



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

“ESTUDIO DE AGENTES EN SERVICIOS WEB PARA LA REPLICACIÓN

DE BASES DE DATOS HETEROGENEAS CASO PRÁCTICO:

APLICACIÓN SATÉLITE CONTILIBRO”

TESIS DE GRADO

Previa a la obtención del título de

INGENIERO EN SISTEMAS INFORMÁTICOS

Presentado por:

ÁNGEL EDUARDO APINA REINO

RICHARD FERNANDO ROBLES CARRILLO

RIOBAMBA – ECUADOR

2011

AGRADECIMIENTO

Agradezco a Dios por haberme dado la perseverancia necesaria para poder culminar este proyecto, a mi familia por ese apoyo incondicional que me han brindado desde la niñez, a mi gran amor Mirian Riera y a mis amigos quienes de una u otra forma nos han encaminado en el desarrollo de la tesis así como a los Ingenieros Wladimir Castro, Ivan Menes y Jorge Huilca que fueron piezas fundamentales para la culminación satisfactoria de esta investigación.

Ángel Apina Reino

Mi agradecimiento a Dios, a mi familia y amigos, que han sido el eje principal para la conclusión de este trabajo de investigación, en especial quiero dar gracias a mis padres y hermanas, quienes a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento.

Al Ing. Wladimir Castro quien nos orientó durante el desarrollo del trabajo y a mi compañero de tesis por su apoyo incondicional.

Richard Robles Carrillo

DEDICATORIA

El presente trabajo está dedicado a Dios por haberme dado la vida, a mis padres José Luis y Anita Cecilia que han sido el ejemplo de superación a pesar de las adversidades y obstáculos que la vida presenta. A mis hermanos José Luis, Eliana y Johanna por el constante apoyo que me permitieron llegar a esta anhelada meta.

Ángel Apina Reino

A mis padres y hermanas por todo el apoyo y comprensión durante el desarrollo de la tesis, quienes aportaron de una u otra manera con sus consejos y buenos deseos para la conclusión de este trabajo de investigación dándome ejemplos dignos de superación y entrega.

Richard Robles Carrillo

FIRMAS RESPONSABLES Y NOTAS

ING. IVAN MENES

**DECANO DE LA FACULTAD DE
INFORMÁTICA Y ELECTRÓNICA**

ING. RAUL ROSERO

**DIRECTOR DE LA ESCUELA
DE INGENIERÍA EN SISTEMAS**

ING. WLADIMIR CASTRO

DIRECTOR TESIS

ING. IVAN MENES

MIEMBRO DE TESIS

LCDO. CARLOS RODRIGUEZ

**DIRECTOR DEL CENTRO
DE DOCUMENTACIÓN**

“Nosotros, Ángel Eduardo Apina Reino y Richard Fernando Robles Carrillo, somos los responsables del contenido, ideas y resultados planteados en el presente proyecto de tesis, y el patrimonio intelectual del mismo pertenece a la Escuela Superior Politécnica Chimborazo”.

Ángel Eduardo Apina Reino

Richard Fernando Robles Carrillo

ABREVIATURAS

AES: Advanced Encryption Standard.

API: Interfaz de Programas de Aplicación.

ASP: Active Server Pages (Páginas de Servidor Activas)

CORBA: Common Object Request Broker Architecture

HTML: HyperText Markup Language (lenguaje de marcado de hipertexto)

HTTP: Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto)

ISP: Proveedor de Servicios de Internet

J2EE: JavaTM 2 Platform, Enterprise Edition.

OLE DB: Object Linking and Embedding for Databases ("Enlace e incrustación de objetos para bases de datos)

RPC: Remote Procedure Call, Llamada a Procedimiento Remoto

SOAP: Simple Object Access Protocol

SQL: Lenguaje de Consulta estructurado.

SW: Servicios Web

UDDI: Universal Description, Discovery and Integration

URI: Identificador Uniforme de Recurso.

URL: Localizador de Recurso Uniforme.

WSDL: Web Services Description Language

W3C: World Wide Web Consortium

XML: Lenguaje de Etiquetado Extensible

INDICE

CAPÍTULO I MARCO PROPOSITIVO.....	17
1.1 ANTECEDENTES.....	17
1.2 JUSTIFICACIÓN DEL ANTEPROYECTO DE TESIS.....	18
1.2.1 JUSTIFICACIÓN TEÓRICA.....	19
1.2.2 JUSTIFICACIÓN APLICATIVA.....	21
1.3 OBJETIVOS.....	22
1.3.1 OBJETIVO GENERAL:.....	22
1.3.2 OBJETIVOS ESPECÍFICOS:.....	22
1.4 HIPÓTESIS.....	23
CAPÍTULO II MARCO TEÓRICO TECNOLOGIAS PARA LA DISTRIBUCION Y ACCESO A DATOS.....	24
2.1 REPLICACION DE DATOS CON SQL SERVER.....	24
2.1.1 INTRODUCCIÓN A LA REPLICACIÓN DE DATOS EN SQL SERVER.....	24
2.1.2 COMPONENTES DEL MODELO DE REPLICACIÓN.....	26
2.1.2.1 Publicador.....	26
2.1.2.2 Distribuidor.....	27
2.1.2.3 Suscriptor.....	27
2.1.2.4 Publicación.....	28
2.1.2.5 Suscripción.....	28
2.1.3 Agentes de Replica.....	30
2.1.3.1 Agente de instantáneas.....	30
2.1.3.2 Agente de Registro de LOG.....	31
2.1.3.3 Agente de distribución.....	31

2.1.3.4	Agente de mezcla.....	32
2.1.3.5	Agente de lectura de cola.....	33
2.1.4	TIPOS DE REPLICACIÓN.....	33
2.1.4.1	Replicación de instantáneas.....	33
2.1.4.2	Replicación Transaccional.....	39
2.1.4.3	Replicación de mezcla.....	48
2.2	SERVICIOS WEB.....	61
2.2.1	<i>Introducción a los Servicios Web.....</i>	<i>61</i>
2.2.2	<i>Tecnologías base de servicios Web.....</i>	<i>65</i>
2.2.3	<i>Descripción de las tecnologías de los Servicios Web.....</i>	<i>69</i>
2.2.3.1	SOAP (Simple Object Access Protocol):.....	69
2.2.3.2	WSDL (Web Services Description Language).....	75
2.2.3.3	UDDI (Universal Description Discovery and Integration).....	86
2.3	ADO.NET COMO TECNOLOGIA PARA EL ACCESO A DATOS.....	90
2.3.1	ARQUITECTURA DE ADO .NET.....	91
2.3.2	CONJUNTO DE DATOS.....	93
2.3.2.1	DATATABLECOLLECTION.....	95
2.3.2.2	DATARELATIONCOLLECTION.....	95
2.3.2.3	EXTENDEDPROPERTIES.....	96
2.3.2.4	SERIALIZACION Y DESERIALIZACION DEL DATASET.....	96
2.3.3	PROVEEDORES DE DATOS.....	98
CAPÍTULO III MODELO DE SERVICIOS WEB PARA LA REPLICACIÓN DE DATOS.....		99
3.1	SERVICIO WEB GENERADOR DE SENTENCIAS SQL.....	100

3.1.1 Estructura de la Base de Datos.....	100
3.1.2 Configuración de la Base de Datos.....	101
3.2 SERVICIO WEB PARA LA DISTRIBUCIÓN DE LA INFORMACIÓN.....	104
3.3 SERVICIO WEB PARA LA ADMINISTRACIÓN DEL CATALOGO DE REPLICACIÓN.....	105
3.4 SUSCRIPCIÓN POR EXTRACCIÓN	109
3.5 CONSOLA DE ADMINISTRACIÓN DE REPLICA.....	112
3.6 DEMOSTRACION DE LA HIPÓTESIS.....	116
<i>Técnica de comprobación de la hipótesis:</i>	116
<i>Pruebas Realizadas para la comprobación de la hipótesis:</i>	117
<i>Población:</i>	117
<i>Método de Muestra:</i>	118
<i>Muestra:</i>	118
<i>Fórmula para población finita:</i>	119
<i>Fórmula para población infinita:</i>	119
<i>Pruebas:</i>	120
<i>Procedimiento de análisis de resultados:</i>	127
CAPITULO IV DESARROLLO DE LA HERRAMIENTA DE SINCRONIZACIÓN.....	130
4.1 INTRODUCCIÓN.....	130
4.2 PLANIFICACIÓN Y ANÁLISIS: DOCUMENTO SRS.....	131
4.3 VISIÓN DEL PROYECTO.....	132
4.4 PERFILES DE USUARIOS.....	133
4.5 ÁMBITO DEL PROYECTO.....	133

4.5.1 Alcance.....	133
4.5.2 Requerimientos Generales.....	133
4.5.3 Entrevista.....	134
4.5.4 Requerimientos.....	134
4.5.5 Concepto de la Solución.....	135
4.5.6 Uso de Herramientas.....	135
4.5.7 Planteamiento de la Arquitectura de la Aplicación.....	136
4.5.8 Objetivos del Proyecto.....	137
4.6 FACTORES CRÍTICOS.....	137
4.6.1 Análisis de Riesgos.....	137
4.6.2 Listado de Riesgos.....	138
4.6.3 Tabla de Valores.....	138
4.6.4 Análisis de Riesgos.....	138
4.6.5 Resultados.....	139
4.6.6 Restricciones.....	140
4.6.7 Planificación Inicial.....	140
4.7 PLANIFICACIÓN DE ACTIVIDADES.....	141
4.7.1 Planificación.....	142
4.8 CASOS DE USO Y ESCENARIOS.....	145
4.9 GLOSARIO Y TÉRMINOS.....	150
4.10 DISEÑO LÓGICO.....	152
4.10.1 Diagrama de Secuencia.....	153

4.10.2 Diagrama de Componentes.....	155
4.11 DESARROLLO.....	156
4.12 DISEÑO DE INTERFACES DE USUARIO.....	157
4.12.1 Implementación del Acceso a Datos.....	159
4.12.2 Diseño de Esquemas.....	162
4.13 ESPECIFICACIONES DE SEGURIDAD.....	172
4.13.1 Estabilización.....	172
4.14 INSTALACIÓN.....	173
4.14.1 Procedimientos y Procesos.....	173
4.14.2 Historial del Soporte Técnico.....	174
COCLUSIONES	
RECOMENDACIONES	
RESUMEN	
SUMARY	
ANEXOS	
GLOASARIO	
BIBLIOGRAFIA	

INDICE DE ILUSTRACIONES

FIGURA II.1 COMPONENTES DE REPLICA.....	26
FIGURA II. 2 DUPLICACIÓN DE INSTANTÁNEAS.....	35
FIGURA II.3 DUPLICACIÓN TRANSACCIONAL.....	41
FIGURA II.4 DUPLICACIÓN DE MEZCLA.....	50
FIGURA II.5 ESTÁNDARES PARA SERVICIOS WEB.....	69
FIGURA II.6 ANATOMÍA DE UN MENSAJE SOAP.....	73
FIGURA II.7 EJEMPLO DE UN MENSAJE SOAP.....	75
FIGURA II.8 ESQUEMA DE UN MENSAJE WSDL.....	77
FIGURA II.9 LOCALIZACIÓN DEL SERVICIO WEB.....	86
FIGURA II.10 DATOS DEL REGISTRO UDDI.....	87
FIGURA II. 11. COMPONENTES ADO.NET.....	91
FIGURA II. 12. ARQUITECTURA DE ADO.NET.....	92
FIGURA II. 13. MODELO DE OBJETO DATASET.....	94
FIGURA II. 14. FUNCIONAMIENTO DE LA SERIALIZACIÓN.....	97
FIGURA III.15. CAMPOS AGREGADOS PARA EL CONTROL DE CAMBIOS.....	102
FIGURA III.16. DIRECTORIOS CREADOS EN EL FTP PERTENECIENTES A LOS SERVIDORES CONFIGURADOS.....	105
FIGURA III.17. ESTRUCTURA DEL CATALOGO DE REPLICACIÓN.....	106
FIGURA III.18. ARCHIVOS GENERADOS POR EL PUBLICADOR.....	106
FIGURA III.19. ARCHIVO GENERADO POR EL SUSCRIPTOR.....	107

FIGURA III.20. ARCHIVO GENERADO POR EL DISTRIBUIDOR.....	107
FIGURA III.21. PUBLICACIÓN DE INFORMACIÓN	111
FIGURA III.22. PROCESO DE SINCRONIZACIÓN	112
FIGURA III.23. REPRESENTACIÓN ESQUEMÁTICA DE UNA ARQUITECTURA CLIENTE INTELIGENTE.....	113
FIGURA III.24. ESCALA PARA MEDIR LA SINCRONIZACIÓN.....	129
FIGURA IV.25. DIAGRAMA DE CASO DE USO GENERAL.....	146
FIGURA IV.26. DIAGRAMA DE CASO SINCRONIZACIÓN.....	147
FIGURA IV.27. DIAGRAMA DE CASO DE USO CONFIGURAR AGENDA.....	147
FIGURA IV.28. DIAGRAMA DE SECUENCIA PROCESO DE SINCRONIZACIÓN.....	154
FIGURA IV.29. DIAGRAMA DE SECUENCIA CONFIGURAR AGENDA.....	155
FIGURA IV.30. DIAGRAMA DE COMPONENTES.....	156
FIGURA IV.31. PANTALLA MENÚ PRINCIPAL.....	157
FIGURA IV.32. PANTALLA MENÚ PUBLICADOR.....	158
FIGURA IV.33. PANTALLA MENÚ REGISTRO SEVIDORES.....	158
FIGURA IV.34. PANTALLA CONFIGURAR CONEXIÓN.....	159

ÍNDICE DE TABLAS

TABLA II. I. OBJETOS CENTRALES QUE CONSTITUYEN UN PROVEEDOR DE DATOS DE .NET FRAMEWORK	98
TABLA III.II. ESPECIFICACIÓN DE VALORES DEL CAMPO ACROW.....	101
TABLA III.III. FUNCIONAMIENTO DE LOS TRIGGERS EN LAS TABLAS A PUBLICARSE.....	103
TABLA III.IV. ARCHIVOS GENERADOS POR LO SERVIDORES.....	108
TABLA III.V. TIPOS DE DATOS USADOS PARA LA COMPROBACIÓN DE LA HIPÓTESIS.....	118
TABLA III.VI. PRUEBAS DE INSERCIÓN DE DATOS (MYSQL COMO PUBLICADOR).....	121
TABLA III.VII. PRUEBAS DE ACTUALIZACIÓN DE DATOS (PUBLICADOR MYSQL).....	122
TABLA III.VIII. PRUEBAS DE ELIMINACIÓN DE DATOS (PUBLICADOR MYSQL).....	123
TABLA III.IX. PRUEBAS DE INSERCIÓN DE DATOS (PUBLICADOR SQL).....	124
TABLA III.X. PRUEBAS DE ACTUALIZACIÓN DE DATOS (PUBLICADOR SQL).....	125
TABLA III.XI. PRUEBAS DE ELIMINACIÓN DE DATOS (PUBLICADOR SQL).....	126
TABLA III.XII. RESUMEN DE INCONVENIENTES EN EL PROCESO DE REPLICA.....	128
TABLA IV.XIII. TABLA PERFILES DE USUARIOS DEL SISTEMA.....	133
TABLA IV.XIV. TABLA DE HERRAMIENTAS UTILIZADAS EN EL DESARROLLO DE LA HERRAMIENTA DE SINCRONIZACIÓN.....	136
TABLA IV.XV. TABLA DE LA ARQUITECTURA DE LA HERRAMIENTA DE SINCRONIZACIÓN.....	137
TABLA IV.XVI. TABLA DE RIESGOS.....	138
TABLA IV.XVII. TABLA DE VALORES DE RIESGOS.....	138
TABLA IV.XVIII. TABLA DE ANÁLISIS DE RIESGOS.....	139
TABLA IV.XIX. TABLA DE RESULTADOS DEL ANÁLISIS DE RIESGOS.....	139

TABLA IV.XX. TABLA DE RESTRICCIONES.....	140
TABLA IV.XXI. TABLA DEL RECURSO HUMANO – INTEGRANTES Y FUNCIONES.....	141
TABLA IV.XXII. TABLA DEL RECURSO FÍSICO.....	141
TABLA IV.XXIII. USUARIOS DEL SISTEMA.....	145
TABLA IV.XXIV. CASO DE USO GENERAL.....	148
TABLA IV.XXV. CASO DE USO SINCRONIZACIÓN.....	149
TABLA IV.XXVI. CONFIGURAR AGENDA.....	150
TABLA IV.XXVII. TABLA NOMENCLATURA DE PROGRAMACIÓN.....	157

INTRODUCCION

Para las distintas empresas que cuentan con diferentes sucursales es muy importante que sus bases de datos se encuentren sincronizadas entre sí, es decir, que se necesita que en algún instante de tiempo todas las bases de datos cuenten con la misma información para que de esta manera se tenga datos reales y consistentes; por esta razón se ha decidido crear una herramienta que independientemente de los motores de base de datos, logre sincronizar los datos, que como ya es conocido es la parte más importante de una empresa u organización.

El proceso de la replicación de datos es responsabilidad principal del motor de base de datos que normalmente corresponde a versiones empresariales, que solo por adjuntar la funcionalidad de la replicación tienen un costo adicional, es por esto que se ha decidido desarrollar un modelo de servicios web que permitan realizar el proceso de duplicación y sincronización de datos en bases de datos heterogéneas.

El modelo propuesto está compuesto por un conjunto de agentes (servicios web) que son los responsables de ejecutar las tareas de: captura de cambios en el publicador y suscriptores, registro de los servidores que participarán en el proceso de sincronización, mezcla de información, resolución de los posibles conflictos que se puedan presentar y sincronizar todos los servidores que se encuentran registrados.

Durante el desarrollo de la tesis se abordarán todos los temas relacionados a la replicación de bases de datos heterogéneas mediante servicios web, que es el objetivo principal de la tesis.

CAPÍTULO I MARCO PROPOSITIVO

1.1 ANTECEDENTES

En la actualidad, el proceso de replicación de datos es responsabilidad principal del motor de base de datos, que normalmente corresponde a versiones empresariales de los productos comerciales de mayor uso en el mercado. En este escenario, la aplicación depende de los servicios de distribución que ofrece el motor, y normalmente estos servicios “avanzados” tienen un costo adicional muy elevado.

Por lo anterior, las empresas enfrentan un problema económico, ya que deben pagar un valor adicional para adquirir “in house” servicios que no están incluidos en las versiones estándar de los actuales motores de base de datos. En otro contexto, cuando las empresas despliegan sus aplicaciones bajo el esquema de hosting, el problema es aún mayor, ya que los proveedores de servicios de Internet (Internet Service Provider, ISP) o los proveedores de servicios de aplicación (Application Service Provider, ASP), normalmente no pueden cubrir necesidades tan específicas en cuanto a tecnología y en el caso de hacerlo, el costo es muy elevado. Además, en muchos casos las empresas terminan pagando por un servicio que es mucho más de lo que realmente necesitan, es decir, en muchos casos sólo se necesita un método específico de replicación y no todo el conjunto general de posibilidades ofrecidas, y por las que se paga en conjunto.

Por otro lado, si bien la interoperabilidad entre motores de bases de datos se ha hecho un requerimiento necesario, pocos de ellos ofrecen verdaderos servicios que permitan replicar información entre distintos motores. Normalmente, en estos casos se recurre a productos de terceros que también tienen un costo elevado o bien no soportan la cantidad de motores de base de datos que se necesitan.

La solución propuesta para este problema consiste en asignar la responsabilidad de la replicación de datos a un modelo de servicios Web que soporte la replicación sobre múltiples motores de base de datos. Este modelo va acompañado de un conjunto de servicios del sistema operativo (agentes) que revisan la agenda de creación de publicaciones y la recepción de las mismas en los suscriptores.

En la actualidad CONTILIBRO ha implantado una solución web basada en VIRTUE MART que es un sistema GPL –GNU de comercio electrónico, el cual le permite llevar un cierto control de inventarios. En el caso de CONTILIBRO se necesita que las bases de datos de SQL Server (Aplicación Satélite en Desarrollo) y MYSQL estén sincronizadas de tal manera que cuando exista un cambio en la base de datos de SQL (base de datos local) este cambio también debe reflejarse en la base de datos que está en la web (base de datos MYSQL).

1.2 JUSTIFICACIÓN DEL ANTEPROYECTO DE TESIS

Para sustentar la razón, importancia y visión de la presentación del anteproyecto de tesis, establecemos dos tipos de justificación:

- Un elemento teórico encaminado al aporte investigativo, así como al aprovechamiento de los recursos tecnológicos.

