una de las tablas. Por ejemplo, nombres amigables (a partir de los nombres establecidos en la base de datos) que permitan mostrar información al usuario final para facilitar su entendimiento.
- Tabla de tipos que será diseñada para minimizar el número de tablas a crear en el modelo de datos. Por ejemplo, para manejar un listado fijo de tipos de identificación (C.C., T.I., Cédula Extranjería, etc.) o estados civiles (Soltero, casado, divorciado, viudo, unión libre, etc.). Esta información se asociará directamente con el campo involucrado, ya que al generar dinámicamente el formulario este deberá analizar si tiene elementos en la tabla de tipos.

4.3.2.1 Entidad de datos. Las figuras que se presentan a continuación muestran una esquematización del modelo entidad relación de la herramienta. Este modelo se ha fraccionado de acuerdo a sus características y funciones dentro del sistema así: (a) Búsquedas especializadas, (b) Versiones desconectadas de las vistas de análisis, (c) Agrupación de los campos de la tabla, (d) Administración de Usuarios y Grupos de Usuarios, (e) Modelado de los datos de las tablas y (f) Almacenamiento de las características de tablas y columnas (campos).

Figura 12. Parte del Modelo Entidad Relación de la Herramienta para fines de Búsquedas Especializadas.

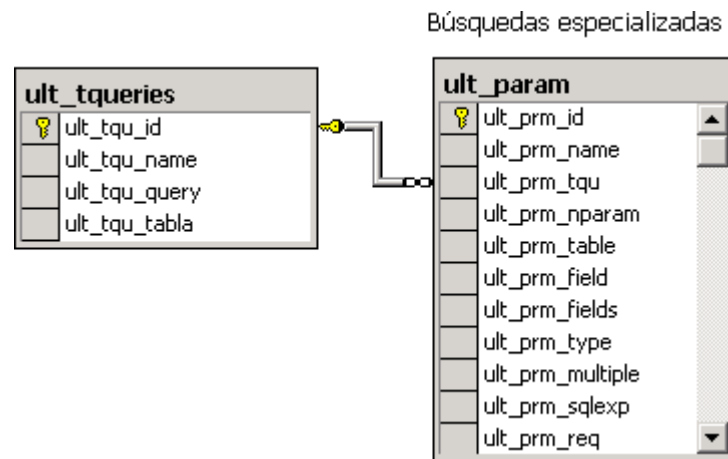


Figura 13. Parte del Modelo Entidad Relación de la Herramienta para la vista de análisis.

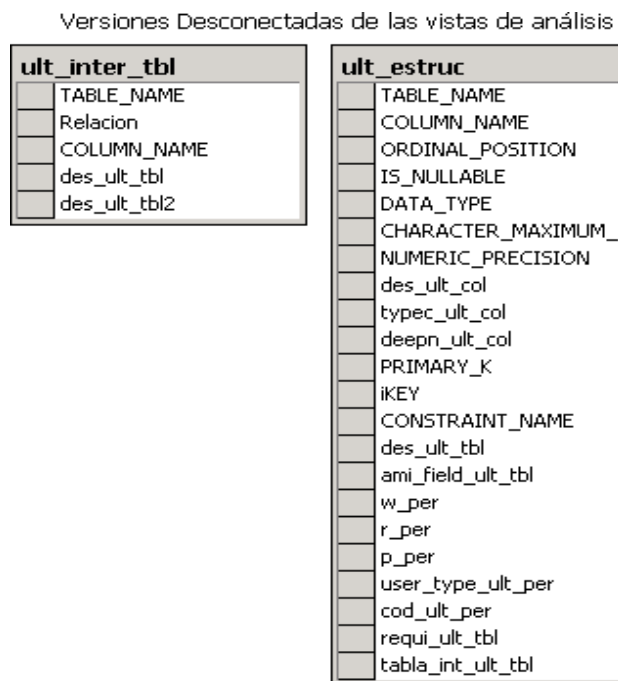


Figura 14. Parte del Modelo Entidad Relación de la Herramienta para la agrupación de campos.

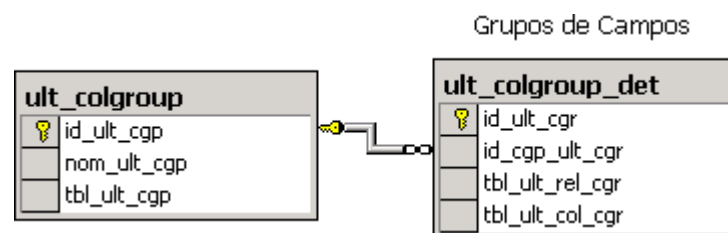


Figura 15. Parte del Modelo Entidad Relación de la Herramienta que permite la administración de Usuarios y Grupos.

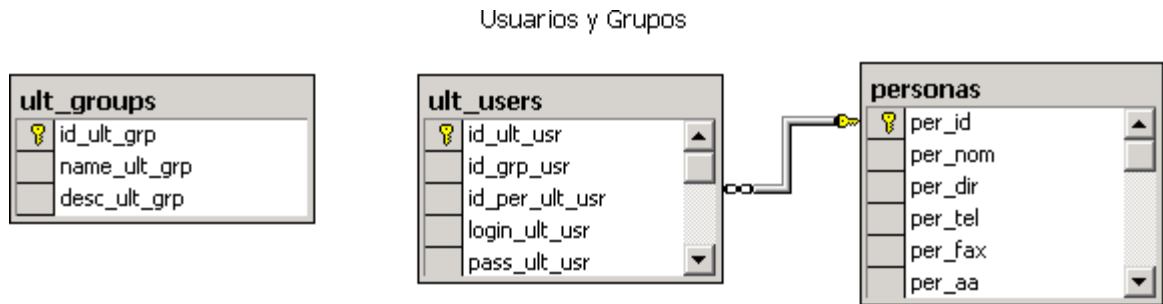


Figura 16. Parte del Modelo Entidad Relación de la Herramienta para fines de modelado de datos.

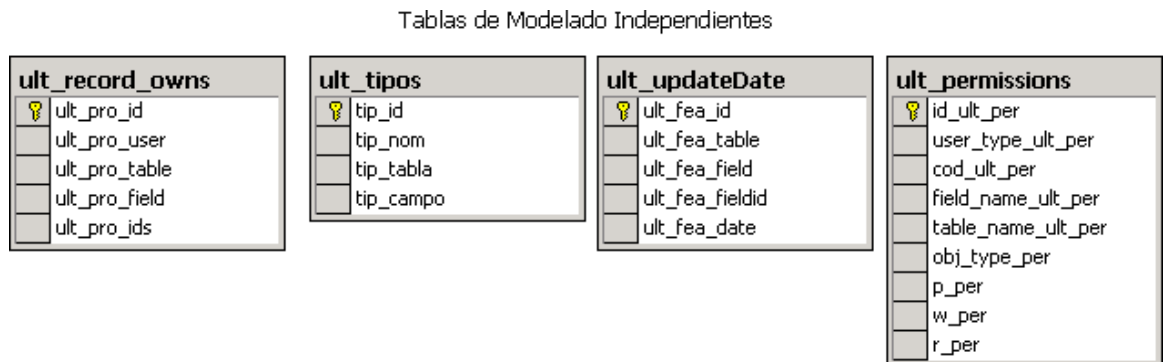
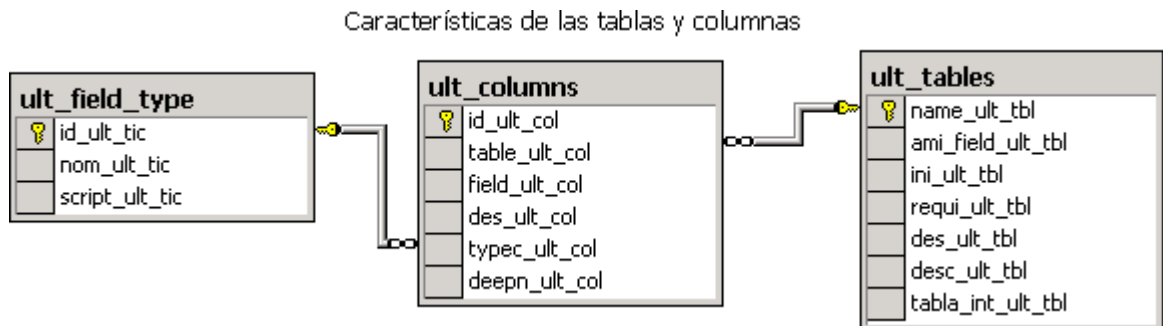


Figura 17. Parte del Modelo Entidad Relación de la Herramienta que almacena las características de tablas y columnas (campos).



4.4 MODELADO DEL PROCESO.

En la etapa de modelado del proceso se logra obtener el flujo de información necesario para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir o recuperar datos de la base de datos.

En esta fase se da forma a la arquitectura del sistema para que soporte tanto los requisitos funcionales como los no funcionales. Por ello, el objetivo fundamental en la fase de diseño es hacer un bosquejo de una estructura preliminar que podría convertirse en la arquitectura del sistema. Como resultado de este flujo de trabajo, se obtiene un modelo de diseño y un modelo de despliegue partiendo del análisis de los casos de uso relevantes anteriormente mencionados.

4.4.1 Modelo de Diseño. El diseño es el flujo de trabajo donde se especifican cuáles son los componentes e interfaces que se desarrollarán. Para ello se obtendrá un modelo de diseño basado en el refinamiento de los casos de uso o actividades que desarrollan los actores del sistema, a partir del análisis realizado sobre los mismos.

Con el modelo de diseño se espera poder consolidar una arquitectura que sustente la funcionalidad que se irá incorporará con los casos de uso de diseño. Con ello se logra concluir la especificación de requisitos, la elaboración de una arquitectura estable y poder hacer un seguimiento de los riesgos identificados y la aparición de otros nuevos para incluir las mejoras necesarias.

Como el sistema no está orientado a objetos, no se utilizará el diagrama de paquetes para representar los módulos que lo componen; sin embargo, en el numeral 4.3.2.1 se ha especificado la forma en que se agrupan los componentes de la herramienta representados en el modelo entidad relación.

4.4.1.1 Modelo de Casos de uso de diseño. A continuación, se presentan aquellos actores que fueron identificados y que actúan sobre el sistema; también se han especificado la mayor parte de los casos de uso del sistema.

Actores:

Cliente: Las actividades de los actores *Público* y *Cliente* descritos en el numeral 4.2.1.3 han sido consolidadas en un único usuario denominado como éste último. Esto se debe a que las funciones de dichos actores representan un único usuario indirecto de la herramienta, que está enlazada a la aplicación Web que se desarrolle, y por consiguiente, son usuarios directos pero de esta última, cuya funcionalidad a implementar no es tema del presente documento.

Administrador: La descripción referente a este actor se encuentra en el numeral 4.2.1.3.

Cabe aclarar que la herramienta permite la configuración de varios tipos de usuarios de la aplicación Web, de acuerdo a los permisos (lectura, escritura) de acceso a la información de las tablas; sin embargo, el acceso a la herramienta como tal únicamente lo tienen: de manera directa para la configuración y administración de la base de datos, el *Administrado*; el *Cliente* lo hace de manera indirecta, cuando recorre la aplicación Web, ya sea que este haya sido clasificado en cualquiera de los grupos de usuarios configurados por el desarrollador en el modo de administración del sistema.

Casos de uso

A continuación se especifican los casos de uso del sistema que se consideran serán implementados:

TABLA 5. Casos de uso a Implementar

<i>Nombre Caso de Uso</i>	Integrar Base de Datos
<i>Propósito</i>	Permite integrar la base de datos que ha sido diseñada por el desarrollador de la aplicación Web a la herramienta de administración.
<i>Precondiciones</i>	El Administrador debe haber realizado el modelo entidad relación de la base de datos en SQL Server 2000 ó 7.
<i>Actor iniciador</i>	<i>Administrador</i>
<i>Flujo principal</i>	1. El actor ejecuta el script de instalación del sistema. 2. El actor copia los archivos de la aplicación a la carpeta

	<p>destinada para el sitio Web que se desea crear.</p> <p>3. El actor crea un directorio virtual en el Internet Information Server.</p> <p>4. El actor modifica el archivo conn.asp, ubicado en el directorio dB de la estructura de directorios asignando el valor de la cadena a la variable ProvStr ubicada la línea 13 de ese archivo.</p> <p>5. El actor abre el archivo ult_const.asp, modifica las líneas 8 y 9 respectivamente y asigna a las variables ult_sav_path y ult_viw_path los caminos en los que se desea almacenar los archivos anexos y el camino de directorio virtual que debe seguir para observar el contenido del archivo.</p>
Requisitos especiales	<ul style="list-style-type: none"> • El actor debe conocer las características que debe cumplir la estructura del modelo entidad relación de la base de datos y los requerimientos especificados en el Anexo A del presente documento. • El actor debe conocer los pasos necesarios para la creación del directorio para Windows 2000 (Anexo F).