- La justificación aplicativa correspondiente a la descripción de la aplicación práctica.

1.2.1 JUSTIFICACIÓN TEÓRICA

La replicación de base de datos permite asegurar la disponibilidad de los datos sincronizando copias de una fuente de datos en diferentes servidores.

Se dispone de un modelo de servicios web que permita replicar los datos aprovechando beneficios tales como:

1. Promover la interoperabilidad: La interacción entre un proveedor y un solicitante de servicio está diseñada para que sea completamente independiente de la plataforma y el lenguaje. Esta interacción requiere un documento WSDL para definir la interfaz y describir el servicio, junto con un protocolo de red (generalmente HTTP).
2. Permitir la integración “justoatiempo”: El proceso de descubrimiento se ejecuta dinámicamente, a medida que los solicitantes de servicio utilizan a los agentes para encontrar proveedores de servicio. Una vez el solicitante y el proveedor de servicio se han ubicado, se utiliza el documento WSDL del proveedor para enlazar al solicitante con el servicio. Esto significa que los solicitantes, los proveedores y los agentes actúan en conjunto para crear sistemas que son autoconfigurables, adaptativos y robustos.
3. Reducen la complejidad por medio del encapsulamiento: Los solicitantes y los proveedores del servicio se preocupan por las interfaces necesarias para interactuar. Como resultado, un solicitante de servicio no sabe cómo fue implementado el servicio por parte del proveedor, y éste a

su vez, no sabe cómo utiliza el cliente el servicio. Estos detalles se encapsulan en los solicitantes y proveedores. El encapsulamiento es crucial para reducir la complejidad.

4. Disminuyen el tiempo de desarrollo de las aplicaciones: Pues gracias a la filosofía de orientación a objetos utilizada, el desarrollo se convierte más bien en una labor de composición.

5. Seguridad: los servicios Web usan el protocolo WSSecurity (Web Service Security) aceptado como estándar por OASIS (Organization for the Advancement of Structured Information Standards). Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados. WSSecurity incorpora características de seguridad en el encabezado de un mensaje SOAP, trabajando en la capa aplicación. Así asegura seguridad extremo a extremo.

Los agentes de réplica realizan muchas tareas asociadas con la réplica, lo que incluye la creación de copias de esquema y datos, detección de actualizaciones en el publicador o el suscriptor, y propagación de cambios entre servidores. Cada agente de réplica admite un conjunto de parámetros en tiempo de ejecución que se utilizan para controlar cómo se ejecuta; estos parámetros se especifican en un perfil de agente o en la línea de comandos.

En el momento que se ejecuten estos agentes de réplica deberán usar los servicios web que están adaptados a los requerimientos que implica una replicación de datos, es decir, para que los servicios web permitan realizar

esta tarea estos deben basarse en los conceptos y elementos propios de una replicación como:

- Publicador
- Distribuidor
- Suscriptor

1.2.2 JUSTIFICACIÓN APLICATIVA

La empresa CONTILIBRO desea contar con una aplicación satélite que le permita:

- Llevar el control de sus clientes, administración de relación con los clientes (CRM), vendedores y proveedores.
- Inventarios.
- Facturación.
- Cuentas por cobrar y cuentas pagar.

La obtención de los datos para esta aplicación se realizará a través de replicas a la base de datos local a través de una aplicación de escritorio que se desarrollará en .NET. El proceso de replicas entre las bases de datos (MySQL y SQLServer) se realizará usando un modelo de servicios web.

El modelo de replicación propuesto está compuesto por:

1. Un conjunto de servicios Web XML (eXtensible Markup Language) que expone la lógica de replicación.
2. Agente de publicación y agente de suscripción que usan los servicios Web que a su vez pueden llamar a otros servicios Web ubicados en otros equipos (servidores) de la red.

Este modelo proporciona los objetos necesarios (agentes y servicios) para establecer los distintos roles (publicador, suscriptor, distribuidor) que involucra la réplica de datos permitiendo la sincronización de los datos de la nueva aplicación de escritorio con el sistema VIRTUE MART, además permitirá ahorrar recursos como el Internet debido a que el tiempo requerido para la conexión es mínimo.

1.3 OBJETIVOS.

1.3.1 OBJETIVO GENERAL:

- Realizar el estudio de agentes en servicios web para la replicación de bases de datos heterogéneas para la aplicación Satélite de CONTILIBRO.

1.3.2 OBJETIVOS ESPECÍFICOS:

- Realizar el modelo de servicios web necesarios para publicar y consumir datos entre motores de bases de datos heterogéneos.

- Desarrollar agentes de replica para realizar tareas asociadas con el seguimiento de cambios y la distribución de los datos los mismos que se ejecutarán en un determinado periodo de tiempo.
- Desarrollar la aplicación de escritorio de CONTILIBRO (aplicación satélite) e implementar el modelo de servicios web propuesto.
- Verificar que el proceso de sincronización de datos corresponda con los requerimientos.

1.4 HIPÓTESIS

Los agentes de réplica en servicios web permitirán la sincronización de los sistemas satélites de CONTILIBRO.

CAPÍTULO II MARCO TEÓRICO TECNOLOGIAS PARA LA DISTRIBUCION Y ACCESO A DATOS

2.1 REPLICACION DE DATOS CON SQL SERVER

2.1.1 INTRODUCCIÓN A LA REPLICACIÓN DE DATOS EN SQL SERVER

La replicación de datos permite que ciertos datos de la base de datos sean almacenados en más de un sitio, y su principal utilidad es que permite aumentar la disponibilidad de los datos y mejora el funcionamiento de las consultas globales a la base de datos.

La replicación transaccional se usa normalmente en escenarios servidor a servidor que requieren un alto rendimiento, como por ejemplo, la mejora de la escalabilidad y la disponibilidad, el almacenamiento de datos y la creación de informes, la integración de datos procedentes de varios sitios, la integración de datos heterogéneos, y la descarga del procesamiento por lotes. La replicación de mezcla se ha diseñado principalmente para las aplicaciones móviles o de servidores distribuidos que pueden encontrarse con conflictos de datos. Los escenarios más frecuentes son: el intercambio de datos con usuarios móviles, las aplicaciones de punto de venta (POS) a consumidores, y la integración de datos de varios sitios. La replicación de datos se usa para proporcionar el conjunto de datos inicial para la replicación

transaccional y de mezcla; también se puede usar cuando está indicada una actualización completa de los datos. Con estos tres tipos de replicación, SQL Server proporciona un sistema eficaz y flexible para la sincronización de datos en toda la organización.

Según msdn Microsoft Developer Network^[1]: La replicación es un conjunto de tecnologías destinadas a la copia y distribución de datos y objetos de bases de datos desde una base de datos a otra, para luego sincronizar ambas bases de datos y mantener su coherencia. La replicación permite distribuir datos entre diferentes ubicaciones y entre usuarios remotos o móviles mediante redes locales y de área extensa, conexiones de acceso telefónico, conexiones inalámbricas e internet.

La replicación en SQL Server consiste, en el transporte de datos entre dos o más instancias de servidores. Para ello SQL Server brinda un conjunto de soluciones que permite copiar, distribuir y posiblemente modificar datos de toda la organización. Se incluyen, además, varios métodos y opciones para el diseño, implementación, supervisión y administración de la replicación, que le ofrecen la funcionalidad y flexibilidad necesarias para distribuir datos y mantener su coherencia.

En la replicación se utiliza una metáfora de la industria de la publicación para representar los componentes y procesos de una topología de replicación. De esta forma el modelo se compone, básicamente, de los siguientes

¹ <http://msdn.microsoft.com/es-es/library/ms151198.aspx>

elementos: publicador, distribuidor, suscriptores, publicaciones, artículos y suscripciones.

2.1.2 COMPONENTES DEL MODELO DE REPLICACIÓN

Para representar los componentes y procesos de una topología de replicación se utilizan metáforas de la industria de la publicación. El modelo se compone de los siguientes objetos: el publicador, el distribuidor, el suscriptor, la publicación, el artículo y la suscripción; así como de varios agentes, que son los procesos responsabilizados de copiar los datos entre el publicador y el suscriptor. Estos agentes son: agente de instantáneas, agente de distribución, agente del lector del registro, agente del lector de cola y agente de mezcla.

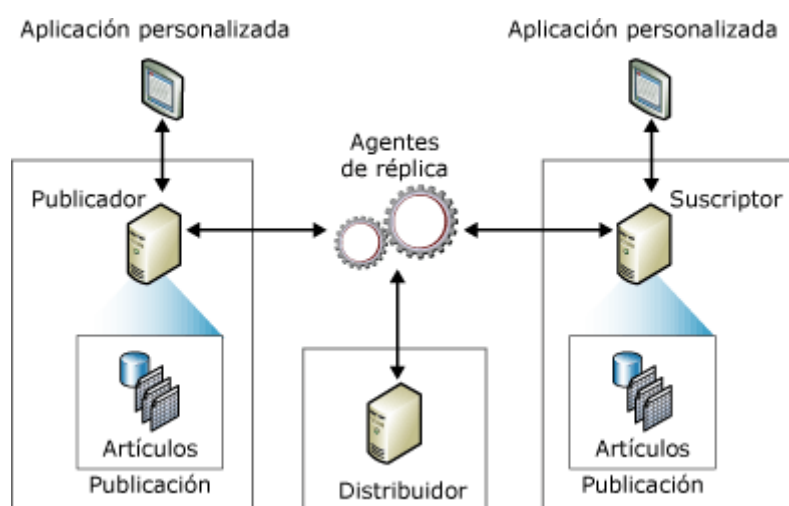


Figura II.1 Componentes de Replica

2.1.2.1 Publicador

Es un servidor que pone los datos a disposición de otros servidores para poder replicarlos.

Según msdn Microsoft Developer Network²: El publicador es una instancia de base de datos que permite que los datos estén disponibles para otras ubicaciones a través de la réplica. El publicador puede tener una o más publicaciones, cada una de las cuales representa un conjunto de datos de objetos relacionados lógicamente para su réplica.

2.1.2.2 Distribuidor

Es un servidor que aloja la base de datos de distribución y almacena los datos históricos, transacciones y metadatos.

Según msdn Microsoft Developer Network: El distribuidor es una instancia de base de datos que funciona como almacén para datos específicos de réplica asociados a uno o varios publicadores. Cada publicador está asociado a una sola base de datos (conocida como la base de datos de distribución) en el distribuidor. La base de datos de distribución almacena los datos de estado de la réplica, los metadatos acerca de la publicación y, en algunos casos, funciona como cola para los datos que se transfieren del publicador a los suscriptores. En muchos casos, una sola instancia de servidor de bases de datos funciona como publicador y como distribuidor. Esto se conoce como un distribuidor local. Cuando el publicador y el distribuidor se configuran en instancias distintas del servidor de bases de datos, el distribuidor se denomina distribuidor remoto.

2.1.2.3 Suscriptor

² [http://msdn.microsoft.com/es-es/library/ms165654\(SQL.90\).aspx](http://msdn.microsoft.com/es-es/library/ms165654(SQL.90).aspx)

Según msdn Microsoft Developer Network³: Un suscriptor es una instancia de base de datos que recibe datos replicados. Un suscriptor puede recibir datos de varios publicadores y publicaciones. En función del tipo de réplica elegida, el suscriptor también puede devolver los datos modificados al publicador o volver a publicar los datos en otros suscriptores.

2.1.2.4 Publicación

Es un conjunto de artículos de una base de datos. Esta agrupación de varios artículos facilita especificar un conjunto de datos relacionados lógicamente y los objetos de bases de datos que desea replicar conjuntamente. Un artículo de una publicación puede ser una tabla de datos la cual puede contar con todas las filas o algunas (filtrado horizontal) y simultáneamente contar de todas las columnas o algunas (filtrado vertical), un procedimiento almacenado, una definición de vista, la ejecución de un procedimiento almacenado, una vista, una vista indizada o una función definida por el usuario.

Una publicación es un conjunto de uno o más artículos de una base de datos. La agrupación de varios artículos en una publicación permite especificar más fácilmente un conjunto de objetos y datos de bases de datos relacionados lógicamente, que se replican como una unidad.

2.1.2.5 Suscripción

³ [http://msdn.microsoft.com/es-es/library/ms165654\(SQL.90\).aspx](http://msdn.microsoft.com/es-es/library/ms165654(SQL.90).aspx)

Es una petición de copia de datos o de objetos de base de datos para replicar. Una suscripción define qué publicación se recibirá, dónde y cuándo. Las suscripciones pueden ser de inserción o de extracción; y una publicación puede admitir una combinación de suscripciones de inserción y extracción. El publicador (en las suscripciones de inserción) o el suscriptor (en las suscripciones de extracción) solicita la sincronización o distribución de datos de una suscripción.

El publicador puede disponer de una o más publicaciones, de las cuales los suscriptores se suscriben a las publicaciones que necesitan, nunca a artículos individuales de una publicación. El publicador, además, detecta qué datos han cambiado durante la replicación transaccional y mantiene información acerca de todas las publicaciones del sitio.

La función del distribuidor varía según la metodología de replicación implementada. En ocasiones se configura como distribuidor el mismo publicador y se le denomina distribuidor local. En el resto de los casos el distribuidor será remoto, pudiendo coincidir en algún caso con un suscriptor.

Según msdn Microsoft Developer Network⁴: Una suscripción es una solicitud de una copia de una publicación que se entrega a un suscriptor. La suscripción define qué publicación se recibirá, dónde y cuándo. Hay dos tipos de suscripciones: de inserción y de extracción.

⁴ [http://msdn.microsoft.com/es-es/library/ms165654\(SQL.90\).aspx](http://msdn.microsoft.com/es-es/library/ms165654(SQL.90).aspx)

2.1.3 Agentes de Replica

La réplica utiliza varios programas independientes, llamados agentes, para realizar las tareas asociadas con el seguimiento de los cambios y la distribución de los datos. De forma predeterminada, los agentes de réplica se ejecutan como trabajos programados en el Agente SQL Server y es necesario que se esté ejecutando el Agente SQL Server para que puedan ejecutarse los trabajos.

El Agente SQL Server aloja y programa los agentes utilizados en la réplica, y proporciona una manera sencilla de ejecutar los agentes de réplica. El Agente SQL Server también controla y supervisa las operaciones fuera de la réplica.

Los agentes que intervienen en la réplica de datos en SQL Server son:

2.1.3.1 Agente de instantáneas

El Agente de instantáneas se utiliza con todos los tipos de réplica. Prepara esquemas y archivos de datos iniciales de tablas publicadas y otros objetos, almacena los archivos de instantáneas y registra la información acerca del estado de sincronización en la base de datos de distribución. El Agente de instantáneas se ejecuta en el distribuidor.

La herramienta Agente de instantáneas de duplicación configura e inicia el Agente de instantáneas, que prepara archivos de instantáneas que contienen esquemas y datos de tablas publicadas y objetos de base de datos, almacena los archivos en la carpeta de instantáneas y registra los trabajos de sincronización en la base de datos de distribución.

2.1.3.2 Agente de Registro de LOG

El Agente de registro del LOG se utiliza en la réplica transaccional. Mueve las transacciones marcadas para réplica desde el registro de transacciones del publicador a la base de datos de distribución. Cada base de datos publicada con la réplica transaccional tiene su propio Agente de registro del LOG, que se ejecuta en el distribuidor y se conecta al publicador (el distribuidor puede estar en el mismo equipo que el publicador).

La herramienta Agente de lector del registro de duplicación configura e inicia el Agente de lector del registro, que supervisa el registro de transacciones de todas las bases de datos configuradas para duplicación y copia a la base de datos de distribución las marcadas para duplicación en el registro de transacciones.

2.1.3.3 Agente de distribución

El Agente de distribución se utiliza en la réplica de instantáneas y transaccional. Aplica la instantánea inicial al suscriptor y mueve las transacciones contenidas en la base de datos de distribución a los suscriptores. El Agente de distribución se ejecuta en el distribuidor, para

las suscripciones de inserción, o en el suscriptor, para las suscripciones de extracción.

El Agente de distribución de duplicación configura e inicia el Agente de distribución, que mueve la instantánea (para duplicación de instantánea y transaccional) existente en las tablas de la base de datos de distribución (para duplicación transaccional) a las tablas de destino de Suscriptores.

2.1.3.4 Agente de mezcla

El Agente de mezcla se utiliza con la réplica de mezcla. Aplica la instantánea inicial al suscriptor, y transfiere y reconcilia los cambios incrementales de datos que se producen. Cada suscripción de mezcla tiene su propio Agente de mezcla, que se conecta con el publicador y con el suscriptor, y los actualiza. El Agente de mezcla se ejecuta en el distribuidor, para las suscripciones de inserción, o en el suscriptor, para las suscripciones de extracción. De forma predeterminada, el Agente de mezcla carga los cambios del suscriptor al publicador y, a continuación, descarga los cambios del publicador al suscriptor.

La herramienta Agente de mezcla de duplicación configura e inicia el Agente de mezcla de duplicación, que aplica a los suscriptores instantáneas iniciales que se encuentran en las tablas de bases de datos de los suscriptores. También mezcla los cambios incrementales de los datos que tienen lugar en el publicador después de la creación de la instantánea inicial, y reconcilia los conflictos de acuerdo con unas reglas configuradas.

2.1.3.5 Agente de lectura de cola

El Agente de lectura de cola se utiliza con la réplica transaccional y la opción de actualización en cola. El agente se ejecuta en el distribuidor y transfiere los cambios realizados en el suscriptor de vuelta al publicador. A diferencia del Agente de distribución y del Agente de mezcla, sólo existe una instancia del Agente de lectura de cola para todos los publicadores y las publicaciones de una determinada base de datos.

La herramienta Agente de lectura de cola de duplicación configura e inicia el Agente de lectura de cola, que lee los mensajes almacenados en una cola de SQL Server o de Microsoft Message Queue y, a continuación, aplica estos mensajes al Publicador. El Agente de lectura de cola se utiliza con publicaciones de instantáneas y transaccionales que admitan la actualización en cola.

2.1.4 TIPOS DE REPLICACIÓN

Los tipos básicos de replicación son:

- ✓ Replicación de instantáneas
- ✓ Replicación transaccional
- ✓ Replicación de mezcla

2.1.4.1 Replicación de instantáneas

En la replicación de instantáneas los datos se copian tal y como aparecen exactamente en un momento determinado. Por consiguiente, no requiere un control continuo de los cambios. Las publicaciones de instantáneas se suelen replicar con menos frecuencia que otros tipos de publicaciones. Puede llevar más tiempo propagar las modificaciones de datos a los suscriptores. Se recomienda utilizar: cuando la mayoría de los datos no cambian con frecuencia; se replican pequeñas cantidades de datos; los sitios con frecuencia están desconectados y es aceptable un periodo de latencia largo (la cantidad de tiempo que transcurre entre la actualización de los datos en un sitio y en otro).

En ocasiones se hace necesario utilizarla cuando están involucrados algunos tipos de datos (text, ntext, e image) cuyas modificaciones no se registran en el registro de transacciones y por tanto no se pueden replicar utilizando la metodología de replicación transaccional.

¿Cómo funciona la duplicación de instantáneas?

La duplicación de instantáneas la implementan el Agente de instantáneas y el Agente de distribución. El Agente de instantáneas prepara archivos de instantáneas que contienen esquemas y datos de las tablas y objetos de base de datos publicados, almacena los archivos en la carpeta de instantáneas y registra los trabajos de sincronización en la base de datos de distribución del distribuidor. De forma predeterminada, la carpeta de instantáneas se encuentra en el distribuidor, pero puede especificar una ubicación alternativa, que sustituya o complemente a la predeterminada.

El Agente de distribución mueve las instantáneas almacenadas en las tablas de la base de datos de distribución a las tablas de destino de los suscriptores. La base de datos de distribución sólo se utiliza para la duplicación y no contiene tablas de usuario.