Nombre Caso de Uso	Autenticar Usuario
Propósito	Permite identificar que el usuario que va a ingresar a la herramienta sea el actor calificado como <i>Administrador</i> , de manera que se protejan los datos de las tablas.
Precondiciones	Ingresar a la herramienta a través del navegador Web.
Actor iniciador	<i>Administrador</i>
Flujo principal	<ol style="list-style-type: none"> 1. El actor ingresa el nombre de usuario 2. El actor ingresa la contraseña 3. El actor selecciona el tipo de usuario del sistema: Administrador 4. El actor da clic en el botón entrar 5. El sistema muestra la aprobación de la operación. <p>Concluye el CU y continúa con el <u>CU Administrar Perfiles</u></p>
Flujo excepcional	<p>Si se ingresa un nombre de usuario y/o una contraseña o si se ha seleccionado el tipo de usuario en forma errónea el sistema ejecutará el siguiente flujo de sucesos:</p> <ol style="list-style-type: none"> 1. Se mostrará una pagina donde se le informa al actor que no está autorizado para ingresar al modulo administrativo. 2. Dar clic para regresar a la página principal de la herramienta e intentar nuevamente el ingreso.

	3. Ingresar nuevamente los datos requeridos.
Estructura de control	<ul style="list-style-type: none"> En el paso 2 el sistema espera por 10 segundos para repetir el <u>CU Autenticar Usuario</u>. Si el sistema no recibe respuesta del usuario (paso 2 del flujo excepcional) lo lleva a la página principal de la aplicación Web.
Requisitos especiales	El actor debe conocer las capacidades del sistema y debe haber realizado previamente la configuración necesaria para abrir la herramienta.

Nombre Caso de Uso	Administrar Grupos de Usuarios
Propósito	Permite crear, modificar y eliminar grupos de usuarios dentro de los cuales se deberán clasificar los usuarios de la aplicación Web. Esto hace que el otorgamiento de permisos sea por grupo y no para cada uno de los usuarios.
Precondiciones	Ejecución exitosa del <u>CU Autenticar Usuario</u>
Actor iniciador	<i>Administrador</i>
Flujo principal	<ol style="list-style-type: none"> El actor despliega el menú Administración. El actor seleccionar la opción “Administrar Grupos” El sistema despliega las opciones disponibles.
Flujos secundarios	<ul style="list-style-type: none"> Crear Grupo <ol style="list-style-type: none"> El actor oprime el botón “Modificar Grupos” El sistema muestra los campos a diligenciar: Nombre del grupo y descripción del grupo. El actor ingresa los datos solicitados. El actor da clic en el botón “Guardar”. El sistema muestra un mensaje de confirmación de la creación del grupo. Actualizar o Eliminar Grupo <ol style="list-style-type: none"> El actor oprime el botón “Modificar Grupos” De la lista de opciones, el actor selecciona un grupo existente. El sistema muestra los campos diligenciados previamente en la creación del grupo. El actor selecciona la opción “Ver usuarios del grupo” El sistema despliega el conjunto de usuarios que han sido asignados al grupo. Continúa con el <u>CU Administrar Usuarios</u>
Estructura de control	<ul style="list-style-type: none"> En el paso 2 del flujo secundario: Crear Grupo y paso 3 del flujo secundario: Actualizar o Eliminar Grupo, el

	<p>sistema valida cada uno de los campos disponibles de acuerdo a los tipos de datos definidos para ellos.</p> <ul style="list-style-type: none"> • En el paso 3 del flujo secundario: Actualizar o eliminar grupo, el sistema le pedirá la confirmación de la acción en caso de elegir la opción eliminar. • En el paso 6 del flujo secundario: Actualizar o Eliminar Grupo, el sistema da la posibilidad de crear, editar o eliminar usuarios del grupo en cuestión.
Requisitos especiales	<ul style="list-style-type: none"> • El sistema por defecto tendrá definidos dos grupos de usuarios de acuerdo a los actores que se identificaron: Administrador y Cliente. • El nombre de los grupos debe estar directamente relacionado con los permisos asignados.

Nombre Caso de Uso	Administrar Usuarios
Propósito	Permite crear, modificar y eliminar usuarios de determinado grupo. Los permisos otorgados a cada uno de los usuarios se hacen efectivos dentro de la aplicación Web enlazada a la herramienta.
Precondiciones	<ul style="list-style-type: none"> • Ejecución exitosa del <u>CU Autenticar Usuario</u>. • Creación de un grupo a través del <u>CU Administrar Grupos de Usuarios</u>
Actor iniciador	<i>Administrador</i>
Flujo principal	<ol style="list-style-type: none"> 1. El actor despliega el menú Administración. 2. El actor selecciona la opción "Administrar Usuarios" 3. El sistema muestra las opciones disponibles a ejecutar.
Flujos secundarios	<ul style="list-style-type: none"> • Crear Usuario <ol style="list-style-type: none"> 1. El actor selecciona la opción "Nuevo Usuario" 2. El sistema muestra los campos a diligenciar clasificados en dos grupos: Información personal y de contacto (Nombre, Dirección, Teléfono, e-mail, Url) e Información de ingreso (Nombre de usuario, clave, confirmar clave) 3. El actor diligencia los datos. 4. El actor selecciona de un combo de opciones el grupo de usuario al que pertenece. 5. El actor da clic en el botón Crear. 6. El sistema muestra un mensaje de confirmación de la creación del usuario. • Actualizar o Eliminar Usuario

	<ol style="list-style-type: none"> 1. El sistema muestra un combo de nombres de usuarios creados. 2. El actor selecciona un usuario previamente creado de un combo de opciones. 3. El sistema muestra los campos diligenciados previamente en la creación del usuario. 4. El actor modifica o elimina datos. 5. El actor da clic en el botón Modificar o Eliminar usuario. 6. El sistema muestra un mensaje de confirmación de la acción realizada.
Estructura de control	<ul style="list-style-type: none"> • En el paso 2 del flujo secundario: Crear usuario y paso 3 del flujo secundario: Actualizar o Eliminar Usuario, el sistema valida cada uno de los campos disponibles de acuerdo a los tipos de datos definidos para ellos. • En el paso 5 del flujo secundario: Actualizar o eliminar Usuario, el sistema le pedirá la confirmación de la acción en caso de elegir la opción eliminar.
Requisitos especiales	Los identificadores, de los campos de los datos del usuario, que se encuentran en negrilla implican que su diligenciamiento es obligatorio para ejecutar posteriormente la acción sobre los registros de usuarios.

Nombre Caso de Uso	Administrar Permisos
Propósito	Permite asignar, modificar o eliminar los permisos o tipos de acceso a los datos de las tablas. Estos permisos pueden ser de tres tipos: proveedor, lectura y escritura.
Precondiciones	<ul style="list-style-type: none"> • Ejecución exitosa del <u>CU Autenticar Usuario</u>. • Creación de un grupo a través del <u>CU Administrar Grupos de Usuarios</u>. • Asignación de algún usuario al grupo mediante la ejecución del <u>CU Administrar Usuarios</u>.
Actor iniciador	Administrador
Flujo principal	<ol style="list-style-type: none"> 1. El actor despliega el menú Administración. 2. El actor selecciona la opción "Administrar Permisos" 3. El sistema muestra un combo con los grupos de usuarios creados.
Flujos secundarios	<ul style="list-style-type: none"> • Asignar Permisos <ol style="list-style-type: none"> 1. El actor selecciona un grupo de usuarios. 2. El actor da clic en el botón "Ver Permisos del grupo de usuarios" 3. El sistema despliega un cuadro que contiene la lista de

	<p>tablas obtenidas de la integración de la herramienta con la base de datos diseñada.</p> <p>4. El actor activa, para cada tabla, el tipo de acceso que tiene el grupo seleccionado haciendo clic en el recuadro que identifica a cada permiso.</p> <p>5. El actor oprime el botón Actualizar Permisos.</p> <ul style="list-style-type: none"> • Actualizar o Eliminar Permisos <p>1. El actor selecciona un grupo de usuarios.</p> <p>2. El actor da clic en el botón “Ver Permisos del grupo de usuarios”</p> <p>3. El sistema despliega un cuadro que contiene la lista de tablas obtenidas de la integración de la herramienta con la base de datos diseñada.</p> <p>4. El actor activa o desactiva, para cada tabla, el tipo de acceso que tiene el grupo seleccionado sobre cada una de las tablas haciendo clic en el recuadro que identifica a cada permiso.</p> <p>5. El actor oprime el botón Actualizar Permisos.</p>
Estructura de control	<ul style="list-style-type: none"> • Si el usuario selecciona la opción “proveedor” se activa automáticamente la opción de “lectura”. • Si el usuario selecciona la opción “escritura” se desactiva automáticamente la opción de “proveedor”. • Si el usuario desactiva la opción “escritura” se desactiva automáticamente la opción de “lectura”.
Requisitos especiales	<p>La información que el sistema despliega en el cuadro para asignar las operaciones que puede o no realizar el usuario sobre los registros, dependerá de la conclusión exitosa del <u>CU Integrar Base de Datos</u></p>

Nombre Caso de Uso	Administrar Tablas
Propósito	El caso de uso permite listar, crear o ver las propiedades de una tabla.
Precondiciones	<ul style="list-style-type: none"> • Ejecución exitosa del <u>CU Integrar Base de Datos</u>. • Ejecución exitosa del <u>CU Autenticar Usuario</u>.
Actor iniciador	<i>Administrador</i>
Flujo principal	<p>1. El actor despliega el menú Administración.</p> <p>2. El actor selecciona la opción “Administrar Tablas”</p> <p>3. El sistema despliega un listado de las tablas de la base de datos con las tres opciones disponibles a ejecutar: Listado, Nuevo y Propiedades.</p>

Flujos secundarios	<ul style="list-style-type: none"> • Listado <ol style="list-style-type: none"> 1. El actor selecciona la opción Listado. 2. El sistema muestra un listado de los diferentes campos de la tabla con las siguientes características: Nombre, Descripción y Ruta (Permite anexar un archivo). 3. El actor selecciona la opción “Ver” 4. El sistema mostrará los campos de la tabla de manera que el contenido de sus campos pueda se modificado. 5. El actor realiza las modificaciones y da clic en el botón “Actualizar” • Nuevo <ol style="list-style-type: none"> 1. El actor selecciona la opción Nuevo. 2. El sistema muestra los campos de la tabla en formato de edición. Los campos mostrados son: Tipo de Anexo, Tipificación, Nombre, Descripción, Ruta del Archivo Anexo, Nombre de la Tabla, Identificador, Publicar este anexo (Si /No). 3. El actor diligencia los campos. 4. El actor oprime el botón Guardar. • Propiedades <ol style="list-style-type: none"> 1. El actor selecciona la opción Propiedades. 2. El sistema despliega la información general de la tabla: Nombre y campos asociados.
Flujo excepcional	<ul style="list-style-type: none"> • Después del paso 2 del flujo secundario: Listado el actor puede seleccionar la opción “Borrar”. • Después del paso 2 del flujo secundario: Propiedades se puede continuar con el <u>CU Renombrar Tablas</u>.

Nombre Caso de Uso	Renombrar Tablas
Propósito	Asigna un nombre amigable a cada una de las tablas y campos de tal manera que estos sean los identificadores que el usuario de la aplicación Web visualice en los formularios.
Precondiciones	<ul style="list-style-type: none"> • Ejecución exitosa del <u>CU Integrar Base de Datos</u>. • Ejecución exitosa del <u>CU Autenticar Usuario</u>.
Actor iniciador	<i>Administrador</i>
Flujo principal	<ol style="list-style-type: none"> 1. El actor despliega el menú Administración. 2. El actor selecciona la opción “Administración de Tablas”

	<p>3. El sistema muestra un combo con los nombres de las tablas.</p> <p>4. El actor selecciona una tabla.</p> <p>5. El sistema muestra una caja de texto para escribir un nombre amigable a la tabla.</p> <p>6. El actor introduce el nombre amigable.</p> <p>7. El sistema muestra la siguiente información de la tabla: Campos, si es una tabla intermedia y si la tabla es requerida para su publicación.</p> <p>8. El actor oprime el botón "Actualizar"</p> <p>9. Continúa con el <u>CU Caracterizar Campos</u></p>
Flujo excepcional	<ul style="list-style-type: none"> • En el paso 5 el actor puede modificar o eliminar el nombre amigable de la tabla si previamente se ha configurado. • En el paso 7 el actor puede seleccionar los campos de la tabla que desea mostrar al usuario de la aplicación Web • Si una tabla contiene muchos campos, el actor podrá agruparlos de manera que representen una manera más sencilla de desplegar la información ejecutando el <u>CU Agrupar Campos</u>.
Estructura de control	<ul style="list-style-type: none"> • Los nombres de campos correspondientes a llaves foráneas son identificados con el color rojo. • El sistema le indicará al actor que si elige un campo resaltado con rojo permitirá que los campos de la tabla relacionada se muestren en el formulario.
Requisitos Especiales	<p>Si una tabla se ha declarado como intermedia, los números que en ella residan no se mostrarán al <i>Cliente</i>, en cambio se desplegará la información relacionada con estos números a partir de otras tablas.</p>

Nombre Caso de Uso	Caracterizar Campos
Propósito	Se utiliza este caso de uso para asignar las características de los campos de las tablas, es decir, el actor puede asignar un nombre amigable al campo y seleccionar el tipo de dato para que de acuerdo a esta última especificación el sistema acondicione la salida de dicho dato por pantalla.
Precondiciones	Selección de una tabla específica en el <u>CU Renombrar Tablas</u> .
Actor iniciador	Administrador

Flujo principal	<ol style="list-style-type: none"> 1. El actor da clic sobre la opción “Nombres Amigables de los campos”. 2. El sistema despliega un esquema que contiene 3 columnas con espacios disponibles para diligenciar la información de los campos de la tabla seleccionada: Nombre del campo, Nombre Amigable y Script (tipo de dato). 3. El actor digita el nombre amigable de la tabla y selecciona un Script. Esta acción la ejecuta para cada uno de los campos. 4. El actor oprime el botón “Actualizar” 5. El sistema muestra un mensaje de confirmación de la acción.
Flujo excepcional	En el paso 3 el actor puede modificar o eliminar el nombre amigable de cada campo si previamente se ha configurado.
Estructura de control	El sistema genera un combo (listado) con los datos de la tabla asociada cuando se trate de llaves foráneas.
Requisitos especiales	<ul style="list-style-type: none"> • Para la configuración del nombre amigable de un campo cuyo tipo de datos es bit, se le deberá asignar en el nombre amigable la pregunta que se muestra al usuario y las opciones de respuesta (por ejemplo, ¿desea publicar esta tabla? ¥Si¥No¥). • En caso que el campo requiera la opción de adjuntar archivos, el actor deberá seleccionar el Script de tipo Attach del combo de opciones. • Se aconseja que el campo que corresponde a la llave primaria no se le asigne un nombre amigable, porque normalmente este campo no se lista en los formularios. • El actor no podrá utilizar los siguientes tipos de datos de SQL Server: Binary, Image, sql_variant, timestamp, unitqueidentifier, varbinay y tipos de datos creados por el usuario.

Nombre Caso de Uso	Anexar Archivos
Propósito	Se utiliza este caso de uso para anexar archivos a las tablas de la base de datos.
Precondiciones	Ejecución del flujo secundario “Listado” con el <u>CU Administrar Tablas</u> .
Actor iniciador	<i>Administrador</i>
Flujo principal	<ol style="list-style-type: none"> 1. El actor oprime el botón “Browse”. 2. El sistema muestra la ventana para permitir la búsqueda del archivo a anexar. 3. El actor selecciona el archivo y oprime el botón “Open”.

	<p>4. El sistema muestra el nombre del archivo a un lado del botón "Browse".</p> <p>5. El sistema muestra las opciones disponibles a ejecutar: Nuevo, Actualizar y Borrar.</p>
Flujo secundario	<p>Nuevo/ Actualizar/ Borrar</p> <p>1. El actor oprime el botón de acuerdo a la operación a realizar.</p> <p>2. El sistema actualiza la ventana con la información del anexo.</p>
Flujo excepcional	<p>Después del paso 4 el actor puede dar clic sobre el archivo adjunto para mostrar su contenido.</p>
Estructura de control	<ul style="list-style-type: none"> • Si la tabla seleccionada es intermedia, la manera de desplegar la información al actor <i>Cliente</i> se mostrará mediante combos de los datos relacionados a la tabla. • Los archivos que se crean como anexos se suben al servidor internamente con una nomenclatura especial, para evitar que dos archivos de diferente contenido pero con el mismo nombre sean sobrescritos.

Nombre Caso de Uso	Agrupar Campos
Propósito	<p>Este caso de uso tiene como fin agrupar los campos de una tabla de acuerdo a sus características comunes de manera que representen una manera más sencilla de desplegar la información.</p>
Precondiciones	<p>Ejecución del flujo secundario "Propiedades" con el <u>CU Administrar Tablas</u>.</p>
Actor iniciador	<i>Administrador</i>
Flujo principal	<p>1. El actor selecciona la opción Agrupación de Campos.</p> <p>2. El sistema despliega el nombre de la tabla, un listado de los campos y las tablas asociadas. Además muestra el espacio disponible para administrar los grupos: Nuevo, eliminar, actualizar.</p> <p>3. El sistema muestra un combo con los grupos de campos ya creados.</p> <p>4. El actor selecciona un grupo existente.</p> <p>5. El sistema lista los campos relacionados a este grupo y las tablas asociadas.</p>
Flujos secundarios	<ul style="list-style-type: none"> • Nuevo <p>1. El actor escribe el nombre del nuevo grupo.</p> <p>2. El actor selecciona la opción "Nuevo Grupo"</p> <p>3. El sistema adiciona el nombre del grupo al listado de los</p>

	<p>ya existentes.</p> <p>4. El actor selecciona los campos que pertenecerán al campo creado.</p> <p>5. El actor selecciona las tablas asociadas al grupo de campos.</p> <p>6. El actor oprime el botón “Actualizar”.</p> <ul style="list-style-type: none"> • Eliminar o Actualizar <p>1. El actor selecciona un grupo existente del combo de opciones.</p> <p>2. El sistema despliega la información general del grupo: Nombre, campos y tablas asociadas.</p> <p>3. El actor modifica o elimina el nombre del campo y/o los campos y tablas asociadas.</p> <p>4. El actor oprime el botón “Eliminar” o “Actualizar”.</p>
Flujo excepcional	<p>En el paso 3 el actor puede continuar con cualquiera de los dos flujos secundarios.</p>
Requisitos especiales	<ul style="list-style-type: none"> • Es responsabilidad del administrador determinar las características comunes de los campos que le permita agruparlos. • Es importante que las tablas seleccionadas para asociarse a los campos de un grupo sean aquellas señaladas con un guión que son las tablas principales. • El actor determinará si un campo puede pertenecer a varios grupos. Los efectos de la presentación de la información de esta forma es responsabilidad directa del actor.

Nombre Caso de Uso	Administrar Tipos
Propósito	Ahorrar la creación de tablas innecesarias en el modelo de datos
Precondiciones	<ul style="list-style-type: none"> • Ejecución exitosa del <u>CU Integrar Base de Datos</u>. • Ejecución exitosa del <u>CU Autenticar Usuario</u>.
Actor iniciador	<i>Administrador</i>
Flujo principal	<p>1. El actor despliega el menú Administración.</p> <p>2. El actor selecciona la opción “Administración de Tipos”.</p> <p>3. El sistema muestra un listado de las tablas existentes.</p> <p>4. El actor selecciona una tabla.</p> <p>5. El sistema muestra el campo al cual se le pueden asignar tipos.</p> <p>6. El actor selecciona el campo.</p> <p>7. El sistema despliega los tipos que tenga asignados.</p>

Flujo excepcional	En el paso 7 el actor puede continuar con cualquiera de los flujos secundarios.
Flujo Secundario	<ul style="list-style-type: none"> • Nuevo <ol style="list-style-type: none"> 1. El actor escribe el nombre del tipo. 2. El actor oprime el botón "Actualizar" 3. El sistema adiciona el nuevo tipo al listado existente. • Borrar <ol style="list-style-type: none"> 1. El actor selecciona un tipo a eliminar. 2. El sistema ubica en la caja de texto el nombre del tipo. 3. El actor oprime el botón "Borrar". 4. El sistema elimina el tipo del listado existente. • Actualizar <ol style="list-style-type: none"> 1. El actor selecciona un tipo a modificar. 2. El sistema ubica en la caja de texto el nombre del tipo. 3. El actor modifica el nombre y oprime el botón "Actualizar". 4. El sistema actualiza el nombre del tipo en el listado existente. • Quitar Asociación <ol style="list-style-type: none"> 1. El actor da clic en el botón "Quitar Asociación" 2. El sistema elimina todos los tipos listados asociados al campo.
Requisitos especiales	El tipo de dato con el que se crean los campos a los cuales se les quiere asociar es de tipo entero (int).

Nombre Caso de Uso	Autorizar Publicación
Propósito	El actor utiliza este caso de uso para autorizar la publicación de los registros de las tablas en la aplicación Web.
Precondiciones	Ejecución del flujo secundario "Propiedades" con el <u>CU Administrar Tablas</u> .
Actor iniciador	Administrador
Flujo principal	<ol style="list-style-type: none"> 1. El actor ubica la opción "Publicación de registros". 2. El actor da clic sobre una de las opciones disponibles: Si/No 3. El actor da clic sobre el botón "Actualizar".

Nombre Caso de Uso	Actualizar Motor
---------------------------	-------------------------

Propósito	Permite utilizar la técnica desconectada para actualizar las modificaciones realizadas sobre las tablas de la base de datos.
Precondiciones	Se pueden haber ejecutado cualquiera de los casos de uso descritos anteriormente.
Actor iniciador	<i>Administrador</i>
Flujo principal	<ol style="list-style-type: none"> 1. El actor despliega el menú Administración. 2. El actor selecciona la opción "Actualizar Motor" 3. El sistema muestra la confirmación de la actualización del motor.

Nombre Caso de Uso	Realizar Búsqueda
Propósito	Proporciona un mecanismo para buscar ciertas palabras en el contenido de las tablas con gran cantidad de registros y de información.
Precondiciones	<ul style="list-style-type: none"> • Ejecución exitosa del <u>CU Integrar Base de Datos</u>. • Ejecución exitosa del <u>CU Autenticar Usuario</u>.
Actor iniciador	<i>Administrador, Cliente</i>
Flujo principal	<ol style="list-style-type: none"> 1. El actor despliega el menú Principal. 2. El actor selecciona la opción "Búsqueda". 3. El sistema muestra un listado de las tablas existentes. 4. El actor digita la(s) palabras(s) que desea buscar dentro de las tablas. 5. El actor da clic en el botón "Buscar". 6. El sistema muestra la cantidad de coincidencias encontradas y un listado de cada uno de los registros. 7. El actor selecciona una de las coincidencias. 8. El sistema despliega la información detallada de este registro.
Flujo excepcional	En el paso 4 el actor puede continuar con el flujo de Búsqueda Especializada.
Flujo Secundario	<p>Búsqueda especializada</p> <ol style="list-style-type: none"> 1. El actor selecciona la opción "Búsqueda Especializada". 2. El sistema muestra un listado de las tablas que el sistema a reconocido. 3. El actor selecciona las tablas sobre las cuales desea realizar la búsqueda. 4. El actor da clic en el botón "Buscar". 5. El sistema muestra la cantidad de coincidencias encontradas y un listado de cada uno de los registros.

	6. El actor selecciona una de las coincidencias. 7. El sistema despliega la información detallada de este registro.
Estructura de control	Cuando se ha ejecutado la búsqueda el sistema resaltará las palabras de digitadas por el usuario en la búsqueda dentro del contenido de los registros encontrados.
Requisitos especiales	Para que los actores puedan realizar la búsqueda se debe tener el servicio Index-Server de Windows y Full-text de SQLServer instalado en el equipo en donde se instale la aplicación.

Nombre Caso de Uso	Consultar Información
Propósito	Los actores utilizan este caso de uso para visualizar la información contenida en las tablas que han sido reconocidas por el sistema.
Precondiciones	<ul style="list-style-type: none"> • Ejecución exitosa del <u>CU Integrar Base de Datos.</u> • Ejecución exitosa del <u>CU Autenticar Usuario.</u>
Actor iniciador	<i>Administrador, Cliente</i>
Flujo principal	<ol style="list-style-type: none"> 1. El actor despliega el Menú Principal. 2. Selecciona la tabla sobre la cual desea desplegar la información. 3. El sistema muestra el contenido de la tabla y sus respectivas asociaciones y anexos en modo de lectura.

4.4.3 Modelo de Casos de uso completo. Se consideraron únicamente dos casos de uso para el actor *Cliente*; las demás operaciones que éste usuario pueda ejecutar no serán implementados, ya que representan las acciones que, de manera indirecta, realizan los usuarios de la aplicación Web que se integra con la herramienta y son responsabilidad de su desarrollador. Sin embargo, se debe indicar que de los *CU Administrar Permisos*, *Administrar Usuarios* y *Administrar Grupos de Usuarios* dependerá la cantidad y características de los actores que puedan definirse para hacer uso de la página Web desarrollada por el *Administrador* del sistema.

La Figura 18 esquematiza los casos de uso para el actor *Administrador y Cliente* a implementar. La expresión “Administrar” incluida en la mayoría de los casos de uso involucra las actividades de Creación, Actualización y Eliminación, las cuales han sido descritas detalladamente en el flujo secundario de cada caso de uso. Sin embargo, estas actividades no han sido especificadas en el diagrama de casos de uso completo. El *CU Realizar Búsqueda* se ha generalizado a través de una *Búsqueda Especializada* cuyos pasos de ejecución han sido incluidos en el flujo secundario de este caso de uso.