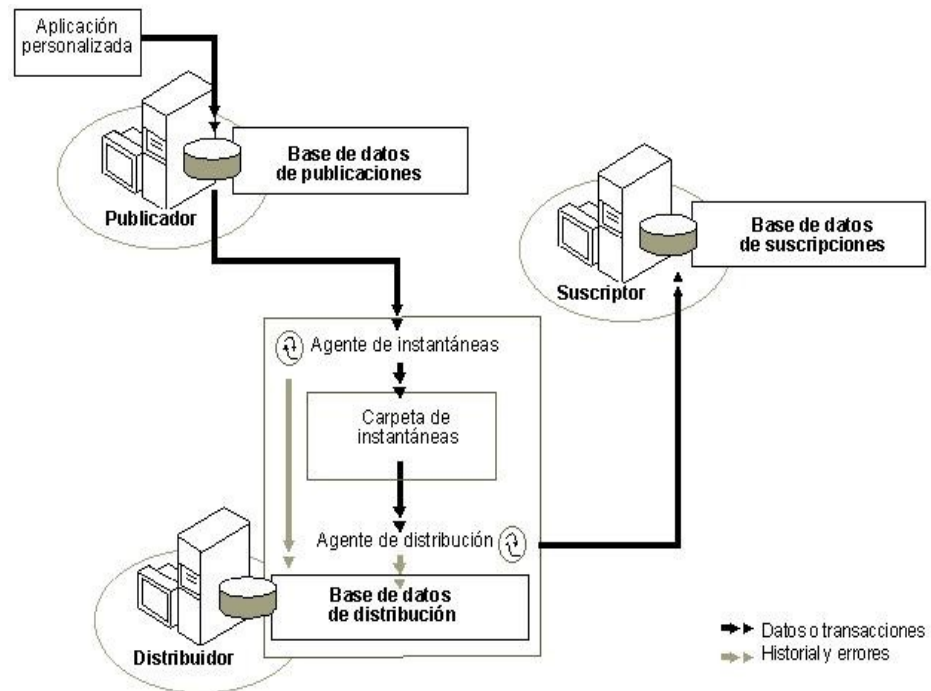


Figura II. 2 Duplicación de Instantáneas

Agente de instantáneas

Cada vez que se ejecuta el Agente de instantáneas, comprueba si se ha agregado alguna suscripción nueva. Si no hay suscripciones nuevas, no se crean secuencias de comandos ni archivos de datos nuevos. Si la publicación se creó con la opción de crear inmediatamente la primera instantánea habilitada, cada vez que se ejecuta el Agente de instantáneas se crean nuevos esquemas y archivos de datos. Todos los

esquemas y los archivos de datos se almacenan en la carpeta de instantáneas, y a continuación el Agente de distribución o el Agente de mezcla los transfiere al suscriptor, o bien el usuario puede realizar la transferencia manualmente. El Agente de instantáneas ejecuta los pasos siguientes:

Establece una conexión desde el distribuidor al publicador y establece un bloqueo compartido de todas las tablas incluidas en la publicación. El bloqueo compartido asegura que la instantánea de los datos sea coherente. Como los bloqueos impiden al resto de los usuarios actualizar las tablas, el Agente de instantáneas debe ser programado para que se ejecute durante los periodos de baja actividad de la base de datos.

Establece una conexión desde el publicador al distribuidor y escribe una copia del esquema de la tabla de cada artículo en un archivo .SCH. Si solicita que se incluyan los índices y la integridad referencial declarativa, el agente generará una secuencia de comandos para extraer los índices seleccionados en un archivo .IDX. También puede publicar, como parte de la duplicación, otros objetos de la base de datos tales como procedimientos almacenados, vistas, funciones definidas por el usuario y otros.

Copia los datos de la tabla publicada en el publicador y escribe los datos en la carpeta de instantáneas. Si todos los suscriptores son instancias de Microsoft® SQL Server, la instantánea se almacena en

forma de archivo de programa nativo de copia masiva. Si uno o más de los suscriptores es un origen de datos heterogéneo, la instantánea se almacena como archivo de modo de caracteres. Los archivos son el conjunto de sincronización que representa la tabla en un momento concreto. Hay un conjunto de sincronización para cada artículo de una publicación.

Agrega filas a las tablas MSrepl_commands y MSrepl_transactions de la base de datos de distribución. Las entradas de las tablas MSrepl_commands son comandos que indican la ubicación del conjunto de sincronización (archivos .SCH y .BCP) y referencias a cualquier secuencia de comandos previa a la creación especificada. Las entradas de la tabla MSrepl_transactions son comandos que hacen referencia a la tarea de sincronización del suscriptor.

Libera los bloqueos compartidos de las tablas publicadas y termina de escribir las tablas del historial de registro.

Agente de distribución

Cada vez que se ejecuta para una publicación de instantáneas, el Agente de distribución mueve el esquema y los datos a los suscriptores.

El Agente de distribución ejecuta los pasos siguientes:

Establece una conexión desde el servidor en el que se encuentra el agente al distribuidor. En las suscripciones de inserción, el Agente de

distribución suele ejecutarse en el distribuidor; en las suscripciones de extracción, suele ejecutarse en el suscriptor.

Examina las tablas MSrepl_commands y MSrepl_transactions de la base de datos de distribución del distribuidor. El agente lee la ubicación del conjunto de sincronización de la primera tabla y los comandos de sincronización del suscriptor de ambas tablas.

Aplica el esquema y los comandos a la base de datos de suscripción. Si el suscriptor no es una instancia de Microsoft SQL Server 2000, el agente convierte los tipos de datos según sea necesario. Todos los artículos de una publicación se sincronizan, conservando la integridad transaccional y referencial entre las tablas subyacentes (asumiendo que la base de datos de suscripción, si no es SQL Server, tiene las capacidades transaccionales necesarias para hacerlo).

Cuando se maneja un número elevado de suscriptores, la ejecución del Agente de distribución en el suscriptor, utilizando suscripciones de extracción o utilizando la activación remota de agentes, puede ahorrar recursos de procesamiento en el distribuidor. Con la activación remota de agentes puede elegir si el Agente de distribución se ejecuta en el suscriptor, para las suscripciones de inserción, o en el distribuidor para las suscripciones de extracción.

Las instantáneas se pueden aplicar cuando se crea la suscripción o de acuerdo con un programa establecido al crear la publicación.

Como la sincronización automática de bases de datos o de tablas individuales necesita más recursos del sistema, una de las ventajas de programar la sincronización automática a intervalos menos frecuentes

es que permite que la instantánea inicial se programe para un periodo de baja actividad en el publicador.

El agente de instantáneas lo ejecuta normalmente el Agente SQL Server y se puede administrar directamente mediante el Administrador corporativo de SQL Server. El Agente de instantáneas y el Agente de distribución también se pueden incrustar en aplicaciones mediante controles de Microsoft ActiveX®. El Agente de instantáneas se ejecuta en el distribuidor. El Agente de distribución suele ejecutarse en el distribuidor, para las suscripciones de inserción, o en los suscriptores para las suscripciones de extracción; la activación remota de agentes puede utilizarse para descargar en otro servidor el procesamiento del Agente de distribución.

2.1.4.2 Replicación Transaccional

En este caso se propaga una instantánea inicial de datos a los suscriptores, y después, cuando se efectúan las modificaciones en el publicador, las transacciones individuales se propagan a los suscriptores. SQL Server 2000 almacena las transacciones que afectan a los objetos replicados y propaga esos cambios a los suscriptores de forma continua o a intervalos programados. Al finalizar la propagación de los cambios, todos los suscriptores tendrán los mismos valores que el publicador. Suele utilizarse cuando: se desea que las modificaciones de datos se propaguen a los suscriptores, normalmente pocos segundos después de producirse; se necesita que las transacciones sean atómicas, que se apliquen todas o ninguna al suscriptor; los suscriptores se conectan en su mayoría al publicador; su aplicación no

puede permitir un periodo de latencia largo para los suscriptores que reciban cambios.

Es útil en escenarios en los que los suscriptores pueden tratar a sus datos como de sólo lectura, pero necesitan cambios a los datos con una cantidad mínima de latencia.

Cómo funciona la duplicación transaccional

La duplicación transaccional la implementan el Agente de instantáneas, el Agente de lector del registro y el Agente de distribución. El Agente de instantáneas prepara archivos de instantáneas que contienen esquemas y datos de las tablas y objetos de base de datos publicados, almacena los archivos en la carpeta de instantáneas y registra los trabajos de sincronización en la base de datos de distribución del distribuidor.

El Agente de lector de registro supervisa el registro de transacciones de cada base de datos configurada para la duplicación transaccional y copia las transacciones marcadas para ser duplicadas desde el registro de transacciones a la base de datos de distribución. El Agente de distribución pasa los trabajos de instantánea inicial y las transacciones almacenadas en las tablas de la base de datos de distribución a los suscriptores.

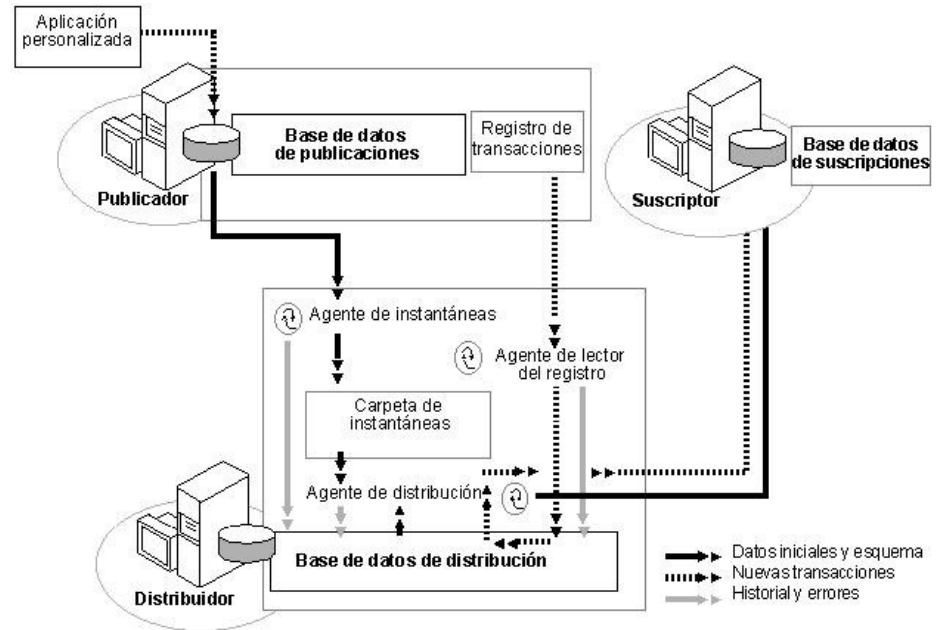


Figura II.3 Duplicación Transaccional

Instantánea inicial

Antes de que un suscriptor de duplicación transaccional pueda recibir los cambios incrementales desde un publicador, debe contener tablas con el mismo esquema y los mismos datos que las tablas del publicador. A la copia de la publicación actual completa desde el publicador al suscriptor se le llama aplicar la instantánea inicial. Microsoft® SQL Server creará y aplicará la instantánea de forma automática, aunque también puede elegir aplicarla manualmente.

Cuando se distribuyen y se aplican instantáneas a los suscriptores, sólo se ven afectados los suscriptores que estén esperando instantáneas iniciales. Los demás suscriptores de la publicación (aquellos que ya están recibiendo inserciones, actualizaciones, eliminaciones u otras modificaciones de los datos publicados) no se ven afectados.

Procesamiento simultáneo de instantáneas

Habitualmente, al generar instantáneas, SQL Server pone bloqueos compartidos en todas las tablas que se publican como parte de la duplicación, mientras dura la generación de la instantánea. Así se puede evitar que se realicen actualizaciones en las tablas de publicación. El proceso simultáneo de instantáneas, disponible solamente con la duplicación transaccional, no mantiene los bloqueos compartidos durante la generación completa de la instantánea, por lo tanto, permite a los usuarios continuar el trabajo sin interrupción mientras SQL Server crea los archivos de instantáneas iniciales.

Si crea una nueva publicación mediante la duplicación transaccional e indica que todos los suscriptores serán instancias de SQL Server 7.0 o SQL Server, está disponible el proceso simultáneo de instantáneas.

Una vez que se inicia la duplicación, el Agente de instantáneas aplica bloqueos compartidos a las tablas de la publicación. Los bloqueos impiden los cambios hasta que en el archivo de registro se introduce un registro que indica el inicio de la instantánea. Una vez se recibe la transacción, los bloqueos compartidos se liberan y las modificaciones de los datos de la base de datos pueden continuar. La duración de mantenimiento de los bloqueos es muy breve (unos segundos), aunque se estén copiando grandes cantidades de datos.

En este momento, el Agente de instantáneas empieza a generar los archivos de instantánea. Cuando se completa la instantánea, en el registro se escribe un segundo registro que indica el final del proceso de instantánea. Las transacciones que afectan a las tablas durante la generación de la instantánea se capturan entre las marcas de inicio y

final, y el Agente de lector del registro las reenvía a la base de datos de distribución.

Cuando se aplica la instantánea al suscriptor, el Agente de distribución aplica primero los archivos de instantánea (esquema y archivos .BCP). Entonces reconcilia cada transacción capturada, para ver si ya se entregó al suscriptor. Durante este proceso de reconciliación, las tablas del suscriptor se bloquean. Dependiendo del número de transacciones capturadas en el publicador durante la creación de la instantánea, deberá esperar una prolongación del tiempo necesario para aplicar la instantánea al suscriptor. Conceptualmente, esto es parecido al proceso de recuperación que utiliza SQL Server al reiniciarse.

Como se ha mencionado anteriormente, se presta atención al proceso simultáneo de instantáneas en sistemas donde la lógica empresarial se indica mediante desencadenadores o restricciones en la base de datos de suscripciones. El proceso simultáneo de instantáneas utiliza inserciones masivas de tablas seguidas de una serie de instrucciones INSERT y DELETE especiales que aportan a la tabla a un estado coherente. Estas operaciones se ejecutan como una transacción, de modo que los usuarios de la base de datos no ven los datos en un estado incoherente; no obstante, las restricciones del Suscriptor se ejecutarán dentro de la transacción y pueden evaluar cambios que no se basan en un conjunto coherente de datos. Para impedirlo, generalmente se recomienda que especifique la opción NOT FOR REPLICATION en todas las restricciones y columnas con la propiedad IDENTITY en la base de datos del Suscriptor. La lógica empresarial implementada mediante procedimientos almacenados personalizados

no se verá afectada porque estos procedimientos no se utilizan durante el proceso simultáneo de instantáneas hasta que las tablas del Suscriptor ofrezcan un estado coherente.

Las restricciones FOREIGN KEY, las restricciones CHECK y los desencadenadores en el Suscriptor no requieren la opción NOT FOR REPLICATION porque se deshabilitarán durante la generación de instantáneas simultáneas y se habilitarán después de haberlas generado.

El Agente de lector del registro debe ejecutarse después de generar la instantánea con procesamiento simultáneo. Si no se ejecuta el Agente de lector del registro, el Agente de distribución continuará devolviendo un error que indica que la instantánea no está disponible y no la aplicará a los suscriptores. Para que el Agente de distribución pueda aplicar la instantánea a los suscriptores, el Agente de lector del registro debe propagar a la base de datos de distribución todos los cambios ocurridos durante la generación de la instantánea. Habitualmente, el Agente de lector del registro se ejecuta en modo continuo, por lo que se ejecutará automáticamente poco después de que se genere la instantánea. Si se elige no ejecutar el Agente de lector del registro en modo continuo, se deberá ejecutarlo de forma manual.

Aunque el procesamiento simultáneo de instantáneas permite que continúen las actualizaciones sobre las tablas de la publicación, el rendimiento disminuirá debido a la sobrecarga propia de la instantánea.

Agente de instantáneas

Los procedimientos por los que el Agente de instantáneas implementa la instantánea inicial en la duplicación transaccional son los mismos que se utilizan en la duplicación de instantáneas (excepto en lo indicado anteriormente acerca del procesamiento simultáneo de instantáneas).

Modificar datos y el Agente de lector del registro

El Agente de lector del registro se ejecuta de forma continua o de acuerdo con un programa establecido al crear la publicación. Al ejecutarse, el Agente de lector del registro lee primero el registro de transacciones de la publicación (el mismo registro de la base de datos que se utiliza para seguir las transacciones y realizar las recuperaciones durante el funcionamiento normal de SQL Server 2000) e identifica las instrucciones INSERT, UPDATE y DELETE, u otras modificaciones efectuadas en las transacciones de datos que se hayan marcado para ser duplicadas. A continuación, el agente copia por lotes dichas transacciones en la base de datos de distribución del distribuidor. El Agente de lector del registro utiliza el procedimiento almacenado interno `sp_replcmds` para obtener el siguiente conjunto de comandos marcados para su duplicación desde el registro. La base de datos de distribución se convierte así en una cola de almacenamiento y reenvío desde la que se envían los cambios a los suscriptores. A la base de datos de distribución sólo se envían las transacciones confirmadas.

Hay una correspondencia de uno a uno entre las transacciones del publicador y las transacciones de duplicación de la base de datos de

distribución. Una transacción almacenada en `MSrepl_transactions` puede componerse de uno o más comandos y cada comando se puede dividir en partes de 500 caracteres Unicode en la tabla `MSrepl_commands`. Después de que el lote completo de transacciones se ha escrito correctamente en la base de datos de distribución, se confirma. A continuación de la confirmación de cada lote de comandos en el distribuidor, el agente de lector del registro llama a `sp_repldone` para marcar hasta dónde se ha llegado en la duplicación. Por último, el agente marca las filas del registro de transacciones que están listas para ser truncadas. Las filas que están en espera de ser duplicadas no se truncan. El registro de transacciones del publicador se puede volcar sin interferir con la duplicación, puesto que sólo se purgan las transacciones no marcadas para duplicación.

Las modificaciones de datos realizadas en el Suscriptor siempre se propagarán como una serie de instrucciones de una sola fila, siempre que no modifiquen una columna restringida exclusivamente. Si una instrucción `UPDATE` no modifica una columna restringida de forma única, `UPDATE` se propagará como una serie de instrucciones `DELETE` seguida de una serie de instrucciones `INSERT`. Una columna restringida de forma única es una columna que participa en un índice único o en un índice agrupado, aunque el índice agrupado no se declare como único. La instrucción `UPDATES` hace que las vistas indizadas o las tablas base en las que se basan las vistas indizadas se propaguen como parejas `DELETE/INSERT`.

Agente de distribución

Los comandos de las transacciones están almacenados en la base de datos de distribución hasta que el Agente de distribución los propaga a todos los suscriptores o hasta que un Agente de distribución del suscriptor extrae los cambios. La base de datos de distribución sólo se utiliza para la duplicación y no contiene tablas de usuario. No debe crear otros objetos en la base de datos de distribución. Los suscriptores recibirán las transacciones en el mismo orden en que fueron aplicadas en el Publicador.

El Agente de distribución es un componente del Agente SQL Server y se puede administrar directamente mediante el Administrador corporativo de SQL Server. El Agente de instantáneas y el Agente de distribución también se pueden incrustar en aplicaciones mediante controles de Microsoft ActiveX®. El Agente de instantáneas se ejecuta en el distribuidor. El Agente de distribución suele ejecutarse en el distribuidor, para las suscripciones de inserción, o en los suscriptores para las suscripciones de extracción; la activación remota de agentes puede utilizarse para descargar en otro servidor el procesamiento del agente.

SQL Server puede validar los datos que se actualizan en el suscriptor durante el proceso de duplicación, para garantizar que los datos son los mismos en el publicador y en los suscriptores.

Limpieza de la duplicación transaccional

Cuando se crea la base de datos de distribución, SQL Server agrega automáticamente las siguientes tareas al Agente SQL Server del distribuidor, para purgar los datos que ya no sean necesarios:

- ✓ Comprobación del agente
- ✓ Limpieza del historial del agente
- ✓ Limpieza de transacciones
- ✓ Limpieza de distribución
- ✓ Limpieza del historial
- ✓ Limpieza de suscripción caducada

Cuando todos los suscriptores han recibido transacciones, el Agente de limpieza de distribución quita de la base de datos de distribución las transacciones entregadas. Las transacciones entregadas se mantienen en la base de datos de distribución durante un periodo definido, conocido como periodo de retención. El establecimiento del periodo de retención durante la programación de copias de seguridad puede asegurar que la información necesaria para recuperar automáticamente una base de datos de destino se encuentre disponible en la base de datos de distribución.

2.1.4.3 Replicación de mezcla

Permite que varios sitios funcionen en línea o desconectados de manera autónoma, y mezclar más adelante las modificaciones de datos

realizadas en un resultado único y uniforme. La instantánea inicial se aplica a los suscriptores; a continuación SQL Server 2000 hace un seguimiento de los cambios realizados en los datos publicados en el publicador y en los suscriptores. Los datos se sincronizan entre los servidores a una hora programada o a petición. Las actualizaciones se realizan de manera independiente, sin protocolo de confirmación, en más de un servidor, así el publicador o más de un suscriptor pueden haber actualizado los mismos datos. Por lo tanto, pueden producirse conflictos al mezclar las modificaciones de datos. Cuando se produce un conflicto, el Agente de mezcla invoca una resolución para determinar qué datos se aceptarán y se propagarán a otros sitios. Es útil cuando: varios suscriptores necesitan actualizar datos en diferentes ocasiones y propagar los cambios al publicador y a otros suscriptores; los suscriptores necesitan recibir datos, realizar cambios sin conexión y sincronizar más adelante los cambios con el publicador y otros suscriptores; el requisito de periodo de latencia de la aplicación es largo o corto; la autonomía del sitio es un factor crucial.

Cómo funciona la duplicación de mezcla

La duplicación de mezcla la implementan el Agente de instantáneas y el Agente de mezcla. El Agente de instantáneas prepara archivos de instantáneas que contienen el esquema y los datos de las tablas publicadas, almacena los archivos en el directorio de instantáneas e inserta trabajos de sincronización en la base de datos de publicación. El Agente de instantáneas crea también desencadenadores, tablas del sistema y procedimientos almacenados específicos de la duplicación.

El Agente de mezcla aplica al suscriptor los trabajos de instantáneas iniciales almacenados en las tablas de la base de datos de publicación. También mezcla los cambios incrementales de los datos que tienen lugar en el Publicador o en los suscriptores después de la creación de la instantánea inicial y reconcilia los conflictos de acuerdo con las reglas que se configuran o la resolución de conflictos personalizada que se cree.

La función del Distribuidor está muy limitada en la duplicación de mezcla, por lo que es muy habitual implementar el distribuidor localmente (en el mismo servidor que el Publicador). El Agente de distribución no se utiliza durante la duplicación de mezcla; la base de datos de datos de distribución en el distribuidor almacena el historial e información diversa acerca de la duplicación de mezcla.

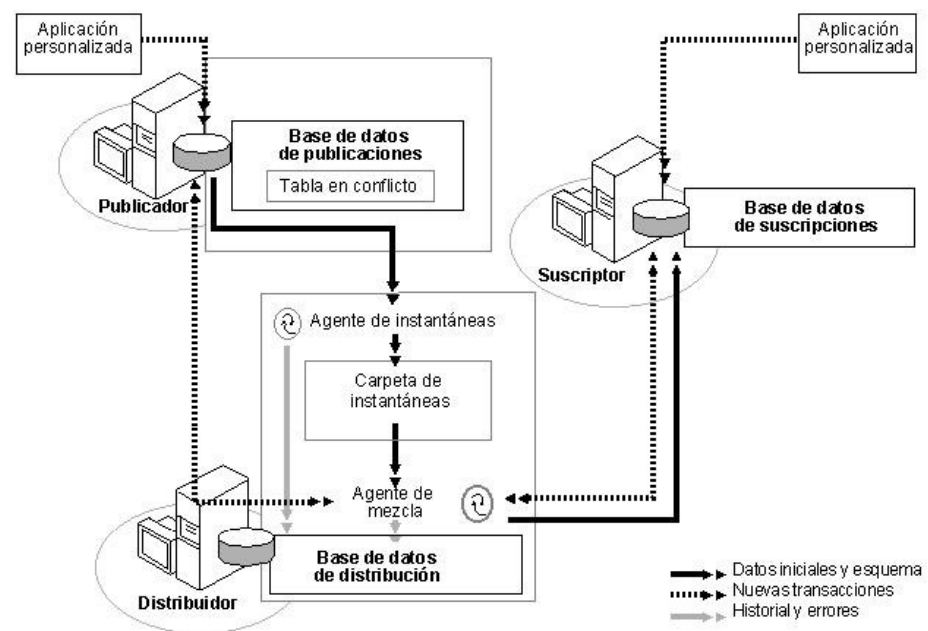


Figura II.4 Duplicación de Mezcla

Columna UNIQUEIDENTIFIER

Microsoft SQL Server identifica una columna única por cada fila de la tabla que se va a duplicar. Esto permite que la fila se pueda identificar de forma única entre varias copias de la tabla. Si la tabla ya contiene una columna con la propiedad ROWGUIDCOL con un índice único o una restricción de clave principal, SQL Server utilizará automáticamente dicha columna como identificador de fila para la tabla de publicación.

Si no es así, SQL Server agrega a la tabla de publicación una columna uniqueidentifier, titulada rowguid, que tiene la propiedad ROWGUIDCOL y un índice. Agregar la columna rowguid aumenta el tamaño de la tabla de publicación. La columna rowguid y el índice se agregan a la tabla de publicación la primera vez que el Agente de instantáneas se ejecuta para la publicación.

Desencadenadores

SQL Server instala desencadenadores que realizan un seguimiento de los cambios en los datos de cada fila o de cada columna. Los desencadenadores capturan los cambios efectuados en la tabla de publicación y los registran en tablas del sistema de mezcla. Mientras se ejecuta por primera vez el Agente de instantáneas para la publicación, en las tablas de publicación se crean desencadenadores de seguimiento. Cuando se aplica la instantánea al suscriptor, se crean desencadenadores en el suscriptor.

Se generan desencadenadores diferentes para los artículos que siguen los cambios en las filas o los que lo hacen en las columnas. Dado que

SQL Server admite varios desencadenadores del mismo tipo en la tabla de publicación, los desencadenadores de duplicación de mezcla no interfieren con los desencadenadores definidos en la aplicación.

Procedimientos almacenados

El Agente de instantáneas crea también procedimientos almacenados personalizados que actualizan la base de datos de suscripción. Hay un procedimiento almacenado personalizado para las instrucciones INSERT, uno para las instrucciones UPDATE y uno para las instrucciones DELETE. Cuando se actualizan los datos y hay que introducir nuevos registros en la base de datos de suscripción, se utilizan procedimientos almacenados personalizados, en lugar de instrucciones INSERT, UPDATE y DELETE individuales.

Tablas del sistema

SQL Server agrega entonces varias tablas del sistema a la base de datos para admitir el seguimiento de los datos, la sincronización y la detección, resolución e informe de conflictos. Por cada fila creada o cambiada, la tabla MSmerge_contents contiene la generación en la que se ha producido la modificación más reciente. También contiene la versión de la fila completa y cada uno de sus atributos. MSmerge_tombstone almacena las eliminaciones de datos dentro de una publicación. Estas tablas utilizan la columna rowguid para combinarse con la tabla de publicación.

La columna generation en estas tablas actúa como un reloj lógico que indica cuándo se actualizó por última vez una fila en un sitio determinado. Los valores datetime reales no se utilizan para indicar cuándo se han producido los cambios, ni para decidir conflictos y no hay dependencia de los relojes sincronizados entre distintos sitios. Esto hace que los algoritmos de detección y resolución de conflictos sean más resistentes a las diferencias de zona horaria y a las diferencias entre relojes físicos de diferentes servidores. En un sitio concreto, los números de generación se corresponden con el orden en el que el Agente de mezcla o los usuarios del sitio hicieron los cambios.

Hay varias columnas de seguimiento agregadas a una tabla de publicación de mezcla. Si la tabla de publicación tiene nombres de columnas reservados para el proceso de mezcla, no podrá generar una instantánea inicial debido a los nombres de columnas duplicados. Los nombres de columnas reservados son:

- ✓ reason_code
- ✓ source_object
- ✓ reason_text
- ✓ Pubid
- ✓ conflict_type
- ✓ origin_datasource
- ✓ tablenick
- ✓ create_time

La instantánea inicial y el Agente de instantáneas

Antes de que un suscriptor nuevo pueda recibir cambios incrementales desde un publicador, debe contener tablas con el mismo esquema y los mismos datos que las tablas del publicador. A la copia de la publicación actual completa desde el publicador al suscriptor se le llama aplicar la instantánea inicial. SQL Server se encarga de crear y aplicar las instantáneas, o bien pueden aplicarse manualmente.

Incluso cuando se crea suscripciones para las cuales la instantánea no se aplica automáticamente (conocidas a veces como suscripciones nosync), se continúan aplicando partes de la instantánea. En el suscriptor se crean las tablas y desencadenadores de seguimiento necesarios, lo que significa que tendrá que crear y aplicar una instantánea incluso cuando las suscripciones especifiquen que la instantánea no se aplicará automáticamente.

La duplicación de los datos modificados sólo tiene lugar después de que la duplicación de mezcla se asegure de que el suscriptor tiene la instantánea generada más reciente del esquema y los datos de la tabla. Cuando se distribuyen y se aplican instantáneas a los suscriptores, sólo se ven afectados los suscriptores que necesitan instantáneas iniciales. Los suscriptores que ya están recibiendo instrucciones INSERT, UPDATE, DELETE u otras modificaciones de los datos publicados no se ven afectados, a menos que la suscripción o la publicación estén marcadas para una reinicialización, en cuyo caso todas las suscripciones correspondientes a una publicación determinada se reinicializarán durante el siguiente proceso de mezcla.

Una tabla de suscripción sólo puede suscribirse a una publicación de mezcla a la vez. Por ejemplo, si se publica la tabla Customers en dos publicaciones y, a continuación, se suscribe a las dos publicaciones desde un Suscriptor, lo que indica que la misma base de datos de suscripciones recibirá datos de dos publicaciones. Uno de los Agentes de mezcla provocará un error durante la sincronización inicial.

La instantánea inicial puede ser una base de datos de suscripción adjunta en la duplicación de instantáneas, la duplicación transaccional y la duplicación de mezcla. Si utiliza una base de datos de suscripción que se pueda adjuntar, se copiará una base de datos de suscripción y sus suscripciones, y podrá aplicarlas a otro suscriptor.

El Agente de instantáneas implementa la instantánea inicial en la duplicación de mezcla utilizando pasos similares al Agente de instantáneas en la duplicación de instantáneas.

Instantáneas dinámicas

Las instantáneas dinámicas ofrecen un mejor rendimiento si se aplica la instantánea de una publicación de mezcla con filtros dinámicos. Al utilizar archivos de programación de copia masiva de SQL Server para aplicar datos a un suscriptor específico en lugar de utilizar series de instrucciones INSERT, mejorará el rendimiento de la aplicación de la instantánea inicial para publicaciones de mezcla filtradas dinámicamente.

Agente de mezcla

Después de aplicar la instantánea inicial a un suscriptor, los desencadenadores de SQL Server inician el seguimiento de las instrucciones INSERT, UPDATE y DELETE realizadas en el publicador y en los suscriptores.

Cada tabla que participa en la duplicación de mezcla se asigna a una zona de generación en la tabla MSmerge_articles. Cuando se actualiza una fila en una publicación de mezcla en el publicador o en los suscriptores, aunque no estén conectados, un desencadenador actualiza el valor de esa fila en la columna generation de la tabla del sistema MSmerge_contents, a la zona de generaciones adecuada para la tabla base en cuestión. Cuando el publicador y el suscriptor se vuelven a conectar y se ejecuta el Agente de mezcla, el Agente de mezcla recopila todos los cambios de fila no entregados (con nuevos valores de generación) en uno o más grupos, y asigna valores de generación más altos que los de todas las generaciones anteriores. Esto permite que el Agente de mezcla prepare lotes de cambios de las diferentes tablas en generaciones separadas y que procese estos lotes para conseguir una eficiencia mayor en las redes lentas.

El Agente de mezcla de cada sitio realiza un seguimiento del valor más alto de generación que ha enviado a cada uno de los otros sitios y del mayor valor de generación que cada uno de los otros sitios le han enviado a él. Esto proporciona puntos de partida, de forma que cada tabla pueda ser examinada sin observar los datos ya compartidos con el otro sitio. Las generaciones almacenadas en una fila concreta pueden

diferir entre los sitios, porque los números de un sitio reflejan el orden en que los cambios fueron procesados en el sitio.

Se puede limitar el número de procesos de mezcla que se ejecutan simultáneamente definiendo el parámetro `@max_concurrent_merge` de `sp_addmergepublication` o `sp_changemergepublication`. Si ya se está ejecutando el número máximo de procesos de mezcla, los nuevos procesos de mezcla esperarán su ejecución en una cola. Se puede configurar `-StartQueueTimeout` en la línea de comandos del Agente de mezcla para especificar el tiempo que debe esperar el agente a que terminen otros procesos de mezcla. Si se supera el período de `-StartQueueTimeout` y un nuevo proceso de mezcla aun está en espera, éste se detendrá y finalizará.

Sincronización

La sincronización se produce cuando los publicadores y los suscriptores de una topología de duplicación de mezcla se conectan de nuevo, los cambios se propagan entre los sitios y, si es necesario, se detectan y se resuelven los conflictos. En la sincronización, el Agente de mezcla envía todos los datos cambiados al suscriptor. Los datos fluyen desde el originador del cambio al sitio que necesita actualizarse o sincronizarse.

La dirección del intercambio controla si el Agente de mezcla carga los cambios desde el suscriptor (`ExchangeType='Upload'`), descarga los cambios en el publicador (`ExchangeType='Download'`) o ejecuta una carga seguida por una descarga (`ExchangeType='Bidirectional'`). Si se debe controlar el número de cambios aplicados, es posible configurar

los parámetros de la línea de comandos del Agente de mezcla – MaxUploadChanges y –MaxDownloadChanges. En este caso, los datos del publicador y de los suscriptores convergerán sólo cuando se propaguen todos los cambios.

En la base de datos de destino, las actualizaciones propagadas desde otros sitios se mezclan con los valores existentes, de acuerdo con las reglas de detección y resolución de conflictos. Un Agente de mezcla evalúa los valores de los datos entrantes y los actuales, y resuelve automáticamente los conflictos entre los valores antiguos y nuevos sobre la base de la resolución predeterminada, una resolución especificada al crear la publicación o una resolución personalizada. La duplicación de mezcla en SQL Server ofrece muchas resoluciones personalizadas ya preparadas que le ayudarán a implementar la lógica de negocio.

Los valores de datos que cambiaron se duplican en otros sitios y convergen con los cambios realizados en los sitios sólo cuando tiene lugar la sincronización. Las sincronizaciones pueden producirse en minutos, días o semanas y se definen en el programa del Agente de mezcla. Los datos convergen y, finalmente, todos los sitios contienen los mismos valores de datos, pero para que esto ocurra, deberá detener todas las actualizaciones y realizar mezclas entre sitios un par de veces.

El periodo de retención para suscripciones que se especifica para cada publicación controla la frecuencia con la que deberían sincronizarse el publicador y los suscriptores. Si las suscripciones no se sincronizan con el publicador dentro del periodo de retención, se marcan como ‘caducadas’ y deben reinicializarse. Esto ocurre para evitar que los

datos antiguos del suscriptor se sincronicen y carguen estos cambios en el publicador. El periodo de retención predeterminado para una publicación es de 14 días. Puesto que el Agente de mezcla limpia las bases de datos de publicación y de suscripción basándose en este valor, es preciso configurarlo con cuidado para que resulte adecuado para la aplicación.

El proceso de mezcla requiere una entrada para el publicador en la tabla `syssservers` del suscriptor. Si no existe la entrada, SQL Server intentará agregarla. Si el Agente de mezcla utiliza un inicio de sesión que no tiene acceso para agregar la entrada (como `db_owner` de la base de datos de suscripciones), se devuelve un error.

Reinicializar suscripciones

Los suscriptores de la duplicación de mezcla actualizan los datos sobre la base de la instantánea original que se les proporciona, a menos que se marque la suscripción para reinicialización. Cuando la suscripción se marca para reinicialización, el Agente de mezcla aplica una nueva instantánea al suscriptor la próxima vez que se ejecuta. Los cambios realizados en el suscriptor pueden cargarse en el publicador, opcionalmente, antes de aplicar otra vez la instantánea. Esto asegura que al reinicializar la suscripción no se pierdan los cambios realizados en los datos del suscriptor.

Si reinicializa todas las suscripciones en una publicación de mezcla, las suscripciones especificadas sin sincronización de instantánea inicial se reinicializarán del mismo modo que se reinician las suscripciones

que tienen un tipo de sincronización “automático”. Para evitar que se aplique otra vez la instantánea al suscriptor, se quita la suscripción especificada sin sincronización de instantánea inicial y a continuación se crea de nuevo, después de la reinicialización.

El Agente de mezcla es un componente del Agente SQL Server y se puede administrar directamente mediante el Administrador corporativo de SQL Server. El Agente de instantáneas y el Agente de mezcla también se pueden incrustar en aplicaciones mediante controles de Microsoft ActiveX®. El Agente de instantáneas se ejecuta en el distribuidor. Normalmente, el Agente de mezcla se ejecuta en el Distribuidor para las suscripciones de inserción, y en los Suscriptores para las suscripciones de extracción. Se puede utilizar la activación remota de agentes para descargar el proceso a otro servidor.

SQL Server puede validar los datos que se actualizan en el suscriptor durante el proceso de duplicación, para garantizar que las actualizaciones de los datos aplicadas al publicador se aplican también a los suscriptores.

Limpieza de la duplicación de mezcla

Cuando se crea la base de datos de distribución, SQL Server agrega automáticamente las siguientes tareas, para que el Agente SQL Server purgue los datos que ya no sean necesarios:

- ✓ Limpiar suscripciones en el publicador
- ✓ Limpiar el historial en el distribuidor

Estas tareas contribuyen a que la duplicación funcione de forma efectiva a largo plazo; por tanto, los administradores deben planear este mantenimiento periódico. Las tareas de limpieza eliminan la instantánea inicial de cada publicación y quitan la información del historial en la tabla Msmerge_history.

2.2 SERVICIOS WEB

2.2.1 Introducción a los Servicios Web

Debido a la gran masificación de Internet a niveles insospechables y al gran impacto causado por las tecnologías de la información en las últimas dos décadas del siglo pasado, la manera de hacer negocios y la comunicación entre las personas y las empresas cambió de una manera rotunda. Bajo este contexto se hacía cada vez mayor la necesidad de integrar y compartir información entre distintas plataformas de software y hardware.

Las empresas se percataron que era imposible crear una plataforma integrada de forma individual, así que decidieron atacar el problema de raíz. Para esto decidieron que en vez de crear la mejor plataforma integradora, era mejor buscar un lenguaje común de intercambio de información aprovechando los estándares existentes en el mercado. Bajo este contexto nacieron los Servicios Web basados en XML.

Los Servicios Web surgieron ante una necesidad de estandarizar la comunicación entre distintas plataformas (PC, Mainframe, Mac, etc.) y lenguajes de programación (PHP, C#, Java, etc.).

Anteriormente se habían realizado intentos de crear estándares pero fracasaron o no tuvieron el suficiente éxito, algunos de ellos son DCOM⁵ y CORBA⁶, por ser dependientes de la implementación del vendedor DCOM Microsoft, y CORBA ORB (a pesar que CORBA de múltiples vendedores pueden operar entre si, hay ciertas limitaciones para aplicaciones de niveles más altos en los cuales se necesite seguridad o administración de transacciones).

Otro gran problema es que se hacía uso de RPC (Remote Procedure Call) para realizar la comunicación entre diferentes nodos. Esto, además de presentar ciertos problemas de seguridad, tiene la desventaja de que su implementación en un ambiente como es Internet, es casi imposible (muchos firewalls bloquean este tipo de mensajes, lo que hace prácticamente imposible a dos computadoras conectadas por Internet comunicarse).

Los Web Services surgieron para finalmente poder lograr la tan esperada comunicación entre diferentes plataformas. En la actualidad muchos sistemas legacy están pasando a ser web services.

⁵ DCOM: Distributed Component Object Model, un sistema de Microsoft. Componentes de un software para comunicarse con computadoras en línea.

⁶ CORBA: es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos.

Es por esto que en 1999 se comenzó a plantear un nuevo estándar, el cual terminaría utilizando XML, SOAP, WSDL, y UDDI.

¿Qué es un Servicio Web?

Según la W3C⁷ define "Servicio web" como un sistema de software diseñado para permitir interoperabilidad máquina a máquina en una red. En general, los servicios web son sólo APIs Web que pueden ser accedidas en una red, como internet, y ejecutadas en un sistema de hosting remoto.

Según la W3C Es un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web.

Un servicio web es un servicio, con un interfaz definido y conocido, al que se puede acceder a través de internet. Igual que una página web está definida por un URL (Uniform Resource Locator), un servicio web está definido por un URI (Uniform Resource Identification) y por su interfaz, a través del cual se puede acceder a él.

⁷ W3C: El Consorcio World Wide Web (W3C) es una comunidad internacional donde las organizaciones Miembro [ingles] [inglés], personal [ingles] [inglés] a tiempo completo y el público en general trabajan conjuntamente para desarrollar estándares Web [ingles] [inglés]

Los web services son componentes software que permiten a los usuarios usar aplicaciones de negocio que comparten datos con otros programas modulares, vía Internet. Son aplicaciones independientes de la plataforma que pueden ser fácilmente publicadas, localizadas e invocadas mediante protocolos web estándar, como XML, SOAP, UDDI o WSDL.

Un servicio web consiste en una función disponible en un servidor conectado al web. Esta función puede consistir en cualquier cosa:

- ✓ Realizar un simple cálculo con unos datos que se le envían como parámetro,
- ✓ Acceder a una base de datos para recuperar un conjunto de registros,
- ✓ Validar la corrección de una información o contrastarla frente a otros datos, etc.