Es recomendable que el *Administrador* ejecute los casos de uso en el siguiente orden con el fin de obtener la mejor funcionalidad del sistema:

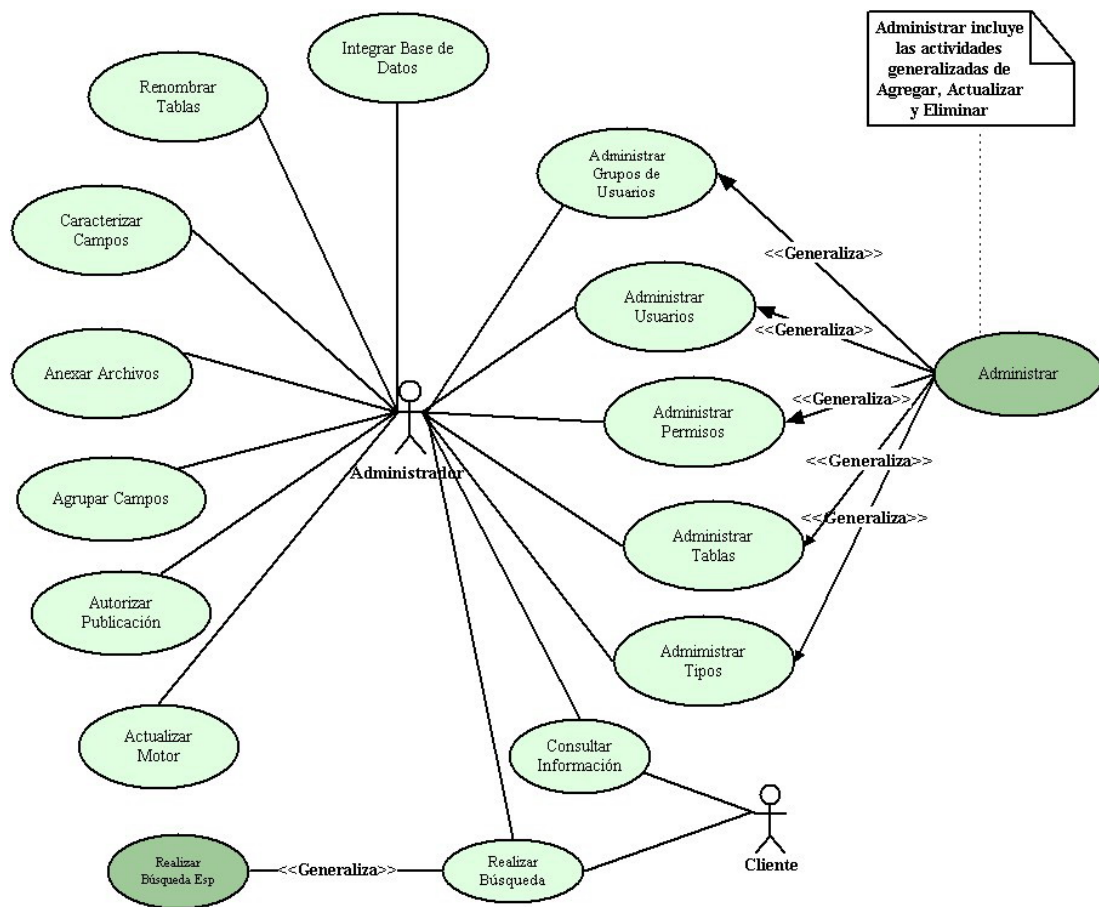
- Integrar Base de Datos
- Administrar Grupos de Usuarios
- Administrar Usuarios
- Administrar Permisos
- Administrar Tablas
- Renombrar Tablas
- Caracterizar Campos
- Anexar Archivos
- Agrupar Campos
- Administrar Tipos
- Autorizar Publicación
- Actualizar Motor

Ver la figura 18.

4.4.2 Modelo de Despliegue. Un modelo despliegue representa los módulos físicos de código, los cuales, corresponden a los elementos ejecutables del sistema y de las entidades con las cuales interactúa; muestra además cómo los componentes del sistema se comunican con otros componentes.

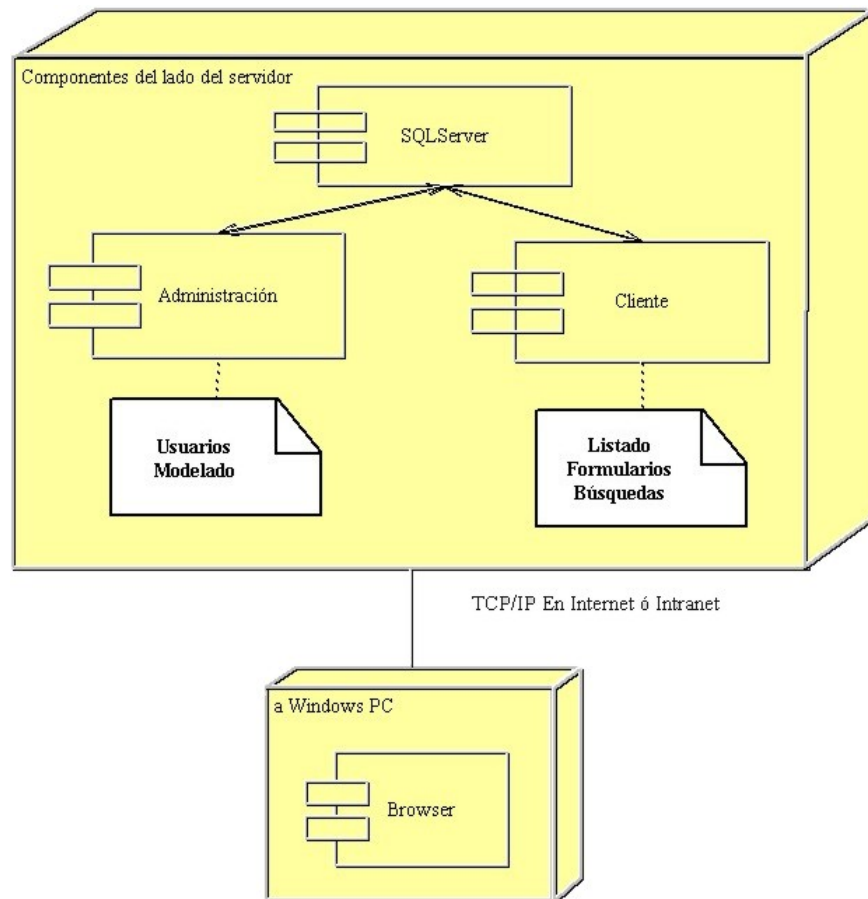
La ventaja de este modelo es que permite al desarrollador de la herramienta comprender mejor la topología de un sistema distribuido en los equipos de cómputo y tiene en cuenta los siguientes requerimientos: disponibilidad del sistema, rendimiento y escalabilidad.

Figura 18. Modelo de casos de uso completo para la etapa de Modelado del Proceso.



La herramienta se ejecuta en un servidor de Base de datos y se integra con una aplicación Web que se accede a través de una estación de trabajo (figura 19).

Figura 19. Diagrama de despliegue del modelo de diseño elaborado para la implementación del sistema.



4.5 GENERACIÓN DE LA APLICACIÓN.

En esta etapa del DRA se trabaja en la creación de nuevos componentes y en la reutilización de componentes de programas ya existentes para conformar la herramienta; esto se realiza con el fin de disminuir los tiempos de implementación, es decir, de la generación de código fuente. Cabe destacar que esta etapa se centra en la implementación de la mayor parte del sistema y en probar su funcionamiento para obtener una versión operativa inicial del sistema, en la cual se detallen todos los requisitos identificados en las etapas anteriores y en el desarrollo de la etapa que se describe.

En este punto del proyecto se procederá a la creación de los módulos del sistema necesarios para alcanzar los objetivos propuestos, partiendo del diseño desplegado en la etapa de modelado y los requisitos funcionales declarados a partir de los diagramas de casos de uso.

4.5.1 Modelado de Interfaces. En la implementación que se realizará se tendrá en cuenta la arquitectura de tres capas, para la cual se desarrollarán módulos y páginas que encierren funcionalidad específica, para ser utilizadas por las diferentes partes del sistema.

Se recuerda que ASP 3.0 no soporta el desarrollo de clases; por lo tanto, el desarrollo no será orientado a objetos y será procedimental.

4.5.1.1 Interfaces de la herramienta. La interfase de usuario de la herramienta está compuesta por un conjunto de formularios con los cuales el usuario podrá realizar la configuración de las tablas y campos de la base de datos que ha sido integrada a la herramienta. A continuación se realiza una breve descripción de las interfases más importantes.

Usuarios

La figura 20 se muestra la interfase de entrada al sistema en el cual el usuario deberá ingresar su nombre de usuario y contraseña para ingresar al Módulo Administrativo.

Figura 20. Página de ingreso al Módulo Administrativo GenApp



The image shows a login interface with the following elements:

- A text input field labeled "Clave" containing five black dots, representing a masked password.
- A dropdown menu labeled "tipo de usuario" with "Administrador" selected and a downward arrow.
- A button labeled "Entrar" (Login) positioned below the other fields.

Cuando se selecciona del menú de usuarios el elemento Administración de permisos (figura 21), el sistema presenta la página en el que se podrá seleccionar un grupo de usuarios (figura 22).

Figura 21. Menú principal del sistema.



Figura 22. Formulario del sistema en el cual se observa el listado de grupos de usuarios.

Administración de Permisos

Grupo Seleccione ▼ Ver Permiso

Tabla	Proveedor	Lectura	Escritura

Figura 23. Formulario del sistema para la Administración de permisos para un usuario hipotético: *Administrador*.

Grupo Administrador ▼ Ver Permisos del Grupo seleccionado

Modificando permisos para el Grupo: Administrador

Tabla	Proveedor	Lectura	Escritura
categorias	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
cursos	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
escuelas	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
estudiante	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
evaluacion	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
materias	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
matest	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
matprof	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
personas	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
planes	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
planmat	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
profes	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
requisitos	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Actualizar Permisos

Figura 24. Formulario del sistema para la Administración de permisos para un usuario hipotético: *Cliente*.

Grupo

Modificando permisos para el Grupo: Casuales

Tabla	Proveedor	Lectura	Escritura
categorias	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
cursos	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
escuelas	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
estudiante	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
evaluacion	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
materias	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
matest	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
matprof	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
personas	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
planes	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
planmat	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
profes	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
requisitos	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Una Vez se selecciona el grupo sobre el cual se desea trabajar, el sistema mostrará un listado de tablas y los permisos asignados a cada una de ellas. En la figura 23 se aprecian los permisos asignados sobre las tablas de la base de datos para los grupos, como por ejemplo: Administrador y en la figura 24 los permisos para un usuario potencial de la aplicación Web denominado Casual, porque solo ejerce actividades de consulta de la información.

Cuando el actor ejecuta el caso de uso *Administrar Grupos de Usuario* el sistema muestra el formulario de la figura 25. En esta página se crean los grupos, se modifican sus nombres y descripciones y se borran. Se debe tener especial cuidado al borrar grupos, ya que al eliminarlos también eliminarán los usuarios pertenecientes al mismo y los permisos otorgados.

Figura 25. Formulario del sistema para crear, modificar o eliminar Grupos de Usuario.

Grupo

Seleccione un Grupo

Seleccione un Grupo

Administrador

Casuales

Registrados

Proveedor CIDLIS

Proveedor UIS

Grupos Santoto

Nuevo Guardar

Cuando se selecciona un grupo, en la parte derecha aparece un enlace que tiene como título “Ver usuarios del grupo” (figura 26), al seguir ese enlace el sistema presenta una página donde se pueden apreciar los usuarios por cada grupo (figura 27). Si se hace clic en el botón borrar, se eliminará la información de **login** del usuario, así como sus datos personales.

Figura 26. Acceso directo para ver la información de los Usuarios del Grupo.

● Campos Requeridos

Grupo

Administrador

Ver Usuarios del Grupo

Nombre del Grupo

Administrador

Descripción del Grupo

Nuevo Actualizar Borrar

Figura 27. Ventana del sistema en donde se listan los Grupos de Usuario.

Grupo
Administrador

Usuario	Editar	Borrar
Admin	Editar	<input type="checkbox"/>

Nuevo Borrar

[Volver al menú](#) - [Volver Asignar Usuario a Grupos](#)

Haciendo clic en la opción editar de la ventana que se muestra en la figura 27, el sistema presenta el detalle de la información del usuario (figura 28); a esta ventana también se puede acceder haciendo clic sobre la opción del menú *Administración de usuarios*.

Figura 28. Ventana del sistema en donde se detallan la información personal y de ingreso de los Usuarios.

Seleccione Usuario Admin - Admin

INFORMACIÓN PERSONAL Y DE CONTACTO			
Nombre	Admin	Email	
Dirección		URL	
Teléfono			

INFORMACIÓN DE INGRESO			
Nombre de Usuario	Admin	Confirmar Clave	
Clave			
Grupo de Usuario	Casuales		

Nuevo Actualizar Borrar

Metamodelado

La zona de meta modelado permite al usuario *Administrador* configurar toda la información requerida para presentar el sistema al usuario final.