El servicio web podrá ser solicitado desde otro programa informático que se ejecute en un ordenador conectado al web. Junto a la solicitud de la ejecución, se pueden enviar al ordenador que ofrece el servicio unos parámetros que el servicio web remoto tomará como base para el cálculo o la función.

2.2.2 Tecnologías base de servicios Web

Introducción a las tecnologías básicas

Un servicio Web es una colección de funciones que son empaquetadas como una sola entidad y publicadas a la red para el uso a través de otros programas. Los servicios Web son bloques componentes para crear sistemas distribuidos abiertos y permiten a las compañías e individuos acceder rápidamente a los recursos digitales disponibles mundialmente.

La base de Servicios Web es la mensajería de XML sobre los protocolos del Web estándar como HTTP. Éste es un mecanismo de comunicación muy ligero en el que cualquier lenguaje de programación, middleware o plataforma pueden participar, suavizando la interoperabilidad grandemente.

Las tecnologías básicas que utilizan los servicios web son:

XML

XML fue desarrollado para superar las limitaciones de HTML, especialmente para soportar la creación y manejo del contenido dinámico. HTML está bien para definir y mantener el contenido estático, pero cuando el Web evoluciona hacia una plataforma de software activado, en la que los datos tienen significado asociado, el contenido necesita ser generado y dirigido dinámicamente. Usando XML, se puede definir cualquier número de elementos para el significado asociado con datos; es decir, describe los datos usando uno o más elementos creados para ese propósito.

SOAP

La especificación de SOAP define una estructura de la mensajería para intercambiar los datos de XML formateados en Internet. La estructura de la mensajería es simple, fácil de desarrollar y completamente neutral con respecto al sistema operativo, lenguaje de programación o la plataforma de computación distribuida.

SOAP es fundamentalmente un modelo de comunicación unidireccional que asegura que un mensaje coherente sea transferido desde un remitente a un destinatario, inclusive potencialmente intermediarios que pueden procesar parte de o agregar a la unidad del mensaje. La especificación de SOAP contiene las convenciones de adaptar la mensajería unidireccional para el paradigma típico de petición/respuesta en comunicación estilo RPC y además define cómo transmitir los documentos XML completos.

SOAP define una regla de codificación optativa para los tipos de datos, pero en los puntos terminales en una comunicación de SOAP puede decidirse por sus propias reglas de codificación a través de acuerdo privado. La comunicación usa a menudo la codificación nativa XML.

SOAP está diseñado para proporcionar un protocolo de comunicaciones independiente, abstracto capaz de pontear o conectando, dos o más negocios o dos o más sitios comerciales remotos. Los sistemas conectados pueden ser construidos usando cualquier combinación de hardware y software que soporten el acceso a Internet para los sistemas existentes, como .NET y J2EE.

Los sistemas existentes típicamente también representan infraestructuras múltiples y productos del paquete de software. SOAP y el resto de la estructura de XML proveen los medios a cualquier dos o más sitios comerciales, mercados o países que comercian entre sí para convenir en un contacto común para exponer sus servicios al Web.

WSDL

El Lenguaje de descripción de servicios Web (WSDL) es un formato de esquema XML que define una estructura extensible por describir las interfaces de los servicios Web. WSDL fue desarrollado principalmente por Microsoft e IBM y fue sometido por compañías al W3C. WSDL tiene el núcleo de la estructura de los servicios Web, proporcionando un modo común para representar los tipos de datos proporcionados en los mensajes, las operaciones a ser realizadas en los mensajes y el mapeo de los mensajes sobre el transporte de la red.

WSDL es dividido en tres elementos mayores:

- ✓ Definiciones de tipo de datos.
- ✓ Definiciones abstractas.
- ✓ Servicio de ligaciones.

Cada elemento mayor puede especificarse en un fragmento de documento XML separado y puede importarse en varias combinaciones para crear una descripción final de los servicios Web o ellos pueden todos ser definidos juntos en un solo documento. Las definiciones del

tipo de datos determinan la estructura y el contenido de los mensajes. Las definiciones abstractas determinan las operaciones realizadas en el contenido del mensaje y el servicio de ligaciones determina el transporte de la red que llevará el mensaje a su destino.

UDDI

Después de que se han definido los datos en los mensajes (XML), descrito los servicios que recibirán y procesarán el mensaje (WSDL) y se han identificado los medios de enviar y recibir los mensajes (SOAP), necesitan una manera para publicar el servicio que ofrecen y encontrar los servicios que otros ofrecen y que desean usar. Ésta es la función que proporciona UDDI (distribución universal, descubrimiento e integración).

La estructura de UDDI define un modelo de datos en XML e interfaces de programación de aplicaciones de SOAP (APIs) para registrar y descubrir la información comercial, inclusive los servicios Web en publicación de negocios.

UDDI es producido por un consorcio independiente de vendedores, fundado por Microsoft, IBM y Ariba, para el desarrollo de un estándar de Internet en el registro de descripción y descubrimiento de servicios Web.

UDDI es similar en concepto a un directorio de las páginas amarillas. Los negocios registran su contacto de información, incluyendo cosas en detalle como números de teléfono y fax, código postal y sitio Web. El registro incluye la información de la categoría por buscar, como ubicación geográfica, razón social de la industria, tipo de negocio

mercantil, y así sucesivamente. Otros negocios pueden buscar la información registrada en UDDI para encontrar suministro de partes, servicios de comida, compraventas o actividades comerciales. Un negocio también puede descubrir la información sobre el servicio Web específico en el registro, encontrando típicamente una URL para un archivo de WSDL que señala el servicio Web de un distribuidor.

2.2.3 Descripción de las tecnologías de los Servicios Web

Los servicios web se definen a partir de las siguientes especificaciones:

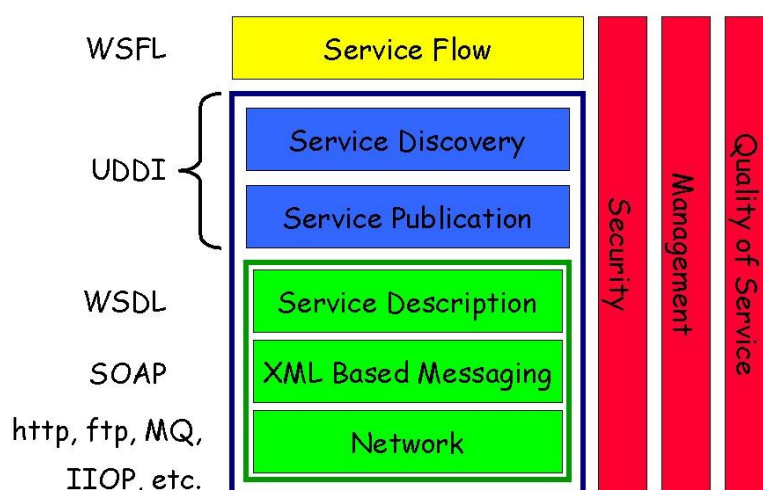


Figura II.5 Estándares para servicios Web

2.2.3.1 SOAP (Simple Object Access Protocol):

La especificación SOAP indica cómo se deben codificar los mensajes que circularán entre las dos aplicaciones.

Fue definido inicialmente por Microsoft, Userland Software y DevelopMentor, a día de hoy se trata de una especificación mantenida por el W3C que cuenta con el apoyo de otros fabricantes como IBM, HP, Oracle, etc.

La especificación SOAP define dos modelos de mensajes:

Un mensaje que se enviará desde la aplicación cliente a la aplicación servidor, solicitando la ejecución de un método al que se pasan una serie de parámetros.

Un mensaje que se enviará desde la aplicación servidor a la cliente, y que contendrá datos XML con los resultados de la ejecución del método solicitado.

En el núcleo de los servicios Web se encuentra el protocolo simple de acceso a datos SOAP, que proporciona un mecanismo estándar de empaquetar mensajes. SOAP ha recibido gran atención debido a que facilita una comunicación del estilo RPC entre un cliente y un servidor remoto. Pero existen multitud de protocolos creados para facilitar la comunicación entre aplicaciones, incluyendo RPC de Sun, DCE de Microsoft, RMI de Java y ORPC de CORBA.

SOAP es el primer protocolo de su tipo que ha sido aceptado prácticamente por todas las grandes compañías de software del mundo. Compañías que en raras ocasiones cooperan entre sí están ofreciendo

su apoyo a este protocolo. Algunas de las mayores Compañías que soportan SOAP son Microsoft, IBM, SUN, Microsystems, SAP y Ariba.

Ventajas:

- ✓ No está asociado con ningún lenguaje: los desarrolladores involucrados en nuevos proyectos pueden elegir desarrollar con el último y mejor lenguaje de programación que exista pero los desarrolladores responsables de mantener antiguas aflicciones heredadas podrían no poder hacer esta elección sobre el lenguaje de programación que utilizan.
- ✓ SOAP no especifica una API, por lo que la implementación de la API se deja al lenguaje de programación, como en Java, y la plataforma como Microsoft .Net.
- ✓ No se encuentra fuertemente asociado a ningún protocolo de transporte: La especificación de SOAP no describe como se deberían asociar los mensajes de SOAP con HTTP. Un mensaje de SOAP no es más que un documento XML, por lo que puede transportarse utilizando cualquier protocolo capaz de transmitir texto.
- ✓ No está atado a ninguna infraestructura de objeto distribuido La mayoría de los sistemas de objetos distribuidos se pueden extender.
- ✓ Aprovecha los estándares existentes en la industria: Los principales contribuyentes a la especificación SOAP evitaron, intencionadamente, reinventar las cosas. Optaron por extender los estándares existentes para que coincidieran con sus necesidades.

Por ejemplo, SOAP aprovecha XML para la codificación de los mensajes, en lugar de utilizar su propio sistema de tipo que ya están definidas en la especificación esquema de XML. Y como ya se ha mencionado SOAP no define un medio de transporte de los mensajes; los mensajes de SOAP se pueden asociar a los protocolos de transporte existentes como HTTP y SMTP.

- ✓ Permite la interoperabilidad entre múltiples entornos: SOAP se desarrolló sobre los estándares existentes de la industria, por lo que las aplicaciones que se ejecuten en plataformas con dichos estándares pueden comunicarse mediante mensaje SOAP con aplicaciones que se ejecuten en otras plataformas. Por ejemplo, una aplicación de Servicios Web en plataforma .NET de escritorio que se ejecute en una PC puede comunicarse con una aplicación del backend ejecutándose en un mainframe capaz de enviar y recibir XML sobre HTTP.

Anatomía de un mensaje de SOAP

SOAP proporciona un mecanismo estándar de empaquetar un mensaje. Un mensaje SOAP se compone de un sobre que contiene el cuerpo del mensaje y cualquier información de cabecera que se utiliza para describir el mensaje.


```

<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <!--Optional header information goes here. -->
    <To>Scott</To>
    <From>Suzanne</From>
  </soap:Header>
  <soap:Body>
    <!--Message goes here. -->
    Please pick up some milk on your way home from work.
  </soap:Body>
</soap:Envelope>

```

Figura II.6 Anatomía de un Mensaje SOAP

El elemento raíz del documento es el elemento Envelope. La figura contiene dos subelementos, Body y Header. Un ejemplo de SOAP válido también puede contener otros elementos hijo en el sobre.

El sobre puede contener un elemento Header opcional que contiene información sobre el mensaje. En la figura anterior, la cabecera contiene dos elementos que describen a quien compuso el mensaje, y posible receptor del mismo.

El sobre debe contener un elemento body. El elemento body (cuerpo) contiene la carga de datos del mensaje. En la figura el cuerpo contiene una simple cadena de caracteres.

Un mensaje debe estar dentro de sobre de SOAP bien construido. Un sobre se compone de un único elemento envelope el sobre puede contener un elemento Header y puede contener un elemento body. Si existe, la cabecera debe ser el elemento hijo inmediato del sobre, con el cuerpo siguiendo inmediatamente a la cabecera.

El cuerpo contiene la carga de datos del mensaje y la cabecera contiene los datos adicionales que no pertenecen necesariamente al cuerpo del mensaje.

Además de definir un sobre de SOAP, la especificación de SOAP define una forma de codificar los datos contenidos en un mensaje. La codificación de SOAP proporciona un mecanismo estándar para serializar tipos de datos no definidos en la especificación del esquema de XML.

La especificación de SOAP también proporciona un patrón de mensaje estándar para facilitar el comportamiento de tipo RPC. Se emparejan dos mensajes de SOAP para facilitar la asociación de un mensaje de petición con un mensaje de respuesta.

La llamada a un método y sus parámetros se serializan en el cuerpo del mensaje de petición en forma de una estructura.

El elemento raíz tiene el mismo nombre que el método objetivo, con cada uno de los parámetros codificado como un subelemento.

El mensaje de respuesta puede contener los resultados de la llamada al método o una estructura de fallo bien definida. Los resultados de la llamada a un método se serializan en el cuerpo de la petición como una estructura. Por convenio, el elemento raíz tiene el mismo nombre que el

método original al que se añade result. Los parámetros de retorno se serializan como elementos hijo, con el parámetro de retorno en primer lugar. Si se encuentra un error el cuerpo del mensaje de respuesta contendrá una estructura de fallo bien definida.

```

Example (Servicio GetWeather –ver web–)
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <GetWeather xmlns="http://www.webserviceX.NET">
      <CityName>Valencia</CityName>
      <CountryName>Spain</CountryName>
    </GetWeather>
  </soap:Body>
</soap:Envelope>

Example (Contenido de la respuesta)
<string>
  <CurrentWeather>
    <Location>Valencia / Aeropuerto, Spain (LEVC) 39-30N 000-28W 62M</Location>
    <Time>Jun 14, 2007 - 10:00 AM EDT / 2007.06.14 1400 UTC</Time>
    <Wind> from the SW (230 degrees) at 21 MPH (18 KT):0</Wind>
    <Visibility> greater than 7 mile(s):0</Visibility>
    <SkyConditions> mostly clear</SkyConditions>
    <Temperature> 91 F (33 C)</Temperature>
    ...
  </CurrentWeather>
</string>

Example (Estructura de la respuesta)
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetWeatherResponse xmlns="http://www.webserviceX.NET">
      <GetWeatherResult> string </GetWeatherResult>
    </GetWeatherResponse>
  </soap:Body>
</soap:Envelope>

```

Figura II.7 Ejemplo de un mensaje SOAP

2.2.3.2 WSDL (Web Services Description Language)

Permite describir los distintos métodos o funciones que están disponibles en un servicio web, así como su signatura, es decir, el número de argumentos o parámetros que se les debe pasar, y el tipo de dato que devolverá la función como resultado. Se establece una equivalencia entre el documento WSDL y un “contrato” que especifica los servicios que el servidor se compromete a ofrecer al cliente, siempre que éste los solicite de la forma adecuada

El lenguaje de descripción de servicios Web (WSDL, Web Service Description Language) es un dialecto basado en XML sobre el esquema que describe un servicio Web. Un documento WSDL proporciona la información necesaria al cliente para interactuar con el servicio Web. WSDL es extensible y se puede utilizar para describir, prácticamente, cualquier servicio de red, incluyendo SOAP sobre HTTP.

Dado que los protocolos de comunicaciones y los formatos de mensajes están estandarizados en la comunidad del Web, cada día aumenta la posibilidad e importancia de describir las comunicaciones de forma estructurada. WSDL afronta esta necesidad definiendo una gramática XML que describe los servicios de red como colecciones de puntos finales de comunicación capaces de intercambiar mensajes. Las definiciones de servicio de WSDL proporcionan documentación para sistemas distribuidos y sirven como fórmula para automatizar los detalles que toman parte en la comunicación entre aplicaciones.

Los documentos WSDL definen los servicios como colecciones de puntos finales de red o puertos. En WSDL, la definición abstracta de puntos finales y de mensajes se separa de la instalación concreta de red o de los enlaces del formato de datos. Esto permite la reutilización de definiciones abstractas: mensajes, que son descripciones abstractas de los datos que se están intercambiando y tipos de puertos, que son colecciones abstractas de operaciones. Las especificaciones concretas del protocolo y del formato de datos para un tipo de puerto determinado constituyen un enlace reutilizable.

Un puerto se define por la asociación de una dirección de red y un enlace reutilizable; una colección de puertos define un servicio. Por esta razón, un documento WSDL utiliza los siguientes elementos en la definición de servicios de red.

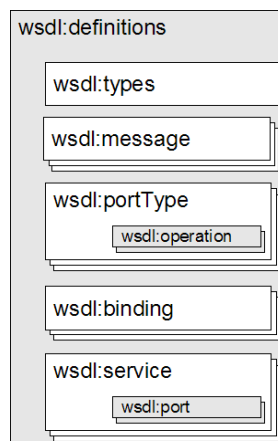


Figura II.8 Esquema de un mensaje WSDL

Elemento types

El elemento Types contiene información de esquema referenciado en el documento WSDL. El sistema de tipos predeterminado que admite WSDL es de esquema de XML. Si se usa esquema de XML para definir los tipos que contiene el elemento Types el elemento schema aparecerá inmediatamente como elemento hijo.

Se pueden utilizar otros sistemas de tipos por extensión. Si se desea, utilizar otro sistema de tipo puede aparecer un elemento de extensibilidad bajo el elemento Types. El nombre de este elemento debería identificar el sistema de tipos utilizados. El siguiente ejemplo muestra los tipos de datos usados en los mensajes en el WSDL del buscador de google.

```

<types>
  <xsd:schema xmlns="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:GoogleSearch">

    <xsd:complexType name="GoogleSearchResult">

      <xsd:all>
        <xsd:element name="documentFiltering" type="xsd:boolean"/>
        <xsd:element name="searchComments" type="xsd:string"/>
        <xsd:element name="estimatedTotalResultsCount" type="xsd:int"/>
        <xsd:element name="estimateIsExact" type="xsd:boolean"/>
        <xsd:element name="resultElements"
type="typens:ResultElementArray"/>
        <xsd:element name="searchQuery" type="xsd:string"/>
        <xsd:element name="startIndex" type="xsd:int"/>
        <xsd:element name="endIndex" type="xsd:int"/>

        <xsd:element name="searchTips" type="xsd:string"/>
        <xsd:element name="directoryCategories"
type="typens:DirectoryCategoryArray"/>
        <xsd:element name="searchTime" type="xsd:double"/>
      </xsd:all>
    </xsd:complexType>

    <xsd:complexType name="ResultElement">
      <xsd:all>
        <xsd:element name="summary" type="xsd:string"/>
        <xsd:element name="URL" type="xsd:string"/>
        <xsd:element name="snippet" type="xsd:string"/>
        <xsd:element name="title" type="xsd:string"/>
        <xsd:element name="cachedSize" type="xsd:string"/>
        <xsd:element name="relatedInformationPresent" type="xsd:boolean"/>
        <xsd:element name="hostName" type="xsd:string"/>
        <xsd:element name="directoryCategory"
type="typens:DirectoryCategory"/>
        <xsd:element name="directoryTitle" type="xsd:string"/>
      </xsd:all>
    </xsd:complexType>

    <xsd:complexType name="ResultElementArray">
      <xsd:complexContent>
        <xsd:restriction base="soapenc:Array">
          <xsd:attribute ref="soapenc:arrayType"
wsdl:arrayType="typens:ResultElement[]"/>
        </xsd:restriction>
      </xsd:complexContent>
    </xsd:complexType>
  </xsd:schema>

```

```

    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="DirectoryCategoryArray">
    <xsd:complexContent>
      <xsd:restriction base="soapenc:Array">
        <xsd:attribute ref="soapenc:arrayType"
wsdl:arrayType="typens:DirectoryCategory[]" />
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="DirectoryCategory">

    <xsd:all>
      <xsd:element name="fullViewableName" type="xsd:string"/>
      <xsd:element name="specialEncoding" type="xsd:string"/>
    </xsd:all>
  </xsd:complexType>

</xsd:schema>
</types>

```

Elemento message

El elemento Message proporciona una abstracción común para el paso de mensajes entre el cliente y el servidor. Como puede utilizar múltiples formatos de definición de esquema en documentos WSDL es necesario disponer de un mecanismo común de identificar los mensajes. El elemento Message proporciona este nivel común de abstracción al que se hará referencia en otras partes del documento WSDL.