Figura 29. Ventana del sistema de Administración de Tablas.

Original	Tabla	Acción		
categorias	categorias	Listado	Nuevo	Propiedades
cursos	cursos	Listado	Nuevo	Propiedades
escuelas	escuelas	Listado	Nuevo	Propiedades
estudiante	estudiante	Listado	Nuevo	Propiedades
evaluacion	evaluacion	Listado	Nuevo	Propiedades
materias	materias	Listado	Nuevo	Propiedades
matest	matest	Listado	Nuevo	Propiedades
matprof	matprof	Listado	Nuevo	Propiedades
personas	personas	Listado	Nuevo	Propiedades
planes	planes	Listado	Nuevo	Propiedades
planmat	planmat	Listado	Nuevo	Propiedades
profes	profes	Listado	Nuevo	Propiedades
requisitos	requisitos	Listado	Nuevo	Propiedades

Cuando el usuario hace clic sobre la opción *Administración de Tablas*, el sistema genera la página con un listado con las tablas de la base de datos (figura 29) con la cual se integró la herramienta permitiendo ver su contenido, crear nuevos registros y ver sus propiedades.

Posteriormente, al dar clic sobre la opción *Asignar Nombres amigables a las Tablas*, u obturar sobre el enlace *Propiedades* de la página de administración de tablas, se muestra una página con las características de la tabla (figura 30). En esta página se introduce el nombre amigable, se configuran los campos amigables y se establece, en caso de necesitarse, cuál de los campos se utilizará para aprobar la publicación de un registro entrado por un usuario registrado.

En la parte inferior derecha de la figura 30 se encuentra un enlace denominado *Nombres Amigables para los Campos de Esta Tabla*, si se hace obtura sobre él se llegará a la zona de configuración de nombres amigables de la tabla seleccionada (figura 31). También se puede llegar a esa página haciendo clic sobre la opción del menú denominada *Asignar Nombres Amigables a los Campos*.

Figura 30. Ventana del sistema correspondiente a la configuración de los Nombres Amigables de las Tablas.

Nombre Real de la Tabla: **materias**

Seleccione la Tabla
materias

Nombre Amigable de la Tabla

Tabla Intermedia

Campo Amigable
mat_nom
mat_esc_id
mat_creditos
mat_exigidos

Campo Requerido para publicación

[Agrupación de los campos y Tablas asociadas Nombres Amigables para los Campos Esta Tabla](#)

- Si elige el campo resaltado en Rojo, se Mostrarán en el formulario, los campos de la tabla relacionada para seleccionar los campos de la tabla actual

Actualizar

[Volver a Administración de Tablas](#)

Se pueden introducir los nombres amigables de cada uno de los campos de cada tabla y modificarlos a discreción. Se puede configurar el script a utilizar y seleccionar el nivel de búsqueda en profundidad.

Figura 31. Ventana del sistema correspondiente a la configuración de los Nombres Amigables de los Campos.

Nombre Real de la Tabla: **materias**

Seleccione una Tabla

materias ▼

Nombre Campo	Nombre Amigable	Script	Búsqueda
mat_id		Predeterminado ▼	0 ▼
mat_nom		Predeterminado ▼	0 ▼
mat_esc_id		Predeterminado ▼	0 ▼
mat_creditos		Predeterminado ▼	0 ▼
mat_exigidos		Predeterminado ▼	0 ▼

Actualizar

Si se elige la opción *Autorizar Publicación de Registros* aparecerá una página que despliega una bandeja en donde se listan todos los registros ingresados por los usuarios y pendientes por autorizar.

En la figura 32 se muestra la ventana del sistema que sirve para crear elementos que normalmente requerirían una tabla relacionada, esto se hace con el fin de permitir una administración más sencilla. A esta zona se llega mediante la opción *Administrador de tipos* del menú de *modelado*.

4.6 PRUEBAS Y ENTREGA. En esta última etapa del DRA se prueban todos los componentes del sistema y se deben ejercitar todas las interfases a fondo. Se pueden listar cambios sugeridos obtenidos a través de las pruebas básicas realizadas sobre el sistema durante la implementación del mismo. Se debe aclarar que los errores detectados son directamente proporcionales al tiempo que corresponde a la metodología de desarrollo utilizada para la implantación del sistema. Se requerirán ciertos procedimientos que no serán especificados en este proyecto con el fin de llevar a cabo las mejoras del sistema. Dentro de estos criterios se encuentran: la planificación detallada de las pruebas que efectuarán los usuarios sobre el sistema implementado, la asignación de personal orientado al servicio de evaluación del proceso y personal para el desarrollo de nuevos módulos de acuerdo la naturaleza de la aplicación.

Figura 32. Ventana del sistema correspondiente a la Administración de tipos.

Selección de tabla
personas

Campos
per_tipodoc

Tipos
Cédula de ciudadanía
Tarjeta de Identidad
Pasaporte
Cédula de extranjería

Actualizar Borrar elemento
Quitar asociación

El producto de la planificación de las pruebas será el plan de pruebas basado en el modelo de casos de uso que se presenta en el numeral 4.4.3. Posteriormente, se deberán implementar las pruebas utilizando grandes y variados modelos entidad relación para estudiar el comportamiento del sistema. Finalmente, se deberán evaluar las pruebas para medir la calidad del sistema e incluir las mejoras necesarias.

4.6.1 Meta modelado. En este aspecto se pueden incluir los elementos mostrados en la tabla 6 que no fueron considerados en la especificación de requisitos del presente documento.

4.6.2 Sistema de Anexos. El sistema de anexos, podría incrementar la seguridad de la herramienta al incluir un sistema de búsqueda de virus en el momento de subir los archivos en el servidor, siguiendo los pasos necesarios para evitar la contaminación del servidor y de los clientes que descarguen esos archivos.

4.6.3 validación de datos. Actualmente, las validaciones especializadas para la entrada de datos por parte del usuario están relacionadas a las entradas en los campos Check de

SQLServer, lo cual no es estándar para todos los manejadores de bases de datos y presenta un grado bajo de complejidad en las validaciones que sea necesario realizar. Para ello podría crearse un sistema entrada de expresiones regulares y de algoritmos de validación más complejos, en los que se puedan incluir campos especiales.

Tabla 6. Perspectivas de la herramienta en cuanto a metamodelado.

CAPACIDAD	DESCRIPCIÓN
<i>Guardar el estado de la configuración por cada cliente</i>	Almacenar de manera segura del lado del cliente o del servidor la configuración de los diferentes elementos que visualiza el cliente cuando esta visitando el sitio, esto permite que cuando vuelva, encuentre el sitio de la manera que a él le parece mas atractiva
<i>Agregar ordenamientos</i>	Permitir el ordenamiento de los listados presentados a los usuarios, estos ordenamientos podría ser creados dinámicamente seleccionando de un listado de campos, por cuales se quiere realizar el filtro, así como su tipo de ordenamiento (Ascendente o Descendente)
<i>Agregar un sistema para mostrar los campos que se deseen por parte del cliente</i>	Utilizando la misma filosofía del campo de ordenamiento, presentar al usuario un especio para que configure para cada uno, los campos que se desea que se desliguen como amigables en los diferentes listados del sistema
<i>Posibilidad de duplicar los tipos entre varias tablas</i>	Cuando se crean tipos para las diferentes tablas, y un tipo repite sus mismos elementos debe duplicarse de manera manual todos sus elementos (o haciendo una consulta sobre la base de datos), la idea es permitir que el administrador del sistema pueda duplicar estos tipos y asignarlos a otra tabla con pocas instrucciones
<i>Crear transacciones</i>	Un sistema que permita llevar un paso a paso, utilizando las tablas del modelo, esto es especialmente importante para sistemas empresariales que lleven a cabo complejos procesos para cada una de sus actividades; un ejemplo clásico de esto, es cuando una empresa comercial desea poner un pedido a un proveedor determinado ³⁴ , este proceso involucraría varias tablas y entradas de usuario.
<i>Incluir un lenguaje para ampliar las posibilidades de</i>	Incluir una funcionalidad para poder programar con lenguaje Standard SQL operaciones del negocio del

³⁴ ALBARRACIN, Jaime O. Integración Organizacional Transaccional. Colombia: Ediciones UIS.

consulta	sistema en general, esta herramienta es alta utilidad para la generación de triggers ³⁵ . También se puede complementar con la inclusión de un lenguaje como VBA ³⁶ , para poder crear funciones de cálculo y algoritmos específicos de la aplicación
Poder crear plantillas dinámicas para los formularios	Tener la posibilidad de cambiar dinámicamente las plantillas de presentación de información en modo lectura, o en modo escritura.

4.6.4 Multilinguaje. El sistema puede extenderse para ser manejado en diferentes lenguajes; esto se logra con la creación de un archivo de recursos y manejando los diferentes codificaciones propias de cada lenguaje en la definición de la meta información de cada página. Esta capacidad también depende del lenguaje en el que esté instalado el motor de base de datos.

4.6.5 Capa de datos. Dada la arquitectura de tres capas con la que se desarrolló el sistema, se pueden implementar capas para diferentes motores de datos para extender el sistema a otros manejadores. Esto permitirá adquirir las propiedades o ventajas que cada uno de estos motores tenga respecto al otro; a la vez que lo haría un producto aun más competitivo al poder instalarlo en varios sistemas operativos.

4.6.6 Permisos y usuarios. Se sugiere la creación de permisos orientados a campos y a usuarios para permitir la administración del sistema de manera específica para cada usuario, pero sin deshabilitar la propiedad de administración por grupos que tiene el sistema actual.

El sistema de permisos actual esta orientado a las tablas. Una eventual mejora podría incluir un administrador que permita la visualización de los campos configurados para determinado usuario. Este administrador podría configurar también la información asociada a mostrar para que las diferentes entidades de la tabla pudieran ser usadas para representar varios elementos de información.

³⁵ Manual en línea de SQLServer: palabra clave triggers: Es un procedimiento almacenado que se ejecuta antes y/o después de realizar una operación de actualización en una tabla.

³⁶ Visual Basic para Aplicaciones. Estándar Microsoft.

También puede desarrollarse un módulo que mejore el sistema de asociación de registros para relacionar varios registros en una sola operación.

La creación de un sistema de búsqueda de los elementos existentes cuando se esta asociando la información a registros existentes también puede ser una opción para enriquecer la experiencia del usuario final.

4.6.7 Creación del sistema de enlaces (entre el sitio Web y la aplicación). De manera jerárquica y utilizando el esquema estándar para la generación de enlaces brinda la posibilidad de unir la aplicación dinámicamente con cualquier diseño gráfico, al utilizar diferentes estilos de menú elaborados en Javascript o en Macromedia flash³⁷.

³⁷ Presentaciones multimedia, el esquema fue desarrollado por la empresa Macromedia Inc., Más información en la Web: www.macromedia.com

5. CONCLUSIONES

- El ciclo de vida DRA demostró ser de gran utilidad para la creación de proyectos, ya que presenta los flujos de desarrollo de una manera sencilla. El uso de esta metodología fue posible y apoyó el proceso de manera adecuada debido a que la investigación había sido realizada en el momento del inicio de la implementación de la herramienta, y se contaba con la experiencia previa por parte del autor del proyecto.
- El motor de base de datos SQLServer demostró ser un excelente elección, dado su alto rendimiento, sus ayudas en línea, el manejo de los modelos entidad relación, y el apropiado manejo que tiene de las tablas del catálogo del sistema; por lo anterior, se hizo posible el desarrollo de un sistema de modelado.
- Al utilizar una herramienta como la desarrollada en este proyecto, se obtiene un aumento significativo en la productividad en el desarrollo de sistemas administrativos para sitios con modelos de datos complejos, ya que reduce notablemente el tiempo requerido para la creación y la modificación de los sistemas administrativos de tablas, con respecto a los sistemas que utilizan un desarrollo manual³⁸.
- Se logró diseñar una representación adecuada de los diferentes módulos del sistema gracias a la utilización de algunos elementos del proceso unificado, ya que el ciclo de vida DRA establece la metodología de desarrollo a seguir, pero no establece una forma de representación de los procesos que han de implementarse ni de los usuarios que interactúan con la herramienta.
- El meta modelado probó ser un concepto importante para el desarrollo de aplicaciones que tengan como propiedad la flexibilidad y el cambio en el tiempo, ya que proporciona la posibilidad de modificar elementos sin necesidad de cambios en el modelo de datos o en el código de la aplicación. A su vez, se tiene la posibilidad de

³⁸ Programando cada página independientemente.

adicionar información a los modelos de datos que permitan una presentación amigable al usuario final.

- Se probó que el uso de la técnica desconectada aumentó radicalmente el desempeño de la aplicación, haciéndola más eficiente y productiva en la generación de formularios y listados, ya que reduce el tiempo de procesamiento del lado del motor de base de datos logrando mantener más clientes concurrentes con el mismo recurso.
- El desarrollo en tres capas representó una ayuda importante en la implementación del sistema porque brindó las siguientes ventajas frente otras arquitecturas: mejoró la organización de la aplicación, permitió que las modificaciones y las revisiones al código se realizaran de manera sencilla y que los cambios no afectaran la presentación final del producto.

6. PERSPECTIVAS DE LA HERRAMIENTA

A continuación se presentan las ideas para evolucionar el sistema y convertirlo así en una herramienta más competitiva. Estas sugerencias provienen de la evaluación del diseño realizado y de las pruebas inherentes a todas las etapas del DRA.

Para versiones futuras se hacen las siguientes sugerencias adicionales a las propuestas en el numeral 4.5 respecto a las pruebas realizadas sobre el sistema:

6.1 DESARROLLO ORIENTADO A OBJETOS

Como se mencionó en el numeral 4.3.1, la herramienta ha sido desarrollada de manera procedimental debido a que la tecnología utilizada no soporta el desarrollo de clases. Sin embargo, un desarrollo orientado a objetos permitiría encapsular la lógica del negocio y adicionalmente habría la posibilidad de proteger el código al presentarlo en una estructura ejecutable para ser usado de manera transparente por futuros desarrolladores que utilicen la herramienta como base de implementación de sus aplicaciones Web.

Se sugiere el uso del paradigma orientado a objetos .Net desarrollado por Microsoft, para incluir las características anteriormente mencionadas.

6.2. SEGURIDAD

De acuerdo a los requerimientos expresados respecto a la seguridad de la herramienta, se consideró que el sistema requiere de una configuración de manejo de seguridad adecuado del lado del servidor. Por lo tanto, se sugiere la creación de un esquema de almacenamiento encriptado para los datos sensibles de la base de datos, y para el manejo de información de los usuarios, utilizando algoritmos de encriptación RSA³⁹ o Rijndael⁴⁰,

³⁹ Mas información en la Web: www.rsasecurity.com

⁴⁰ Información adicional: <http://csrc.nist.gov/CryptoToolkit/aes/rijndael>

para que la información de los diferentes elementos de la base de datos estén protegidos, aunque se tenga acceso físico a la base de datos por parte de un atacante.

6.3 HERRAMIENTA EDUCATIVA

El sistema desarrollado podría ser utilizado igualmente como herramienta didáctica en la enseñanza de estructuras de datos, porque orienta efectivamente al estudiante de manera práctica en la representación de la información modelada en una base de datos y las diferentes operaciones que se pueden realizar en ellas. Esto se lograría con una optimización de la interfase de usuario, de manera que guíe al estudiante a través del diseño del modelo de datos relacional que debe adaptar al sistema, donde se le permita familiarizarse con la configuración de los datos y la terminología utilizada para dichos fines.

BIBLIOGRAFIA

ALBARRACIN, Jaime O. Integración Organizacional Transaccional. Colombia: Ediciones UIS.

BJELETICH Sharon y MABLE Greg. Microsoft SQL Server 7 Al Descubierto, la solución más completa. España: Editorial Prentice Hall, 1999.

CÁRCAMO, S. José. Oracle a su alcance. Escuela de Ingeniería de Sistemas e Informática. Colombia: Universidad Industrial de Santander, 2000.

GOMEZ, F. Luis Carlos. Ciclos de vida de desarrollo software. Grupo de Investigación en sistemas y tecnologías de la información –STI-. Colombia: Universidad Industrial de Santander, 1993.

INSTITUTO TECNOLÓGICO DE LA PAZ. Publicación sobre Bases de Datos. México: 2002. Disponible en el sitio Web: <http://www.itlp.edu.mx/publica/tutoriales/basedat1>

JACOBSON, Ivar; RUMBAUGH, James y BOOCH Grady. El Proceso Unificado de Desarrollo de Software. Rational Software Corporation. España: Editorial Addison Wesley autorizado por Pearson Educación S.A, 1999.

KENDALL, Kenneth y KENDALL, Julie. Análisis y diseño de Sistemas. Tercera edición. México: Editorial Prentice Hall, 1997.

MORALES, C. Mario. Teoría General de los Sistemas. Universidad Santiago De Cali, Colombia: 2003. Disponible en el sitio Web: <http://docentes.usaca.edu.co/marmol/teoriagnalsistemas.htm>

MÉNDEZ, D. Néstor Darío. Elementos a tener en cuenta en el diseño e implementación de bases de datos. Facultad de Ciencias y Administración. Universidad Nacional de

Colombia. Manizales: 2003. Disponible en el sitio Web:
<http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060029>

SANCHEZ, P. Javier. El Proceso de software: Metodología de desarrollo software. España: 2002. Facultad de Informática. Universidad de Las Palmas de Gran Canaria
Disponible en el sitio Web:

SENN, James. Análisis y Diseño de Sistemas de Información. Segunda Edición. México: Editorial McGraw Hill, 1992.

WHITTEN, Jeffrey y BENTLEY Lonnie. Análisis y diseño de Sistemas de Información. Tercera edición. España: Ediciones Irwin, 1996.

ANEXO A. CARACTERÍSTICAS DE LOS DATOS DE ENTRADA AL SISTEMA

Restricciones del modelo de datos

Los modelos de datos a crear deben estar en la forma estándar⁴¹ más elevada posible, cada una de las tablas debe tener una llave primaria tipo entero con la característica “**Identity**” activa⁴² (aunque muchos autores no recomiendan esta práctica, es de importancia para el sistema, ya que con una sola llave el sistema puede identificar fácilmente el registro del que se está tratando y se evita la creación de **triggers** para la generación del dato en la llave primaria).

Adicionalmente, solo se pueden utilizar los tipos de datos ínter contruidos⁴³ en SqlServer (no se pueden utilizar: **Binary, Image, sql_variant, timestamp, uniqueidentifier, varbinary**), ya que los tipos de datos generados por el usuario son desconocidos por la aplicación y podrían generar errores.

Para la creación de modelos de datos se debe seguir una nomenclatura para los nombres de los campos de cada una de las tablas, y deben tener como resultado final un nombre único para cada campo en la base de datos. Un ejemplo de este esquema es: si se está creando una tabla con el nombre ciudad, los campos deberían llamarse *ciu_id*, *ciu_dep_id*⁴⁴, *ciu_nombre*; si está creando una tabla llamada departamento entonces se consideraría la nomenclatura *dep_id* ó *dep_nombre*.

Restricciones de los enlaces

Cuando se desee crear un nuevo enlace entre la aplicación Web y la herramienta ó asociar uno ya existente, se ingresa a la opción de enlaces de cada registro. Si se escribe un enlace en el formulario y este ya se encuentra incluido en el sistema, este no realizará

⁴¹ CARCAMO, S. José. Oracle a su alcance. Escuela de Ingeniería de Sistemas e Informática. Colombia: Universidad Industrial de Santander, 2000.

⁴² Es decir, auto numérica.

⁴³ Manual de usuario en línea: data types-SQL Server

⁴⁴ Nomenclatura para llaves foráneas utilizada por la herramienta.

una validación para confirmar este hecho, por lo cual el usuario es responsable de la verificación del estado del enlace deseado; es decir, si existe actualmente en el sistema o no.

Restricciones de los anexos

Al crear los anexos, el sistema automáticamente sube y maneja el nombre físico y traslada los archivos a una ubicación determinada en el archivo ult_const.asp ubicado dentro del directorio config (Anexo B).

Las restricciones que se presentan en cuanto a los archivos anexos son el tamaño destinado para tal propósito en el servidor; el tamaño máximo configurado (está en el archivo ult_const.asp) para subir archivos al servidor. La posibilidad de contenido malicioso en esos archivos (virus, troyanos, gusanos, etc.) que pueden contaminar o, en el peor de los casos, ocasionar daños al servidor. Es de vital importancia que solo usuarios autorizados usen el sistema de anexos.

Restricciones de las consultas en búsqueda especializada

Las búsquedas especializadas, se basan en algoritmos de análisis de consultas parametrizadas; por tal motivo, deben cumplir con un formato específico y un orden en sus parámetros especificado para generar la consulta final adecuadamente para el usuario que consulta.

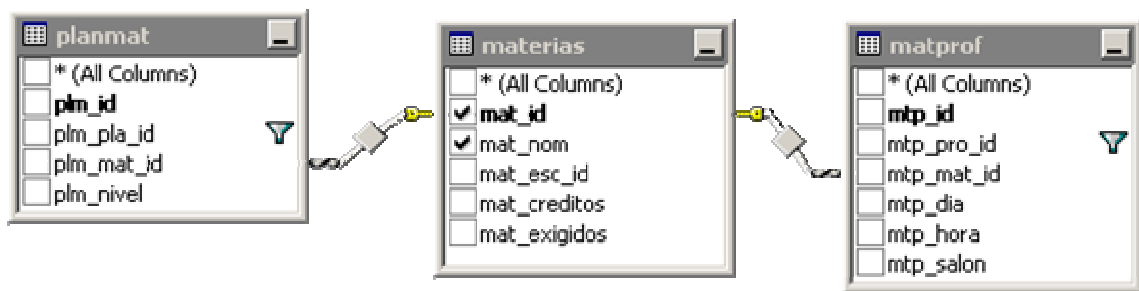
Las consultas pueden ser de dos tipos: Simples, si involucran una sola tabla, o Compuestas, si incluye datos o restricciones de varias tablas.

Un ejemplo de una consulta parametrizada es: `SELECT id AS Id, ciu_nombre AS Texto FROM ciudades WHERE (ciu_dep_id = ¥1¥)`. En donde ¥1¥ representa un parámetro, el cual es ingresado por el usuario a través de la interfase de configuración de parámetros. El parámetro se configura en la tabla ult_param, y contiene información necesaria, según el presente ejemplo, para generar un listado de departamentos. La consulta para la búsqueda especializada debe ser creada usando los ALIAS mostrados en la consulta de ejemplo y los parámetros deben ser congruentes con la información que se desea filtrar,

de otra manera el sistema puede presentar error, o la consulta puede retornar valores inesperados.

Cuando una consulta involucra varias tablas, en ella se incluyen cláusulas SQL “INNER JOIN”, que deben estar en el mismo orden de su restricción en la cláusula WHERE. Para el ejemplo de la base de datos Academia⁴⁵, presenta el siguiente ejemplo de consulta:

```
SELECT  materias.mat_id AS Id, materias.mat_nom AS Texto
FROM    materias INNER JOIN
        planmat ON materias.mat_id = planmat.plm_mat_id INNER JOIN
        matprof ON materias.mat_id = matprof.mtp_mat_id
WHERE   (planmat.plm_pla_id = ¥1¥) AND (matprof.mtp_pro_id = ¥2¥)
```



La anterior consulta permite mostrar las materias que pertenecen a un determinado plan y a un determinado profesor. En el ejemplo se puede observar que el orden de los parámetros, INNER y WHERE mantienen le mismo orden relativo.

⁴⁵ CÁRCAMO, S. José. Oracle a su alcance. Escuela de Ingeniería de Sistemas e Informática. Colombia: Universidad Industrial de Santander, 2000.

ANEXO B. CARACTERÍSTICAS HARDWARE Y SOFTWARE NECESARIAS PARA EL CORRECTO FUNCIONAMIENTO DEL SISTEMA

Especificaciones de Hardware

Existen tres tipos de hardware involucrado para el correcto funcionamiento del sistema en una red Intranet o para su publicación en Internet.

1. Servidor de Aplicaciones y de Base de datos. Estos servidores pueden ubicarse en máquinas independientes, sin embargo, deben presentar una sola unidad lógica.
2. Infraestructura de comunicaciones.
3. Sistemas cliente.

En cuanto a las características del servidor, debe ser una máquina con arquitectura servidor, que cumpla con los siguientes requerimientos:

Requerimiento	Recomendado	Mínimo
<i>Memoria RAM</i>	512Mb	256Mb
<i>Procesador</i>	Doble Procesador a 3Ghz, Cada uno	Un solo Procesador a 1Ghz
<i>Disco Duro⁴⁶</i>	100Mb	20Mb
<i>Tarjeta de Red</i>	Tarjeta de red 10/100/1000	Tarjeta de Red 10/100

Respecto a las comunicaciones se requiere una infraestructura que permita la publicación de sitios Web en la intranet o en Internet.

Del lado del cliente, puede ser un computador con las especificaciones técnicas necesarias para la visualización de páginas Web.