Puede Aparecer, y normalmente aparecerán, múltiples elementos Message en un documento WSDL, uno para cada mensaje que se comunica entre el cliente y el servidor. Cada mensaje contiene uno o más elementos "Part" que describen las piezas del contenido del

mensaje. Un ejemplo de una parte es el cuerpo de un mensaje de SOAP o un parámetro que forma parte de una cadena de petición, un parámetro codificado en el cuerpo del mensaje de SOAP o todo el cuerpo de un mensaje de SOAP. El siguiente ejemplo define los elementos del mensaje usados en el WSDL del buscador de google.

```

<message name="doGetCachedPage">

    <part name="key"           type="xsd:string"/>
    <part name="url"          type="xsd:string"/>
</message>

<message name="doGetCachedPageResponse">
    <part name="return"       type="xsd:base64Binary"/>
</message>

<message name="doSpellingSuggestion">

    <part name="key"           type="xsd:string"/>

    <part name="phrase"       type="xsd:string"/>
</message>

<message name="doSpellingSuggestionResponse">
    <part name="return"       type="xsd:string"/>
</message>

<! note, ie and oe are ignored by server; all traffic is UTF8. >

<message name="doGoogleSearch">

    <part name="key"           type="xsd:string"/>
    <part name="q"             type="xsd:string"/>
    <part name="start"         type="xsd:int"/>
    <part name="maxResults"    type="xsd:int"/>
    <part name="filter"        type="xsd:boolean"/>
    <part name="restrict"      type="xsd:string"/>
    <part name="safeSearch"    type="xsd:boolean"/>
    <part name="lr"           type="xsd:string"/>

```



```
<part name="ie"                type="xsd:string"/>

<part name="oe"                type="xsd:string"/>
</message>

<message name="doGoogleSearchResponse">
  <part name="return"          type="typens:GoogleSearchResult"/>
</message>
```

Elemento portType

El elemento portType contiene un conjunto de operaciones abstractas que representan los tipos de correspondencia que pueden producirse entre el cliente y el servidor.

Un tipo puerto se compone de un conjunto de electos operation que define una determinada acción. Los electos operation se componen de mensajes definidos en el documento WSDL.

WSDL define cuatro tipos de operaciones denominadas tipo operaciones:

- ✓ Requestresponse(peticiónrespuesta) comunicación del tipo RPC en la que le cliente realiza una petición y el servidor envía la correspondiente respuesta.
- ✓ Oneway (un sentido) Comunicación del estilo documento en la que el cliente envía un mensaje pero no recibe una respuesta del servidor indicando el resultado del mensaje procesado.

- ✓ Solicitresponse(solicitudrespuesta) La contraria a la operación peticiónrespuesta. El servidor envía una petición y el cliente le envía de vuelta una respuesta.
- ✓ Notification (Notificación) La contraria a la operación unsentido el servidor envía una comunicación del estilo documento al cliente.

El siguiente ejemplo define las operaciones permitidas y los mensajes intercambiados en el servicio.

```
<portType name="GoogleSearchPort">

  <operation name="doGetCachedPage">
    <input message="typens:doGetCachedPage"/>
    <output message="typens:doGetCachedPageResponse"/>
  </operation>

  <operation name="doSpellingSuggestion">
    <input message="typens:doSpellingSuggestion"/>
    <output message="typens:doSpellingSuggestionResponse"/>
  </operation>

  <operation name="doGoogleSearch">
    <input message="typens:doGoogleSearch"/>
    <output message="typens:doGoogleSearchResponse"/>
  </operation>

</portType>
```

Elemento binding

El elemento binding contiene las definiciones de la asociación de un protocolo como SOAP a un determinado bindingType. Las definiciones binding especifican detalles de formatos del mensaje y el protocolo. Por

ejemplo, la información de asociación especifica si se puede acceder a una instancia de un portType de forma RPC.

Las definiciones binding también indican el número de comunicaciones de red que se requieren para realizar una determinada acción. Por ejemplo, una llamada RPC de SOAP sobre HTTP podría involucrar un intercambio de comunicación HTTP, pero esa misma llamada sobre SMTP podría involucrar dos intercambios de comunicaciones de SMTP discretas.

La asociación se logra utilizando elementos de extensión. Cada protocolo tiene su propio conjunto de elementos de extensión para especificar los detalles del protocolo y el formato de los mensajes. Para un determinado protocolo los elementos de extensión se suelen utilizar para decorar las acciones individuales de una operación y la propia operación con la información de asociación del protocolo. A veces los elementos de extensión se utilizan en el propio nivel portType. Las especificaciones de protocolos de comunicación usados se muestran en siguiente ejemplo.

```
<binding name="GoogleSearchBinding" type="typens:GoogleSearchPort">
  <soap:binding style="rpc"
    transport="http://schemas.xmlsoap.org/soap/http"/>

  <operation name="doGetCachedPage">
    <soap:operation soapAction="urn:GoogleSearchAction"/>
    <input>
      <soap:body use="encoded"
        namespace="urn:GoogleSearch"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </input>
  </operation>
</binding>
```

```

</input>
<output>

  <soap:body use="encoded"
    namespace="urn:GoogleSearch"
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
</output>
</operation>

<operation name="doSpellingSuggestion">
  <soap:operation soapAction="urn:GoogleSearchAction"/>
  <input>
    <soap:body use="encoded"
      namespace="urn:GoogleSearch"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
</input>

  <output>
    <soap:body use="encoded"
      namespace="urn:GoogleSearch"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
</output>
</operation>

<operation name="doGoogleSearch">
  <soap:operation soapAction="urn:GoogleSearchAction"/>
  <input>
    <soap:body use="encoded"
      namespace="urn:GoogleSearch"
      encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

</input>
<output>
  <soap:body use="encoded"
    namespace="urn:GoogleSearch"
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
</output>
</operation>
</binding>

```

Elemento service

Un servicio es un grupo de puertos relacionados y se definen en el elemento `service`. Un puerto es un extremo concreto de un Servicio Web al que se hace referencia por una dirección única.

Los puertos que se definen en determinado servicio son independientes. Por ejemplo, la salida de un puerto que no puede utilizarse como una entrada de otro. El siguiente ejemplo muestra el conjunto de puertos y dirección de los mismos usados en el WSDL del buscador de google.

```
<service name="GoogleSearchService">  
  
  <port name="GoogleSearchPort" binding="typens:GoogleSearchBinding">  
    <soap:address location="http://api.google.com/search/beta2"/>  
  </port>  
</service>
```

Elementos de Extensibilidad

Los elementos de extensibilidad se utilizan para representar determinadas tecnologías. Por ejemplo, se puede utilizar los elementos de extensibilidad para especificar el idioma en que se utiliza en el esquema de los elementos types.

El esquema para un determinado conjunto de elementos de extensibilidad se debe definir dentro de distintos espacios de nombres que WSDL. La definición de los propios elementos puede contener un atributo `wsdl:required` que indique un valor boolean si el atributo `required` se establece a `true` en una definición de elementos una

asociación que haga referencia a ese conjunto concreto de electos de extensibilidad tiene que incluir dicho elemento.

Lo más habitual es que los elementos de extensibilidad se utilicen para especificar especificación de asociación. La especificación WSDL define conjunto de elementos de extensibilidad para la asociación SOAP, HTTP GET, HTTP POS, MIME. Sin embargo, la especificación sólo define las asociaciones para dos de los cuatro tipos de operaciones. Un sentido y petición respuesta.

2.2.3.3 UDDI (Universal Description Discovery and Integration)

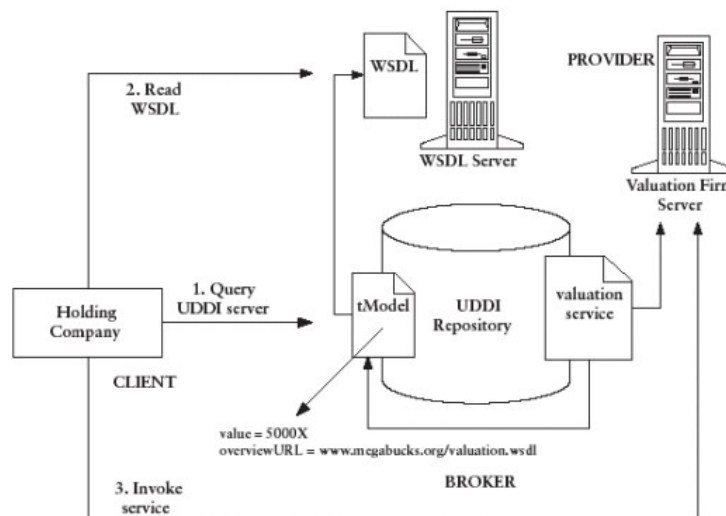


Figura II.9 Localización del servicio web

UDDI posiblemente sea el estándar más joven y de desarrollo más rápido. Descripción, descubrimiento e integración universal, es una iniciativa de Microsoft diseñada para facilitar la localización de servicios web sobre cualquier servidor.

La finalidad de UDDI, es proporcionar un almacén donde las empresas puedan exponer todos los servicios web que tengan. Por ejemplo, una compañía podría listar los servicios que tiene para intercambiar documentos, que describen cómo se deben enviar los pedidos y seguir la pista a la información que se recupera. Para enviar esta información, una empresa debe registrar este servicio.

Básicamente UDDI está concebida como:

- ✓ Una especificación técnica para construir un directorio distribuido de negocios y servicios Web. Datos que son almacenados dentro de un formato de XML específico y detalles de APIs para buscar datos existentes y publicación de nuevos datos.
- ✓ Un registro de negocios que es una implementación totalmente operacional de la especificación de UDDI. El registro de UDDI permite ahora a cualquiera buscar los datos de UDDI existentes.

Registro UDDI

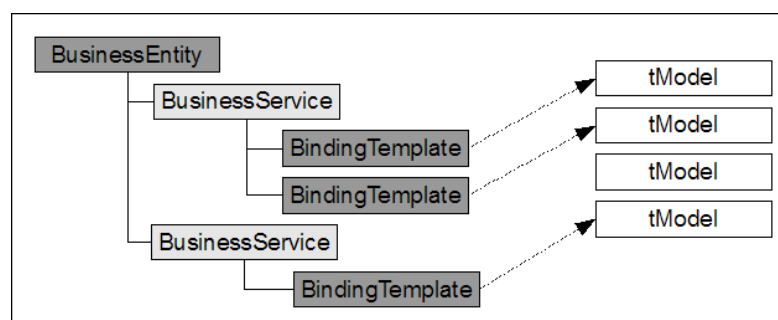


Figura II.10 Datos del Registro UDDI

El registro UDDI puede clasificarse en dos grupos:

- ✓ Públicos: Es un directorio global de descripciones de servicios de negocios internacional. Las instancias de este registro son organizadas por operadores de nodo (son grandes corporaciones como IBM y Microsoft) en una serie de servidores de UDDI dedicados. Se reproducen archivos de UDDI automáticamente entre las instancias del registro. Algunas compañías también actúan como archivistas de UDDI, permitiendo a otros agregar y editar sus perfiles de descripción de servicio Web. El registro público de negocios es complementado por varios mercados de servicio que ofrecen servicios Web genéricos para venta o renta.
- ✓ Privados: Son almacenes de descripciones de servicio organizados dentro de una corporación o empresa (ver figura 1). Los que autorizan el acceso a este directorio pueden incluir la selección de socios de negocio externos. Un registro restringido sólo para usuarios internos puede ser referido como un registro interno.

El registro UDDI define un interface completamente programable que define cómo se pueden utilizar mensajes SOAP para recuperar información sobre una empresa o registrar los servicios web de una empresa.

Categorías UDDI

Un registro de UDDI contiene las entradas sobre los negocios, los servicios que estos negocios proporcionan, e información en cómo a esos servicios pueden tenerse acceso. El directorio de UDDI tiene tres categorías:

- ✓ Páginas blancas. Contienen información básica sobre un proveedor de servicios, incluso el nombre del proveedor, descripción textual del negocio información del contacto y otros identificadores exclusivos.
- ✓ Páginas amarillas. Incluyen la clasificación basada en taxonomías estándares del proveedor de servicio o compañía registrada.
- ✓ Páginas Verdes. Contienen las entradas técnicas para los servicios Web. Generalmente esto incluye un apuntador a la especificación externa y una dirección en la que invocar el servicio.

Publicación y consulta UDDI

La publicación de un servicio Web para un registro UDDI y la búsqueda de un registro UDDI para descubrir un servicio Web pueden realizarse de dos maneras alternativas:

- ✓ Con medios de interfaz basada en Web, los cuales son proporcionados por proveedores como IBM y Microsoft. Estas interfases basadas en Web habilitan trabajar de forma interactiva siguiendo los pasos necesarios para la publicación y búsqueda de servicios Web.
- ✓ Con medios de programación usando las APIs (Application Programming Interfaces) de UDDI, las cuales se dividen en dos tipos, APIs de publicación (publish) y APIs de consulta (inquiry).

2.3 ADO.NET COMO TECNOLOGIA PARA EL ACCESO A DATOS

La mayor parte de aplicaciones que se han desarrollado giran en torno a la lectura y actualización de información de base de datos, necesitando de algún mecanismo que permita el acceso a los datos. El modo de comunicación entre nuestra aplicación y la base de datos implica que ambos manejen un lenguaje de programación común, para resolver este problema de comunicación es que se usa un lenguaje común de bases de datos que tanto los lenguajes de programación existentes como las bases de datos entienden, este lenguaje común de bases de datos es el SQL (Structured Query Language) o lenguaje estructurado de consultas.

La tecnología ADO.NET que es parte del .Net Framework, proporcionan acceso coherente a distintos orígenes de datos como Microsoft SQL Server, así como a orígenes de datos expuestos mediante OLE DB y XML, lo que permite la integración de datos en aplicaciones distribuidas y escalables.

La interfaz ADO.NET está específicamente diseñada para aplicaciones Web basadas en mensajes, proporciona también funcionalidad aplicable a otras arquitecturas de aplicaciones. ADO.NET admite el acceso a datos de correspondencia imprecisa y así permite maximizar el uso compartido de datos mediante la reducción del número de conexiones activas con la base de datos; de este modo, disminuye la probabilidad de

que varios usuarios compitan entre sí por los recursos limitados del servidor de base de datos.⁸

2.3.1 ARQUITECTURA DE ADO .NET

Los componentes principales de una aplicación ADO.NET (DataSet y Proveedores Administrados) se muestran en la siguiente ilustración.

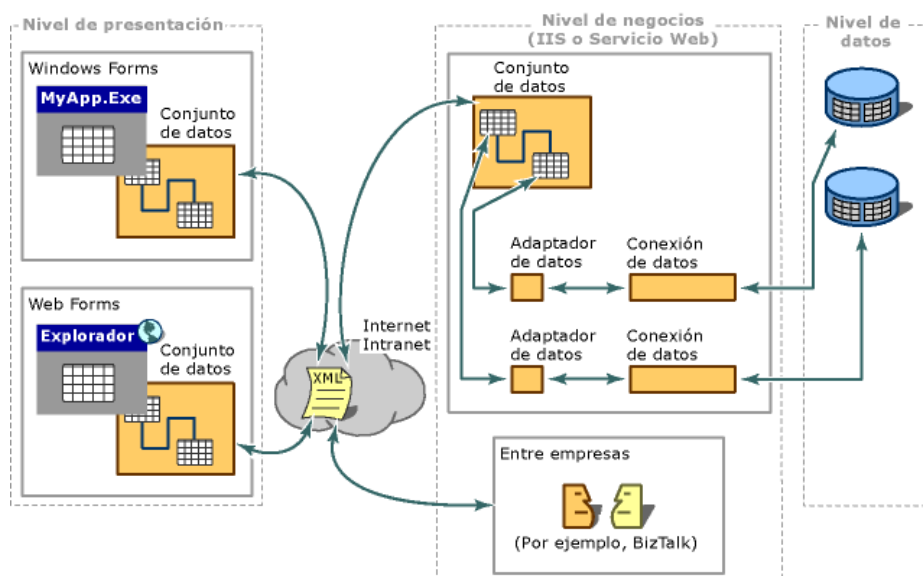


Figura II. 11. Componentes ADO.net⁹

Los componentes de ADO.NET están diseñados para separar el acceso a datos de la manipulación de datos. ADO.NET tiene dos componentes principales que cumplen esta función: el DataSet y el proveedor de datos de .NET Framework, que es un conjunto de componentes entre los que se incluyen los objetos Connection, Command, DataReader y DataAdapter.

⁸ **Referencia:** Esta información fue tomada de MSDN Library en español.

En <http://msdn.microsoft.com/es-es/library/aa290751%28v=vs.71%29.aspx>

⁹ **BIZTALK.**- BizTalk Server 2004 es un producto de Windows Server System que permite a los clientes integrar sistemas, empleados y asociados comerciales con más eficacia y rapidez que nunca.

En el siguiente diagrama se ilustra los componentes de la arquitectura de ADO.NET

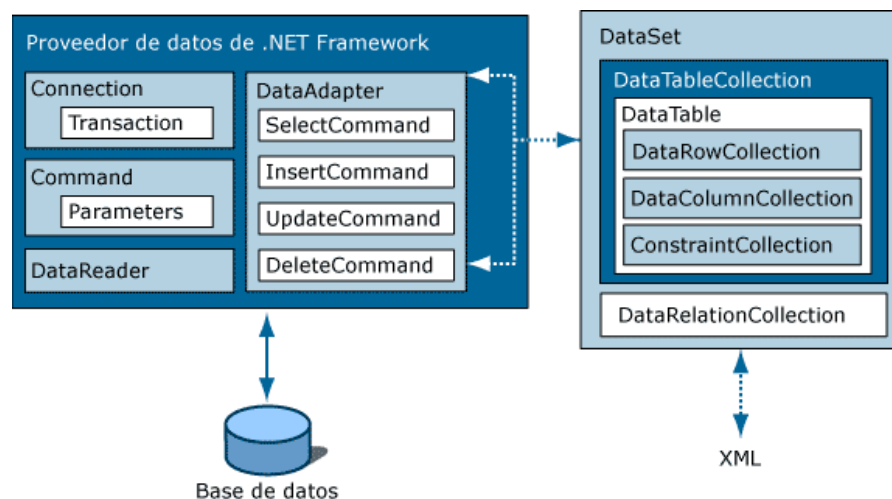


Figura II. 12. Arquitectura de ADO.net

El **DataSet de ADO.NET** es el componente central de la arquitectura sin conexión de ADO.NET. El DataSet está expresamente diseñado para el acceso a datos independientemente del origen de datos. Como resultado, se puede utilizar con múltiples y distintos orígenes de datos, con datos XML o para administrar datos locales de la aplicación. El DataSet contiene una colección de uno o más objetos DataTable formados por filas y columnas de datos, así como información sobre claves principales, claves externas, restricciones y relaciones, relativa a los datos incluidos en los objetos DataTable.

El otro elemento central de la arquitectura de ADO.NET es el **proveedor de datos de .NET Framework**, cuyos componentes están diseñados

expresamente para la manipulación de datos y para el acceso rápido a datos de avance de sólo lectura. El objeto Connection proporciona conectividad con un origen de datos. El objeto Command permite tener acceso a comandos de base de datos para devolver datos, modificar datos, ejecutar procedimientos almacenados y enviar o recuperar información sobre parámetros. El objeto DataReader proporciona una secuencia de datos de alto rendimiento desde el origen de datos. Por último, el objeto DataAdapter proporciona el puente entre el objeto DataSet y el origen de datos. El DataAdapter utiliza objetos Command para ejecutar comandos SQL en el origen de datos tanto para cargar el DataSet con datos como para reconciliar en el origen de datos los cambios aplicados a los datos incluidos en el DataSet.

2.3.2 CONJUNTO DE DATOS

El DataSet es una representación residente en memoria de datos que proporciona un modelo de programación relacional coherente independientemente del origen de datos. Se puede utilizar con múltiples y distintos orígenes de datos, con datos XML o para administrar datos locales de la aplicación. El DataSet representa un conjunto completo de datos entre los que se incluyen tablas relacionadas, restricciones y relaciones entre las tablas. En la siguiente ilustración se muestra el modelo de objeto DataSet.

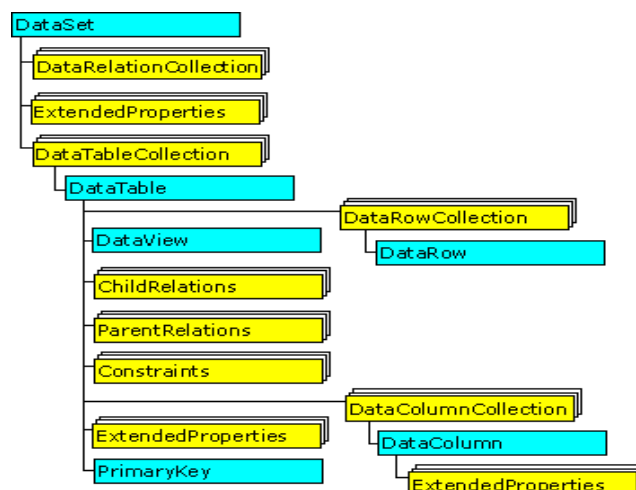


Figura II. 13. Modelo de Objeto DataSet

Los datos del conjunto de datos suelen ser una versión muy reducida de lo que hay en la base de datos. Sin embargo, puede trabajar con ellos igual que lo hace con los datos reales. Mientras se esté trabajando, se permanecerá desconectado de la base de datos, que quedará libre para ejecutar otras tareas.