⁴⁶ Depende principalmente del tamaño de la base de datos que se esté manejando y de los anexos. Los archivos de la aplicación de meta modelado ocupan menos de un 1Mb

Especificaciones de Software

Las siguientes son las características del software que debe estar instalado en el servidor del sistema:

Nombre software	Función
<i>Microsoft Windows Server (Cualquier versión mayor o igual a Windows 2000 Server)</i>	Es el sistema operativo que aloja los archivos de la aplicación, maneja los pormenores de la comunicación en redes, usuarios concurrentes y seguridad
<i>IIS, Microsoft Internet Information Server</i>	Este software es el servidor Web que interpreta las secuencias de comandos del lado del servidor ASP
<i>Microsoft SQLServer 2000 o 7 ó MSDE</i>	Este es el motor de base de datos utilizados para crear la aplicación

Para el lado del cliente se recomienda el uso de exploradores que manejen DHTML y provean un motor de javascript lo suficientemente actualizado para poder mostrar los elementos deseados. Sin embargo el sistema degrada de manera grácil en sistemas de exploradores más antiguos. Se recomienda el uso de Microsoft Internet Explorer 6 o superior y/o Netscape 7.0 o Superior

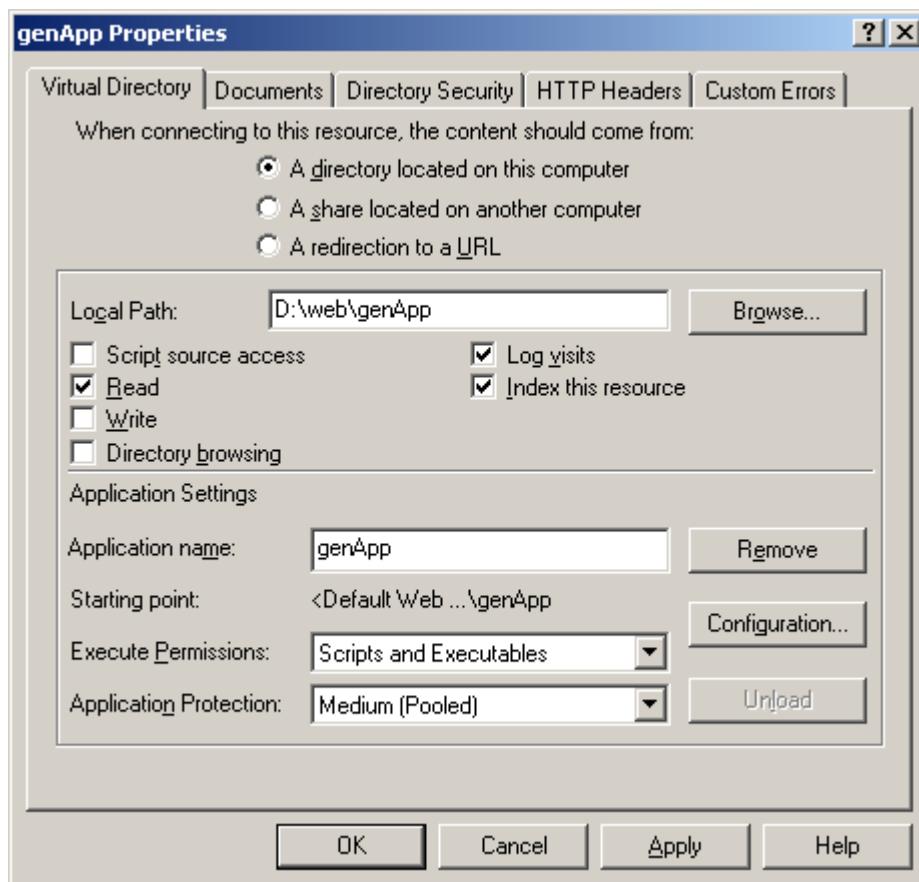
Especificaciones de seguridad

La seguridad debe ser considerada del lado del servidor, es decir, donde están ubicados los archivos ASP del sistema y la base de datos, se debe tener una configuración adecuada para restringir el acceso físico a estos elementos. Se recomienda que SQL Server sea configurado con cuentas de confianza de **Windows NT**, que tenga restricciones de acceso físico al servidor (servidor debe tener clave de acceso y restricciones de acceso físico a él).

El directorio virtual o sitio en el Servidor **Internet Information Server** debe presentar una apropiada configuración; la herramienta debe ser configurada en él como aplicación para

aumentar su seguridad; no debe permitir exploración de directorios, ni acceso a los scripts.

Para modificar estas propiedades el usuario debe ingresar al Panel de Control de Windows, posteriormente seleccionar Herramientas administrativas, Abrir el administrador del Internet Information Server, una vez la consola de administración presente la información del sitio Web en el cual fue instalado la aplicación, el usuario debe obturar el botón derecho del Mouse y posteriormente seleccionar propiedades. Lo cual hará que el sistema despliegue la ventana presentada en el siguiente gráfico de ejemplo.



Como recomendación especial, el sistema no se debe instalar en el directorio por defecto de IIS ya que este es el primer lugar que un atacante busca, es mejor ubicarlo en otra carpeta en el sistema de archivos del servidor.









ANEXO C. DESCRIPCIÓN DE LA CONFIGURACIÓN DE LOS DIRECTORIOS DEL SISTEMA

















Páginas y archivos

Dentro de los archivos de la herramienta se puede encontrar la siguiente estructura de directorios:



Dentro del **Directorio admin** se encuentran los siguientes archivos:

 login.asp	Página para entrada al sistema, Clave, Nombre de Usuario y nombre del grupo de usuarios al que pertenece
 modifdatosadmin.asp	Modificar los datos del administrador
 ult_ami.asp	Formularios para administrar la información amigable de cada una de las tablas del sistema
 ult_ami_aud.asp	Página para almacenar los datos introducidos en ult_ami.asp
 ult_auth.asp	Página que se encarga de autenticar el usuario cuando entra su usuario, contraseña y grupo de usuario
 ult_auth_err.asp	Página que muestra al usuario un error en caso de que los datos de entrada al sistema sean incorrectos
 ult_autori.asp	Página que muestra la bandeja ara autorizar los registros entrados por usuarios casuales
 ult_cgrp.asp	Página para entrar los datos para los grupos de campos y de información asociada


 ult_cgrp_aud.asp	Página que actualiza la información de los grupos entrada en ult_cgrp.asp
 ult_grp.asp	Página para administrar los grupos de usuarios del sistema
 ult_grp_aud.asp	Página para actualizar los datos de los grupos de usuarios entrados en ult_grp.asp
 ult_per.asp	Página para administrar los permisos por grupo de usuarios y tablas
 ult_per_aud.asp	Página para actualizar la información entrada en ult_per.asp
 ult_secur.asp	Página que maneja la seguridad en el sistema, se incluye dentro de cada página, para determinar el nivel de acceso de la misma.
 ult_tab_prop.asp	Lista las tablas del sistema con sus nombre amigables y presenta (dentro de la zona administrativa) enlaces para cada tabla en los cuales se pueden editar las propiedades y datos de cada tabla
 ult_tbl_tprop.asp	Página para asignar meta información a cada tabla del sistema
 ult_tbl_tprop_aud.asp	Página que actualiza la información entrada en ult_tbl_tprop.asp
 ult_tipos.asp	Página para la administración de los tipos de las tablas.
 ult_tipos_aud.asp	Página para actualizar la información entrada en ult_tipos.asp
 ult_updEngine.asp	Página que actualiza la meta información en la tabla ult_estruc
 ult_usr.asp	Página para asignar usuarios a grupos (Cuando se han creado y se quiere modificar el grupo al cual pertenece)
 ult_usr_aud.asp	Página para actualizar la información del grupo del usuario seleccionado en la base de datos
 ult_usradm.asp	Página para la administración de usuarios del sistema
 ult_usradm_aud.asp	Página para actualizar la información de los usuarios en

la base de datos

 **ult_usr_l.asp** Listado de usuarios por grupo

Dentro del directorio “anexos”, se almacenan los archivos que asociados a cada registro (los que usen esta característica). Estos archivos tienen un nombre código para evitar la repetición del mismo y sobrescribir los datos cuando se agregue un nuevo anexo.


En el **directorio config**, se encuentra un archivo que contiene las constantes que configuran el sistema, tales como caminos y número de registros por página.


 **ult_const.asp** Contiene información general de configuración del sistema


Directorio Css: este directorio contiene los archivos de estilo en cascada⁴⁷ utilizados para configurar la apariencia general del sistema

 **s2003.css** Contiene los estilos en cascada del sistema

En el **directorio dB**, están los archivos relacionados al manejo directo de base de datos y los que contienen información específica para el manejo de las diferentes excepciones para los tipos de datos SQLServer.

 **adovbs.inc** Contiene las constantes para el manejo de datos utilizando ADO

 **conn.asp** En este archivo se configura la cadena de conexión del sistema









 **datat.asp** Contiene las excepciones para el manejo de los tipos de datos dentro de todo el sistema (Archivo clave)

⁴⁷ Css: Cascade Style Sheet

El **directorio images** contiene las imágenes usadas a lo largo del sistema.

El **Directorio Install**, contiene un Script SQL, para ser ejecutado por el Motor SQL Server para instalar el sistema en la base de datos a administrar

En el **directorio Lang** se incluyen archivos para el manejo de múltiples lenguajes del sistema, estas es una de las características a desarrollar en futuras versiones. En el directorio scripts se encuentran diferentes archivos javascript, para mejorar la experiencia del lado cliente del sistema.

 <i>AnchorPosition.js</i>	Determina la posición de un enlace en la pantalla cliente, es usado para desplegar el script de fechas
 <i>CalendarPopup.js</i>	Es utilizado para manejar los por menores del calendario desplegado para los tipos de datos datetime
 <i>date.js</i>	Maneja validaciones para las fechas usadas en el sistema, las cuales son manejadas en su totalidad con el estándar ISO 8601 ⁴⁸
 <i>PopupWindow.js</i>	Permite manejar ventanas emergentes para el manejo de fechas
 <i>tooltip.js</i>	Es utilizado para desplegar cuadros de información adicional cuando se esta en la modalidad de solo texto.
 <i>ult_date.js</i>	Manejos de fechas especiales
 <i>ult_functions.js</i>	Funciones especiales: Cambios de página, ventanas emergentes.
 <i>ult_val.js</i>	Algoritmos de validación usados a los diferentes formularios del sistema

⁴⁸ Ayuda en Línea de SQLServer

ANEXO D. CARACTERÍSTICAS DE LAS CONSULTAS REALIZADAS POR EL SISTEMA.

La tabla que se muestra a continuación contiene un resumen de las consultas que utiliza el sistema con una breve descripción de la utilidad que representa cada una de ellas.

NOMBRE CONSULTA	UTILIDAD
<i>Generación de validaciones</i>	
<i>ult_Checks</i>	Consulta del sistema: CHECK_CONSTRAINTS, la cual devuelve los valores de las validaciones creadas en las diferentes tablas del sistema. Base para ult_checks2
<i>ult_checks2</i>	Asocia las cláusulas de Chequeo con las columnas de meta modelado, lo cual permite generar el código de validación de entradas del usuario final en el cliente
<i>ult_validation</i>	Devuelve el texto del Texto de Chequeo
<i>Generación de Formularios y Reportes</i>	
<i>ult_Cols</i>	Consulta del sistema: COLUMNS, devuelve las propiedades de cada columna, para cada tabla en el sistema. Base para ult_Cols
<i>ult_col_ami</i>	Asocia las tablas del sistema con la información del meta modelado
<i>ult_PrimaryKeys</i>	Consulta del sistema: KEY_COLUMN_USAGE, Muestra todas las llaves primarias de las tablas incluidas en el sistema. Sirve como base para ult_vw_estruc
<i>ult_tabls</i>	Trae la información del sistema de las tablas, la asocia con la información de meta modelado y con la de permisos. Sirve como base para ult_vw_estruc

ult_vw_estruc	Contiene el resumen de toda la información de las columnas del sistema, su información de modelado y de permisos, es la consulta mas importante para la generación de formularios y Reportes
Información Asociada	
ult_constraint	Consulta del sistema: CONSTRAINT_COLUMN_USAGE Trae todos los Constraints para columnas creados en el sistema, los cuales representan llaves primarias y llaves foráneas y primarias. Base para
ult_ForeignKeys	Consulta sobre las tablas del catálogo del sistema, permite conocer las propiedades de las llaves foráneas, para usarlas posteriormente en la generación del acceso a la información asociada.
ult_inter_plusinfo	Determina cuando una tabla intermedia solo funciona como enlace entre dos tablas o cuando tiene mas información en ella, esto sirve para facilitar el proceso de asociación de datos
ult_vw_inter_tbl	Sirve para determinar las relaciones de cada tabla (tablas intermedias). Esta consulta tiene una versión desconectada en la tabla ult_inter_tbl.
Búsqueda en profundidad	
ult_deep_search	Determina cuales campos pueden tener la propiedad, búsqueda en profundidad, estos campos son llaves foráneas que a su vez tienen relaciones propias
Búsqueda con la Metodología Full-Text Search (Búsqueda heurística)⁴⁹	
ult_fts	Lista las tablas que tienen la propiedad de búsqueda Full-Text Activa
ult_fts_tables	Determina los permisos para las tablas que tienen la propiedad

⁴⁹ Servicio proporcionado por SQLServer




	Full-Text, combinada con la anterior se genera un listado de las tablas sobre las cuales se pueden efectuar las búsquedas
Permisos	
<i>ult_grp</i>	Lista los grupos de usuario del sistema
<i>ult_per_tbl</i>	Devuelve los permisos sobre cada tabla para perfil de usuario

ANEXO E. GENERACION DE ENLACES PARA LA PUBLICACIÓN FINAL DE LA APLICACIÓN WEB

Cuando el *Administrador* del sistema desea crear un acceso para los usuarios de la aplicación Web (*Cientes*), se deben crear los diferentes enlaces para acceder a la información almacenada en la base de datos. Para esto se debe seguir una metodología específica y pasar parámetros a la página de reportes.

Existen dos casos, y por lo tanto esquemas de parámetros que se deben crear para los enlaces, el primero, se refiere a filtros sobre una sola tabla y el otro cuando involucra más de una tabla.

Los esquemas comparten los siguientes parámetros, los cuales presentan información en forma jerárquica que ubica al usuario en el punto en el que se encuentra dentro del sitio:

a= Nivel 1	 Nivel 1
b= Nivel 2	 Nivel 2
c= Nivel 3	 Nivel 3

A continuación se describen los parámetros necesarios para los dos casos mencionados anteriormente; de acuerdo a la cuidadosa configuración de estos parámetros se obtendrán la verdadera funcionalidad del sistema que se describen en la especificación de requerimientos del documento de la propuesta de grado.

1. Para una sola Tabla

PARÁMETRO	DESCRIPCIÓN
Tbl	Tabla sobre la cual se quiere hacer el filtrado
Ami	Nombre amigable adicional (si deseo mostrar al usuario un título diferente al nombre amigable de la tabla)
ult_par1	Nombre del campo sobre el cual se desea hacer el filtrado

ult_parv1	(pueden ser varios campos siguiendo la nomenclatura ult_par1, ult_par2, etc.) Valor del campo sobre el cual se desea filtrar
Menuc	Le indica a la página de listados que el llamado fue realizado desde un menú

2. Para varias Tablas

PARÁMETRO	DESCRIPCIÓN
Tbl	Tabla base sobre la cual se hace el filtrado y de la cual se listarán sus registros
Ami	Nombre amigable adicional
ult_int_tbl	Nombre de la tabla intermedia
ult_org_org	Llave primaria de la tabla asociada
ult_int_forg	Llave foránea de la tabla asociada en la tabla intermedia
ult_int_foli	Llave foránea de la tabla base en la tabla intermedia
ult_org_tbl	Tabla asociada.
ult_par1	Parámetro sobre el cual se desea filtrar. Este parámetro puede ser sobre cualquiera de las tres tablas involucradas y pueden ser varios parámetros siguiendo la nomenclatura: ult_par1, ult_par2, etc.
ult_parv1	Es el valor respectivo de cada parámetro ult_par1, ult_par2
Menuc	Indica a la página de listados que la llamada fue efectuada desde un enlace.

ANEXO F. DETALLES DE LA INTEGRACION DE LA HERRAMIENTA CON LA BASE DE DATOS

A continuación se presentan los pasos que debe seguir el actor Administrador para integrar la base de datos con la herramienta y posteriormente realizar el metamodelado de la información.

Pasos de Instalación

1. Desarrollar un modelo de datos que cumpla con las propiedades mencionadas en el Anexo A.
2. Ejecutar el script de instalación del sistema. Este Script SQL crea las tablas, vistas y procedimientos almacenados para el funcionamiento del sistema en general. También crea el usuario administrador, el cual es base para el manejo y configuración final de la aplicación (también crea el grupo casual). Este script se ejecuta en el Analizador de consultas de SqlServer o en una aplicación que llame el motor MSDE de SQLServer.
3. Copiar los archivos de la aplicación a la carpeta destinada para el sitio Web que se desea crear, debe conservar la misma estructura de directorios que la documentada en el Anexo B.
4. En el Internet Information Server⁵⁰, crear un directorio virtual para el manejo de la aplicación. Los pasos para crear un directorio virtual para Windows 2000 son los siguientes:
 - Inicio
 - Panel de Control
 - Herramientas Administrativas
 - Servicios de Internet Information
 - Despliega la carpeta del servidor local

50

<http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/iis/iis5/reskit/iis50rg/iischp1.asp>

- Sitios Web: En el sitio Web por defecto hace clic en el botón derecho Selecciona nuevo, nuevo directorio virtual. A continuación aparece un ayudante que indica los pasos a seguir.
 - Siguiente
 - Escribe el Alias del sitio, por ejemplo GenApp
 - Selecciona el directorio físico en donde se copiaron los archivos de la aplicación
 - Siguiente
 - Configura los permisos de la aplicación se recomienda otorgar solo los permisos de Lectura y Ejecución de Scripts solamente.
 - Siguiente
 - Finalizar
5. Modificar el archivo conn.asp, ubicado en el directorio dB de la estructura de directorios, en este archivo se configura la cadena de conexión⁵¹ que permitirá a la aplicación comunicarse con la base de datos que se esta configurando. Se debe asignar el valor de la cadena a la variable ProvStr ubicada la línea 13 de ese archivo.
 6. Abrir el archivo ult_const.asp y configurar los caminos en los que se desea almacenar los archivos anexos y el camino de directorio virtual que debe seguir para observar el contenido del archivo. Para esto debe modificar las líneas 8 y 9 respectivamente, al asignar a las variables ult_sav_path y ult_viw_path los parámetros descritos.

Pasos de configuración

Una vez instalada la aplicación en el servidor local, se debe ingresar a la zona administrativa del sistema:

1. El usuario abre un explorador de Internet (**Internet Explorer, Netscape** o cualquier **browser** reciente) y entra la siguiente direcciones (se asume que se esta trabajando sobre el servidor sobre el cual se está realizando la instalación) <http://localhost/genApp/admin/login.asp>. Allí aparecerá un formulario para que ingrese su nombre de usuario, clave y grupo al que pertenece. Solo el usuario administrativo puede acceder a la zona de administración de permisos y modelado. El nombre de

⁵¹ http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ado270/htm/pg_ado_get_data_4.asp

usuario creado en la instalación es admin. Y la clave es admin. Una vez ingrese a la zona administrativa, este usuario debe cambiar su nombre de usuario y clave a uno de su conveniencia.

2. El siguiente paso dentro del proceso de configuración, es la creación de los grupos de usuarios (aunque los grupos se pueden crear mas adelante durante el funcionamiento del sistema).
3. A continuación, se otorgan los permisos sobre los diferentes grupos de usuarios creados.
4. Posteriormente se crean los usuarios para cada grupo, este proceso puede realizarse en cualquier momento, después de paso 3.
5. A continuación se inicia el proceso de configuración de modelado. Dentro del submenú se ingresa a la opción Administración de Tablas, se selecciona cada una de las tablas y siguen los pasos descritos a continuación por cada tabla:
 - El primer paso es ingresar a la configuración de la tabla, allí se entra un nombre amigable para la misma. Se configura el parámetro tabla intermedia, el cual proporciona información al sistema para tratar a estas tablas de manera especial. Selecciona uno o mas campos amigables del listado (si no selecciona ningún campo amigable el sistema asumirá el segundo campo de cada tabla). Y selecciona el valor para el parámetro Campo requerido para publicación.
 - Posteriormente se procede a configurar los valores amigables de los campos, para realizar esa operación se hace clic en el enlace ubicado en la parte inferior derecha de la página marcado como Nombres Amigables para los Campos Esta Tabla. Allí podrá escribir nombres amigables para los campos que se crearon en el modelo de datos original. Asimismo, puede seleccionar un script especial que desee utilizar sobre el campo, las opciones son. Predeterminado (usa los scripts por defecto diseñados para cada tipo de datos), **Attach** (denota la inclusión de un archivo para ese campo), debe tener cuidado de ser congruente con los scripts que selecciona, ya que seleccionar un tipo **Attach** para un tipo entero, por ejemplo, puede resultar en errores del sistema. En esta zona se procede a configurar la búsqueda en profundidad, y como en el caso del script, este valor solo puede seleccionarse para llaves foráneas que cumplan las condiciones para usar la característica de búsqueda en profundidad. En cuanto a los nombres

amigables para los campos tipo bit: si se introduce un nombre amigable normal, el formulario aparecerá un cuadro check,

activo

Pero se puede configurar un valor adicional para darle al usuario una mejor experiencia, por ejemplo si se quiere configurar el campo tipo de estudiante y existente solo dos valores posibles; Regular y condicional, el nombre amigable se puede configurar de la siguiente manera: Estado del estudiante ¥Activo¥Condicional¥ y se obtiene el siguiente resultado cuando se genera el formulario.

Estado del estudiante
 Activo Condicional

Una configuración adicional que se puede realizar es la de tipos, esta característica no convencional de los modelos entidad relación, permite ahorrar la creación de tablas en exceso, lo cual aumenta la complejidad del modelo de datos, disminuye el desempeño general de la aplicación y dificulta la administración del sistema. Dadas las anteriores razones es de gran utilidad poder contar con sistema que permita asociar datos semi-dinámicos asignados por el administrador del sistema a determinadas tablas, los tipos de datos que pueden tener tipos son los int (Enteros), los cuales son asociados programáticamente con la tabla ult_tipos. Para crear un tipo de datos para un campo, se ingresa a la opción administración de tipos ubicada en el menú administrativo y en el submenú modelado. Una vez carga la página se selecciona la tabla sobre la cual se desea trabajar, se muestran en el recuadro ubicado al lado derecho de la página un listado de los posibles campos a asignarles tipos, es decir, aquellos que tenga como tipo de datos, int. En la parte izquierda se presenta un cuadro de lista, en donde se muestran los tipos actuales del campo (si los tiene), un campo de texto, un botón con el texto Borrar Elemento y Otro botón con el texto Quitar asociación.

Cuando se selecciona un elemento del cuadro de lista, se pone automáticamente en el cuadro de texto, lo cual permite editar el texto del tipo y posteriormente grabar el cambio utilizando el botón actualizar.

Una vez realizada la configuración anterior se puede modificar los elementos del modelo de datos, los permisos y las entradas del metamodelado, lo cual proporciona gran flexibilidad en el sistema final.

Al tener la anterior configuración realizada el último paso a seguir es crear una plantilla con el diseño gráfico deseado, esta plantilla puede tener esquema de colores y diagramación, solo se debe dejar un espacio para que los formularios y reportes generados por el sistema se ubiquen en lugar visible y con espacio suficiente para ser observado por los usuarios finales.

Esta plantilla es totalmente independiente de la aplicación y es a gusto del cliente final de la aplicación por cual, no se dan recomendaciones para su creación. Dentro de esta plantilla se debe ubicar un espacio para los diferentes enlaces siguiendo la metodología de creación de enlaces descrita en el Anexo E.

ANEXO G. METODOLOGÍA PARA LA CREACIÓN DE BÚSQUEDAS FULL TEXT- SEARCH

Las búsquedas full-text proporcionan un mecanismo para buscar contenido en tablas que tienen gran cantidad de registros y de información.

El sistema tiene un manejo para este tipo de búsquedas utilizando las consultas `ult_fts` y `ult_fts_tables`. Estas consultas devuelven cuales tablas tienen índice de búsqueda full-text creado, dicha tablas son listadas y el sistema presenta un cuadro de texto para que el usuario ingrese el texto que desea buscar. Como resultado de la búsqueda se resaltan las palabras clave entradas por parte del usuario.

Para utilizar el servicio full-text se deben cumplir con los siguientes requisitos:

- Tener el servicio Index-Server de Windows instalado en el equipo en donde se instale la aplicación
- Tener instalado el servicio Full-text de SQLServer para realizar las búsquedas sobre las tablas dadas.
- SQLServer versión 2000 o superior

Los pasos para crear índices full-text pueden ser consultados en la ayuda en línea de SQLServer, palabra clave del índice de la ayuda: full-text, (See also full-text indexes).

La búsqueda full-text la puede realizar cualquier actor, cuando sale la página de búsqueda full-text se presentan las tablas que tienen activa la propiedad y permite seleccionar uno o varios criterios por lo cual se desea buscar. Se presiona el botón de Buscar y se le presentan al usuario final un listado con los textos encontrados y con la ubicación del mismo.

BUSCADOR

Buscar

Todos los Criterios

Materiales Estudiante

Escuelas

Una extensión de la búsqueda full text son las búsquedas especializadas. El sistema realiza el siguiente procedimiento para el desarrollo de este tipo de consulta.

- En la tabla `ult_tqueries` se crean las consultas parametrizadas.
- En la tabla `ult_param` se establecen los valores para los parámetros de cada una de las consultas.
- Posteriormente se utiliza la página `ult_prep_busqueda.asp`, y se le envía en el parámetro *consulta* el(las) consultas que se van a mostrar al cliente.

El cliente podrá ver los parámetros de cada una de las consultas para seleccionar los datos por los cuales se podrá buscar. Estos parámetros son congruentes con los datos ingresados en la tabla `ult_param`, cuando se selecciona una consulta, automáticamente el sistema genera un listado con los valores posibles.

Una vez seleccionados los valores de los parámetros, ejecuta las consultas y muestra un listado con los valores que cumplen con los criterios elegidos. Este listado presenta un enlace para mostrar los detalles del registro que desea seleccionar.

Editar	Consulta	Seleccionar
<input checked="" type="radio"/>	Estudiantes por Materia	<input checked="" type="checkbox"/>
<input type="radio"/>	Profesor por escuela	<input type="checkbox"/>

Cobertura Geográfica