El conjunto de datos es un contenedor pasivo para los datos. Para extraer realmente datos de la base de datos y (opcionalmente) escribirlos de nuevo, se utilizan adaptadores de datos. Un adaptador de datos contiene uno o varios comandos utilizados para llenar una única tabla del conjunto de datos y para actualizar la tabla correspondiente de la base de datos. Como el conjunto de datos es realmente una copia privada de los datos de la base de datos, no reflejará necesariamente el estado actual de la base de datos, para ver los últimos cambios realizados por otros usuarios, se puede actualizar el conjunto de datos; llamando al método Fill correspondiente.

2.3.2.1 DATATABLECOLLECTION

Un DataSet de ADO.NET contiene una colección de cero o más tablas representadas por objetos DataTable. La DataTableCollection contiene todos los objetos DataTable de un DataSet.

Un DataTable se define en el espacio de nombres System.Data¹⁰ y representa una única tabla de datos residentes en memoria. Contiene una colección de columnas representadas por una DataColumnCollection y restricciones representadas por una ConstraintCollection que, juntas, definen el esquema de la tabla. Un DataTable también contiene una colección de filas representadas por la DataRowCollection, que contiene los datos de la tabla. Junto con su estado actual, un DataRow conserva tanto la versión original como la actual para identificar los cambios realizados en los valores almacenados en la fila.

2.3.2.2 DATARELATIONCOLLECTION

Un DataSet contiene relaciones en su objeto DataRelationCollection. Una relación, representada por el objeto DataRelation, asocia las filas de un DataTable con las filas de otro DataTable. Es análogo a una ruta de unión que podría existir entre las columnas de claves externas y principales en una base de datos relacional. Un DataRelation identifica columnas coincidentes en dos tablas de un DataSet.

¹⁰ El espacio de nombres **System.Data** proporciona acceso a las clases que representan la arquitectura de ADO.NET.

Las relaciones permiten pasar de una tabla a otra dentro de un mismo DataSet. Los elementos esenciales de un DataRelation son el nombre de la relación, el nombre de las tablas relacionadas y las columnas relacionadas de cada tabla. Se pueden establecer relaciones con más de una columna por tabla, para lo que debe especificar una selección de objetos DataColumn como columnas clave. Cuando se agrega una relación al DataRelationCollection, se puede agregar también un UniqueKeyConstraint y un ForeignKeyConstraint para imponer restricciones de integridad cuando se realicen cambios en los valores de las columnas relacionadas.

2.3.2.3 EXTENDEDPROPERTIES

El DataSet (así como el DataTable y el DataColumn) tiene una propiedad ExtendedProperties. ExtendedProperties es un PropertyCollection en el que se puede colocar información personalizada, como la instrucción SELECT que sirve para generar el conjunto de resultados, o una marca de fecha y hora de cuándo se generaron los datos. La colección ExtendedProperties se conserva con la información del esquema del DataSet (así como el DataTable y el DataColumn).

2.3.2.4 SERIALIZACION Y DESERIALIZACION DEL DATASET

Se debe tomar en cuenta que el objeto DataSet tiene la capacidad de controlar su propia serialización (proceso de convertir un objeto en una secuencia de bytes para conservarlo en memoria, una base de datos o

un archivo. Su propósito principal es guardar el estado de un objeto para poder crearlo de nuevo cuando se necesita) y deserialización (proceso inverso)¹¹.

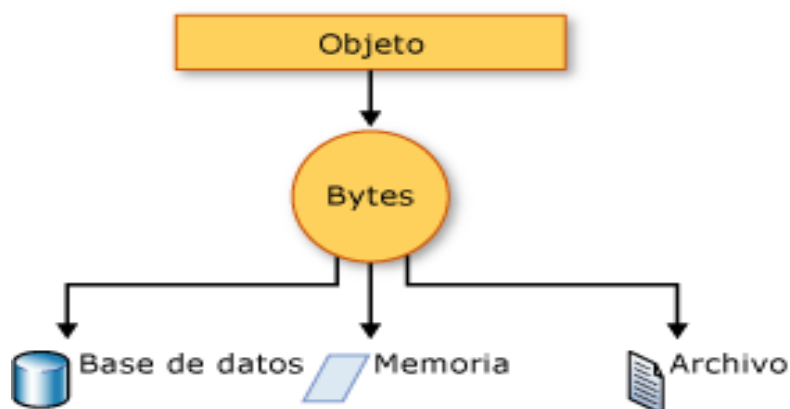


Figura II. 14. Funcionamiento de la serialización

El objeto se serializa en una secuencia que, además de los datos, contiene información sobre el tipo de objeto, como la versión, referencia cultural y nombre de ensamblado. Esa secuencia se puede almacenar en una base de datos, un archivo o en memoria.

Una de las ventajas de utilizar conjuntos de datos es que los componentes pueden intercambiarlos cuando lo necesiten. Por ejemplo, un objeto podría crear y llenar un conjunto de datos y, a continuación, enviarlo a otro componente en otro punto de la aplicación para que lo procese. Esta facilidad significa que los componentes no necesitan consultar la base de datos de forma individual. Un conjunto de datos de ADO.NET es una vista relacional de los datos que se pueden representar en XML.

¹¹ Los objetos **DataSet** y **DataTable** se heredan de **MarshalByValueComponent** y admiten la interfaz **ISerializable** para obtener acceso a ellos de forma remota. Éstos son los únicos objetos ADO.NET a los que se puede tener acceso remoto.

2.3.3 PROVEEDORES DE DATOS

El otro elemento central de la arquitectura de ADO.NET es el proveedor de datos de .NET Framework, cuyos componentes están diseñados expresamente para la manipulación de datos y para el acceso rápido a datos de avance de sólo lectura. Esos resultados se procesan directamente o se colocan en un **DataSet** de ADO.NET con el fin de exponerlos al usuario para un propósito específico, junto con datos de varios orígenes, o de utilizarlos de forma remota entre niveles. El diseño del proveedor de datos de .NET Framework hace que sea ligero, de manera que cree un nivel mínimo entre el origen de datos y su código, con lo que aumenta el rendimiento sin sacrificar la funcionalidad.

Objeto	Descripción
Connection	Establece una conexión a un origen de datos determinado.
Command	Ejecuta un comando en un origen de datos. Expone Parameters y puede ejecutarse en el ámbito de un objeto Transaction de Connection.
DataReader	Lee una secuencia de datos de sólo avance y sólo lectura desde un origen de datos.
DataAdapter	Llena un DataSet y realiza las actualizaciones necesarias en el origen de datos.

Tabla II. I. Objetos centrales que constituyen un proveedor de datos de .NET Framework

CAPÍTULO III MODELO DE SERVICIOS WEB PARA LA REPLICACIÓN DE DATOS

En la actualidad la interoperabilidad entre distintos motores de base de datos se ha convertido en un pilar fundamental en el ámbito empresarial pero se encuentra limitado ya que esta responsabilidad recae en la base de datos que generalmente corresponden a versiones empresariales de mayor uso comercial y la idea de interoperabilidad se restringe aun mas si las bases de datos corresponden a distintos proveedores.

Por esta razón se quiere aprovechar de las múltiples ventajas que se tiene al usar servicios web para dar solución a estos inconvenientes y crear un ambiente que permita representar la réplica de mezcla sin usar las herramientas que son proporcionadas por los distintos proveedores de bases de datos

El modelo propuesto de servicios web permite la interoperabilidad entre servidores que participan en la réplica de datos y específicamente permite el intercambio y distribución de información que se va generando en cada servidor de manera independiente. El intercambio de información entre servidores se la realiza de manera programada lo que significa que el tipo de suscripción que se ha modelado es una suscripción por extracción, es decir, cada cierto periodo de tiempo los servidores

invocan los servicios web necesarios para mantener sincronizada su información con el resto de servidores.

3.1 Servicio Web generador de sentencias SQL

3.1.1 Estructura de la Base de Datos

Es de gran importancia para la réplica de información conocer la estructura de la base de datos, dicha estructura debe ser proporcionada por el servidor que cumpla el rol de Publicador.

*Information Schema*¹² contiene un catálogo completo con las bases, tablas, tipos de columnas etc. El servicio web es capaz de generar el sentencias SQL necesarias para recopilar información fundamental de la base de datos a replicarse como:

- Tablas de una base de datos
- Recuperar claves primarias.
- Identificar campos auto incrementales y campos calculados.
- Identificar los tipos de datos de las columnas de una tabla.

Las sentencias SQL generadas por este servicio web permiten extraer esta información de los motores de base de datos MySQL y Sql Server indistintamente, es decir que se puede configurar como Publicador a cualquiera de estos dos proveedores de datos.

¹² **Information Schema:** es un estándar ANSI que provee de información acerca de las tablas, vistas, columnas y procedimientos en una Base de Datos.

3.1.2 Configuración de la Base de Datos

Es necesario preparar a los servidores de manera que la réplica se ejecute de forma adecuada y obtener los resultados esperados por esta razón la base de datos se debe configurar de tal forma que se pueda identificar las tablas que se replicaran y a su vez identificar la información que se debe propagar hacia los demás servidores.

Para realizar este control el servicio web crea sentencias que permiten definir los objetos de la base de datos. Específicamente se agregan nuevas columnas a todas las tablas que se publicaran. La función de cada campo se detalla a continuación:

Acción Realizada – ACROW, este campo de tipo *int*, permite identificar si el registro ha sido creado o modificado. Puede tomar los siguientes valores:

VALOR	DESCRIPCION
1	Identifica a un registro como creado por primera vez
2	Identifica a un registro como modificado
0	Identifica a un registro en estado pasivo.

Tabla III.II. Especificación de valores del campo ACROW

Ultima modificación – DTROW, este campo de tipo *datetime*, permite registrar la ultima fecha en la que ha sufrido cambios el registro.

Identificación de Fila – IDROW, este campo de tipo varchar(50), permite almacenar un identificador de fila, inicialmente toma el valor de la columna(s) que forma la clave primaria.

	cod_estado	descripcion	acRow	dtRow	idRow
	1	Cancelado	0	12/11/2011 17:07:	1
	2	Pendiente	2	15/11/2011 23:51:	2
	3	Inactivo	1	15/11/2011 23:51:	3

Figura III.15. Campos agregados para el control de cambios

Bitácora de Eliminados, esta tabla es creada con el fin de llevar un registro de las filas que se eliminan de las tablas que se han configurado para su publicación. En esta tabla se registrarán solo las filas que tengan en el campo ACROW el valor de 0 (estado inactivo) o 2 (fila modificada).

La información de los nuevos campos se llenan siguiendo instrucciones que se ejecutan automáticamente después de que se produce un evento en el servidor de base de datos. Los desencadenadores creados en cada tabla a publicarse, cumplen tareas específicas que a continuación se describen:

TRIGGER	DESCRIPCIÓN
TR_INS_[NOMBRE DE LA TABLA]	Al momento de insertar un nuevo registro el trigger de inserción asigna el valor de 1 al campo ACROW, la fecha

	actual al campo DTROW y el valor de la clave primaria al campo IDROW
TR_UPD_[NOMBRE DE LA TABLA]	Si el registro es modificado se asigna el valor de 2 a ACROW y la fecha actual a DTROW. Después de la sincronización se asigna el valor de 0 al campo ACROW y la fecha actual al campo DTROW
TR_DEL_[NOMBRE DE LA TABLA]	Si el campo es eliminado y el valor del campo ACROW es 2 o 0 el registro se elimina y su información (nombre de la tabla, fecha actual, IDROW) se inserta en la tabla bitácora de eliminados caso contrario dicho registro se borrara por completo.

Tabla III.III. Funcionamiento de los triggers en las tablas a publicarse.

Para la propagación y sincronización de la información que se ha recopilado de los distintos servidores es necesario generar las sentencias para la manipulación de datos. Dichas sentencias permiten recuperar los cambios de cada servidor y crea sentencias que se utilizan para manejar los datos de la base de datos (insert, update, delete).

El servicio web debe recibir los parámetros necesarios para identificar el tipo de proveedor a sincronizarse para este caso es necesario especificar: el esquema de la base de datos Publicadora, claves primarias de las tablas a replicar, el servidor a sincronizarse y el proveedor de base de datos del servidor.

Las sentencias para la sincronización tienen la siguiente estructura:

- Estructura de la sentencia Insert para la sincronización:

```

Insert into [nombre_tabla]
(campo1 [,campo2,....campoN] )
values(@campo1_v [,@campo2,.....,@campoN] )

```

- Estructura de la sentencia Update para la sincronización:

```

UPDATE [nombre_tabla]
SET
    campo1 = @campo1_v ,
    .
    .
    idRow = @nuevo_idRow
WHERE idRow = @idRow_v

```

- Estructura de la sentencia Delete para la sincronización:

```

DELETE FROM [nombre_tabla]
WHERE idRow = @idRow_v

```

3.2 Servicio Web para la distribución de la información

La información recopilada es guardada en archivos con estructura y formato XML. Esta información es enviada al Servicio Web que tiene como finalidad establecer un vínculo con el servidor de base de datos y el servidor FTP. Este servicio web cumple con las siguientes tareas específicas:

- Establecer la conexión con el FTP
- Crear directorios
- Eliminar Directorios
- Borrar Archivos
- Descargar Archivo del FTP

- Cargar Archivo al FTP

3.3 Servicio Web para la administración del catalogo de replicación

El catalogo de replicación lleva un registro de los servidores que están configurados para la réplica. En este catalogo se registra:

- El rol que desempeña el servidor en la réplica.
- El nombre del servidor de base de datos.
- Directorio en el FTP donde se almacenara los archivos que genere el servidor, el nombre del directorio estará conformado por las iniciales del rol que cumple (Dist, Publ, Susc) más el nombre del servidor. Ejemplo: Publ_SrvMatriz.



Figura III.16. Directorios creados en el FTP pertenecientes a los servidores configurados.

- Nombre de la base de datos a replicarse.
- Fecha en las que registro el servidor y la fecha en la que el servidor empezó a enviar información.
- Registro de los archivos que ha generado cada servidor a mas de identificar si el archivo contiene información vigente.

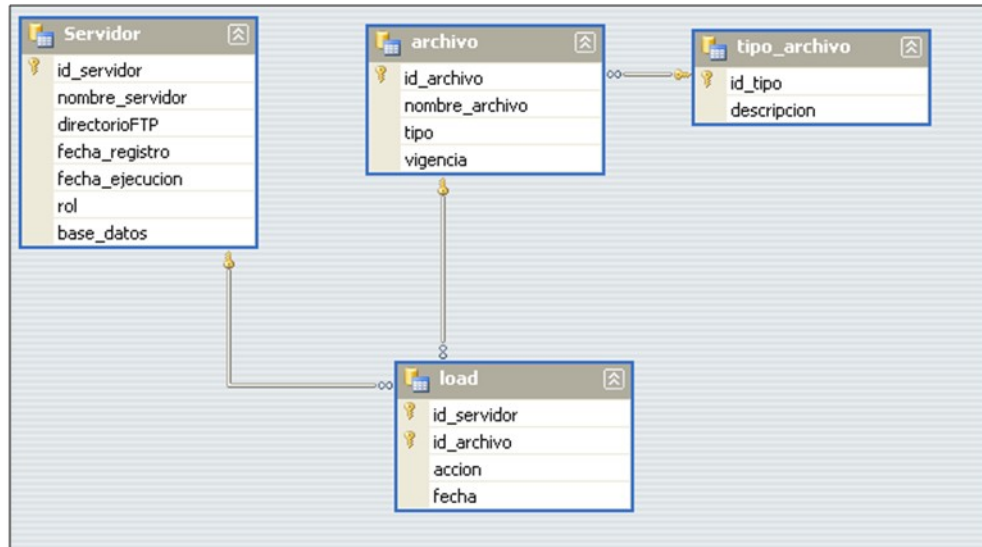


Figura III.17. Estructura del catálogo de replicación

El servidor registrado tiene la capacidad de generar archivos que cumplen con una tarea específica dependiendo del rol que desempeñe en la réplica. Los archivos que pueden generar son:

- **PublicadorSuscriptor**
 - Archivo de control de cambios (CTR)
- **Publicador**
 - Archivo con el Backup de la base de datos (BKP)
 - Archivo con el esquema de la base de datos (SCH)



Figura III.18. Archivos generados por el Publicador



Figura III.19. Archivo generado por el Suscriptor

- **Distribuidor**

- Archivo con el catalogo de replica (REG)
- Archivo con los datos a sincronizase (SYN)
- Archivo con los datos que presentaron conflictos al realizar la sincronización (CFL).



Figura III.20. Archivo generado por el Distribuidor

Cada archivo contiene información fundamental para el proceso de replica y sincronización de bases, la siguiente tabla muestra una descripción de estos archivos.

TIPO DE ARCHIVO	DESCRIPCIÓN
-----------------	-------------

Control de Cambios – CTR	Este archivo XML contiene los cambios que se han generado en el servidor de base de datos (nuevos registros, registros modificados, registros eliminados). Este archivo es creado por el servidor que cumple el rol de Publicador o Suscriptor.
Backup – BKP	El archivo XML contiene la información de la última base de datos sincronizada satisfactoriamente. Este archivo es creado por el Publicador y el Suscriptor pero solo el archivo creado por el Publicador es almacenado en el Distribuidor (FTP).
Esquema – SCH	Este archivo XML contiene la estructura de la base de datos a publicarse, a continuación se detalla su contenido: Nombre de las tablas a publicarse. Nombre de las columnas. Tipo de columna. Longitud. Clave primaria (campo que se llena si la columna es clave primaria caso contrario es este campo estará vacío).
Sincronización – SYN	El archivo XML contiene información de los datos que deben propagarse. Este archivo es generado por el Distribuidor después de realizar la mezcla de datos y resolver los conflictos (manipulación del mismo registro de una tabla por varios servidores).
Conflictos – CFL	Este archivo XML contiene información de los registros que no serán propagados ya que en el proceso de mezcla estos registros fueron descartados porque siguiendo políticas en la resolución de conflictos para la sincronización se acepta el cambio realizado por un solo servidor.
Catalogo de Replica – REG	Este archivo XML da la persistencia a los datos de los servidores, roles que desempeñan en la red e información que ha generado cada uno.

Tabla III.IV. Archivos generados por lo servidores

Dentro de las tareas de administración que permite el servicio web en el Catalogo de Replicación tenemos:

- Registrar la información de nuevos servidores.
- Eliminar información de servidores que se desvinculan de la réplica.
- Registrar los archivos que son generados por los servidores.
- Identificar los archivos con información vigente usada en el proceso de réplica.
- Obtener información de servidores que participan en el proceso de réplica.

3.4 Suscripción por Extracción

Es necesario determinar la interacción del modelo de replicación que se propone por esta razón se hace uso de la metáfora de publicación de una revista en el mundo real, donde se identifica tres roles, ellos son:

- *Publicador*, entidad encargada de la emisión de la publicación de la revista.
- *Suscriptor*, entidad (entidades o personas) que consumen o leen la publicación emitida por dicho publicador.
- *Distribuidor*, entidad encargada de la entrega de las publicaciones emitidas por el publicador a todos los suscriptores.

Además se debe manejar los conceptos básicos de una publicación ellos son:

- *Publicación*, uno de los ejemplares de la revista emitida por el publicador.

- *Artículo*, unidad mínima de información que conforma la publicación.

La metáfora de publicación nos permite especificar dos tipos de suscripción, es decir dos formas de obtener una revista:

- Suscripción por inserción, a través de una suscripción directa a la revista lo que implica que cada vez que sale una edición de la misma, esta se entrega a cada uno de los suscriptores.
- Suscripción por extracción, es solicitándola a cualquiera de las entidades que se encargan de la distribución de la misma.

Esta metáfora permite adaptar los conceptos a la base de datos, redefiniendo de la siguiente forma:

- **Publicador:** equipo servidor de base de datos encargado de generar una publicación.
- **Suscriptor:** equipos servidores de base de datos que consumen una publicación existente.
- **Distribuidor:** equipo servidor que conoce al publicador sus publicaciones y a sus respectivos suscriptores de cada publicación, además se encarga de entregar las publicaciones a cada uno de ellos.

Como se mencionó anteriormente se ha modelado una Suscripción por Extracción, esto significa que el suscriptor solicita la última publicación generada por el publicador, esta petición se la hace directamente al Distribuidor.

La publicación de información, es una tarea realizada por el publicador y todos los suscriptores que estén involucrados en la réplica. El agente de replica para la publicación cada cierto periodo de tiempo invoca al servicio web que recopila toda la información que ha cambiado en el servidor y genera una publicación en archivo XML Encriptado que contiene las tablas a replicarse (artículos). Para finalizar el servicio web de distribución almacena este archivo con los cambios de servidor su respectivo directorio y registra esta carga en el catalogo de replica invocando al servicio web encargado de esta tarea.

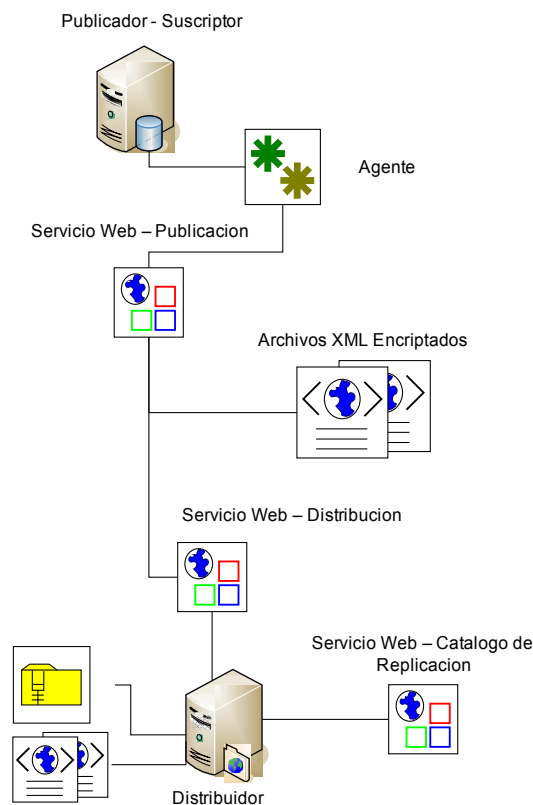


Figura III.21. Publicación de Información

Sincronización de la información, el agente de sincronización invoca al servicio web para verificar la existencia de un archivo que le permita sincronizarse, si existe este archivo se lo descarga e inicia el proceso de sincronización invocando el servicio web

que le permite generar las sentencias SQL necesarias. Se recupera la información de la base de datos (backup) en el caso de que el servidor cumpla con el rol de publicador y envía el archivo XML encriptado con la información usando el servicio web de distribución. Para finalizar se registra esta carga en el catalogo de replica invocando al servicio web encargado de esta tarea.

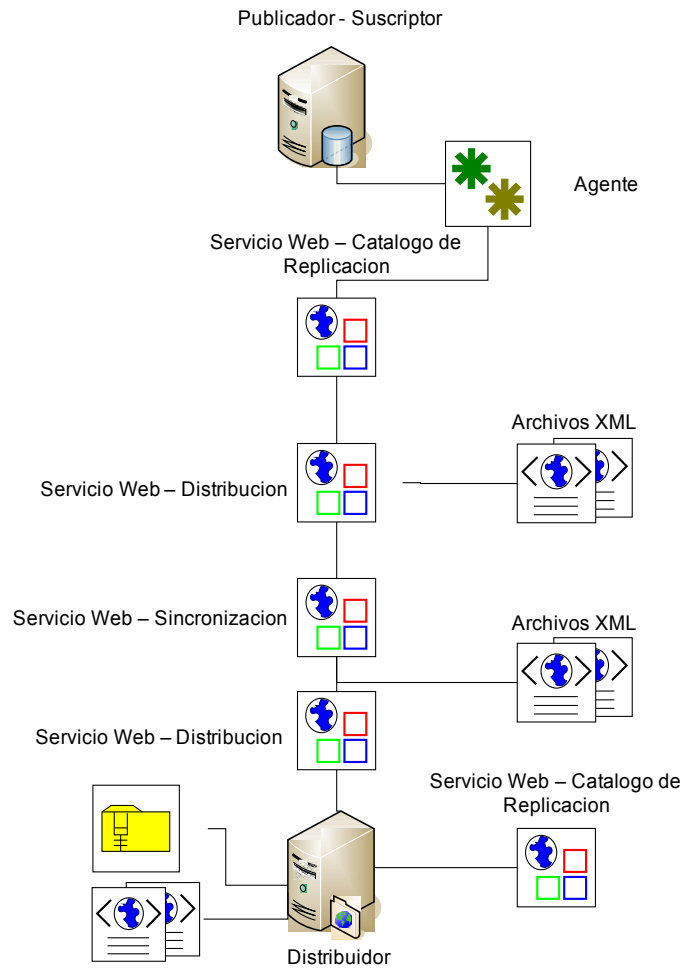


Figura III.22. Proceso de Sincronización

3.5 Consola de Administración de Replica

El modelo desarrollado de servicios web y los agentes (Servicios Windows para la publicación, mezcla de información y sincronización) interactúan para llevar a cabo las

terreas de replicación. Las tareas de administración de la replica se lleva a cabo a través de una consola realizada con una arquitectura de *cliente inteligente* ya que no se basa en un explorador para proporcionar el entorno de ejecución de seguridad y de interfaz de usuario que necesita para su funcionamiento. Esta consola mantiene código compilado (componentes, ensamblados, etc.) que se ejecutan en el equipo cliente para ofrecer la interfaz de usuario de la aplicación y la lógica del lado del cliente.

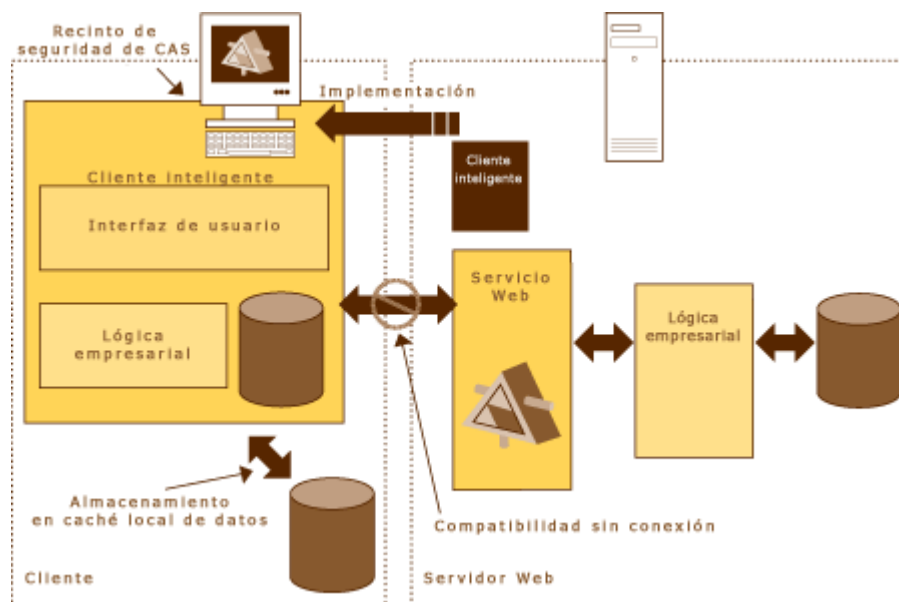


Figura III.23. Representación esquemática de una arquitectura cliente inteligente

Esta aplicación y servicios Windows desarrollados con .NET constan de tres capas: Acceso a Datos, Lógica de Negocios y Presentación.

En la capa de **Acceso a Datos** es usado el componente de ADO.NET "Proveedor de Acceso a Datos de .Net Framework", cuyos componentes diseñados expresamente para la manipulación de datos nos permiten establecer la conexión a un motor de

base de datos en este caso puede ser SQL Server 2000 o MySql 5.1 y tener acceso a sus datos.

La capa **Logica de Negocios**, es la encargada de interpretar las sentencias SQL enviadas por el servicio web para la configuración de la base de datos y la manipulación de la información de esa base. Esta capa también nos proporciona métodos de encriptación para los archivos que generan los servidores como medida de seguridad, el algoritmo de encriptación usado es AES (Advanced Encryption Standard), también conocido como Rijndael. Se le considera el algoritmo de cifrado más seguro de la actualidad. Uno de los aspectos que se destacan en esta capa es la capacidad de resolver conflictos en la réplica de mezcla permitiendo que varios servidores realicen cambios independientemente.

Los métodos desarrollados permiten detectar y resolver conflictos en base a reglas o políticas que se establecieron de acuerdo a las necesidades de la empresa en donde se desarrollo el proyecto, estas reglas son:

1. Si se produce un conflicto entre un publicador y un suscriptor, el cambio del publicador se mantiene y el cambio del suscriptor se descarta.
2. Si se produce un conflicto entre dos suscriptores, se verifica la fecha (DTROW) y prevalece el cambio realizado por el primer suscriptor que ha modificado ese registro y el cambio del segundo suscriptor se descarta.

Aunque la mayoría de los conflictos están relacionados con actualizaciones (una actualización en un servidor entra en conflicto con una actualización o eliminación en otro servidor), existen otros tipos de conflictos.

Conflictos de inserción – inserción

Se detecta un conflicto de este tipo cuando una inserción de fila en una tabla de un servidor coincide a nivel de clave primaria con otra fila insertada por otro servidor, en este caso la fila ganadora se registra en el archivo de sincronización y la fila perdedora se registra en el archivo de conflictos.

Conflictos de actualización – actualización

Se detecta un conflicto de este tipo cuando una actualización de fila en una tabla de un servidor coincide con la actualización de la misma fila en otro servidor, en este caso la fila ganadora se registra en el archivo de sincronización y la fila perdedora se registra en el archivo de conflictos.

Conflictos de eliminación – actualización

Se detecta un conflicto de este tipo cuando una actualización de fila en una tabla de un servidor coincide con la eliminación de la misma fila en otro servidor, en este caso la fila ganadora se registra en el archivo de sincronización y la fila perdedora se registra en el archivo de conflictos. En el momento de la sincronización si la acción que prevalece es la actualización, en el servidor perdedor se insertara la fila con los datos de la fila ganadora; caso contrario si la acción que prevalece es la de eliminación el servidor perdedor eliminara esta fila.

Los agentes (Servicios Windows), y la consola tienen un archivo de configuración que definen los tipos de motores soportados por el modelo en este caso SQL Server 2000 y MySql 5.1. La consola permite configurar los agentes para programar una agenda

que debe seguir para la correcta ejecución de la réplica usando el modelo propuesto de servicios Web.

3.6 DEMOSTRACION DE LA HIPÓTESIS

Para la demostración de la hipótesis se va a utilizar el método científico experimental lo que nos permitirá hacer una descripción del problema y realizar una serie de experimentos o pruebas que determinaran la aceptación o rechazo de la hipótesis que se ha planteado.

Estudiar el proceso de réplica de datos es fundamental para el desarrollo de un modelo de Servicios Web y agentes que cumplan el objetivo de transportar datos entre dos o más servidores, permitiendo que ciertos datos de la base de datos estén almacenados en más de un sitio manteniendo la consistencia entre datos en fuentes remotas.

Hipótesis: “Los agentes de réplica en servicios web permitirán la sincronización de los sistemas satélites de CONTILIBRO”

Técnica de comprobación de la hipótesis:

La hipótesis fue comprobada con una técnica no estadística basada en porcentajes, la única variable de la que depende la hipótesis es el grado de sincronización que será representado por PS.

La cuantificación de PS se basa en la determinación del grado de sincronización alcanzado por cada tipo de dato de la muestra, este valor se ha denotado como psi.

Pruebas Realizadas para la comprobación de la hipótesis

Las pruebas que se realizaron para la comprobación de la hipótesis fueron:

- Pruebas de inserción de datos.
- Pruebas de actualización de datos.
- Pruebas de eliminación de datos.

Cabe recalcar que estas pruebas se las realizó tomando en cuenta el rol de cada motor de base de datos, es decir, ambos motores (SQL Server y MYSQL) actuaron como publicador y suscriptor, esto se realizó debido a que el modelo de Servicios Web debe replicar los datos sin importar el proveedor de base de datos ni el rol que cumpla. Los tipos de datos que se usaron en el desarrollo de la aplicación satélite se decidió darles más ponderación, estos tipos de datos son: int, decimal, varchar, datetime,image.

Población:

La población que se va a tomar en cuenta para su análisis serán los tipos de datos que manejan los motores de base de datos SQL y MYSQL dándoles más énfasis a los tipos de datos que se utilizaron para el desarrollo de la aplicación satélite de Contilibro.

Método de Muestra:

Se va a utilizar un método de muestra no probabilístico por la siguiente razón:

El tiempo que requiere un análisis de todos los tipos de datos de los motores de bases de datos SQL y MYSQL es demasiado extenso.

Para esto se ha considerado los tipos de datos más utilizados por la aplicación los mismos que están inmersos en el proceso de sincronización de datos.

Muestra:

Si bien no es posible analizar el proceso de sincronización con todos los tipos de datos que manejan SQL y MYSQL por las razones antes mencionadas se ha escogido los tipos de datos que utilizan las aplicaciones satélites de Contilibro a las cuáles se les ha dado un mayor porcentaje para la verificación de la hipótesis, estos tipos de datos son:

Nombre del Tipo de Dato	Tipo de Dato SQL	Tipo de Dato MYSQL
Números enteros	bigInt, int, smallInt, tinyInt, bit	bigInt, int, smallInt, tinyInt
Cadena de caracteres	Varchar(), Char()	Varchar(), Char()
Números Decimales	Decimal(7,2), float	Decimal(7,2), float
Fechas	DateTime	DateTime
Tipos de Datos para Imágenes	Image	tinyBlobs, Blob, mediumBlob, longBlob

Tabla III.V. Tipos de datos usados para la comprobación de la hipótesis.

Para obtener el tamaño de la muestra se utilizaron fórmulas estadísticas para población finita e infinita, tomando en cuenta que cuando un tipo de dato posee más de 100000 valores posibles se considera como una población infinita.

Fórmula para población finita:

$$n = \frac{N * Z^2 * p * q}{\left(\frac{e}{100}\right)^2 * (N - 1) + Z^2 * p * q}$$

Fórmula para población infinita:

$$n = \frac{Z^2 * p * q}{\left(\frac{e}{100}\right)^2}$$

Donde:

N= población

Z= nivel de confiabilidad

p=probabilidad de que ocurra un error

q=p 1

e=error

Para este caso se tomo el nivel de confiabilidad Z= 90% lo que equivale a 1,645, y el error e=3%.

Pruebas:

Para realizar las pruebas primeramente se procedió a calcular las muestras con las fórmulas anteriormente descritas, luego se generaron datos aleatorios para realizar inserciones, actualizaciones y eliminaciones; después de procedió a comparar los datos tanto de SQL como de MYSQL, la generación de datos aleatorios y la comparación se las realizó mediante un programa encargado de calcular el porcentaje en el que se sincronizaron los datos. Los resultados obtenidos se muestran en una tabla resumen que permite verificar el porcentaje de sincronización que se logró de esta manera:

Tipo de dato MYSQL	Tipo de dato .NET	Tipo de dato SQL	Población	Tamaño de la muestra	Porcentaje Conseguido	Porcentaje Propuesto	Sincronización Lograda
BIGINT	INT64	BIGINT	INFINITA	752	100,00	5,55	5,55
INT	INT32	INT	INFINITA	752	100,00	10,00	10,00
SMALLINT	INT16	SMALLINT	65535	743	100,00	5,55	5,55
TINYINT	SBYTE	TINYINT	255	191	54,97	5,55	3,05
DECIMAL	DECIMAL	DECIMAL	INFINITA	752	100,00	10,00	10,00
BOOL	BOOLEAN	BIT	2	2	100,00	5,55	5,55
FLOAT	SINGLE	FLOAT	INFINITA	752	100,00	5,55	5,55
DATETIME	DATETIME	DATETIME	INFINITA	752	85,37	10,00	8,54
CHAR	STRING	CHAR	255	191	52,00	5,60	2,91
VARCHAR	STRING	VARCHAR	65455	687	100,00	10,00	10,00
TINYBLOB	BYTE[]	IMAGE	255	191	100,00	5,55	5,55
BLOB	BYTE[]	IMAGE	65535	743	100,00	5,55	5,55
MEDIUMBLOB	BYTE[]	IMAGE	INFINITA	752	100,00	5,55	5,55
LONGBLOB	BYTE[]	IMAGE	INFINITA	752	100,00	10,00	10,00
% PS₁		93,35					

Tabla III.VI. Pruebas de inserción de datos (MYSQL como Publicador)

Tipo de dato MYSQL	Tipo de dato .NET	Tipo de dato SQL	Población	Tamaño de la muestra	Porcentaje Conseguido	Porcentaje Propuesto	Sincronización Lograda
BIGINT	INT64	BIGINT	INFINITA	752	100,00	5,55	5,55
INT	INT32	INT	INFINITA	752	100,00	10,00	10,00
SMALLINT	INT16	SMALLINT	65535	743	100,00	5,55	5,55
TINYINT	SBYTE	TINYINT	255	191	53,40	5,55	2,96
DECIMAL	DECIMAL	DECIMAL	INFINITA	752	100,00	10,00	10,00
BOOL	BOOLEAN	BIT	2	2	100,00	5,55	5,55
FLOAT	SINGLE	FLOAT	INFINITA	752	100,00	5,55	5,55
DATETIME	DATETIME	DATETIME	INFINITA	752	83,11	10,00	8,31
CHAR	STRING	CHAR	255	191	44,96	5,60	2,52
VARCHAR	STRING	VARCHAR	65455	687	100,00	10,00	10,00
TINYBLOB	BYTE[]	IMAGE	255	191	100,00	5,55	5,55
BLOB	BYTE[]	IMAGE	65535	743	100,00	5,55	5,55
MEDIUMBLOB	BYTE[]	IMAGE	INFINITA	752	100,00	5,55	5,55
LOB	BYTE[]	IMAGE	INFINITA	752	100,00	10,00	10,00
% PS₂			92,64				

Tabla III.VII. Pruebas de Actualización de datos (Publicador MYSQL)

Tipo de dato MYSQL	Tipo de dato .NET	Tipo de dato SQL	Población	Tamaño de la muestra	Porcentaje Conseguido	Porcentaje Propuesto	Sincronización Lograda
BIGINT	INT64	BIGINT	INFINITA	752	100,00	5,55	5,55
INT	INT32	INT	INFINITA	752	100,00	10,00	10,00
SMALLINT	INT16	SMALLINT	65535	743	100,00	5,55	5,55
TINYINT	SBYTE	TINYINT	255	191	100,00	5,55	5,55
DECIMAL	DECIMAL	DECIMAL	INFINITA	752	100,00	10,00	10,00
BOOL	BOOLEAN	BIT	2	2	100,00	5,55	5,55
FLOAT	SINGLE	FLOAT	INFINITA	752	100,00	5,55	5,55
DATETIME	DATETIME	DATETIME	INFINITA	752	100,00	10,00	10,00
CHAR	STRING	CHAR	255	191	100,00	5,60	5,60
VARCHAR	STRING	VARCHAR	65455	687	100,00	10,00	10,00
TINYBLOB	BYTE[]	IMAGE	255	191	100,00	5,55	5,55
BLOB	BYTE[]	IMAGE	65535	743	100,00	5,55	5,55
MEDIUMBLOB	BYTE[]	IMAGE	INFINITA	752	100,00	5,55	5,55
LONGBLOB	BYTE[]	IMAGE	INFINITA	752	100,00	10,00	10,00
% PS₃			100,00				

Tabla III.VIII. Pruebas de Eliminación de datos (Publicador MYSQL)

Tipo de dato SQL	Tipo de dato .NET	Tipo de dato MYSQL	Población	Tamaño de la muestra	Porcentaje Conseguido	Porcentaje Propuesto	Sincronización Lograda
BIGINT	INT64	BIGINT	INFINITA	752	100	5,55	5,55
INT	INT32	INT	INFINITA	752	100	10,00	10,00
SMALLINT	INT16	SMALLINT	65535	743	100	5,55	5,55
TINYINT	BYTE	TINYINT	255	191	51,3089	5,55	2,85
DECIMAL	DECIMAL	DECIMAL	INFINITA	752	100	10,00	10,00
BIT	BOOLEAN	BOOL	2	2	100	5,55	5,55
FLOAT	DOUBLE	FLOAT	INFINITA	752	41,22206	5,55	2,29
DATETIME	DATETIME	DATETIME	INFINITA	752	100	10,00	10,00
CHAR	STRING	CHAR	255	191	53,43804	5,60	2,99
VARCHAR	STRING	VARCHAR	65455	687	100	10,00	10,00
IMAGE	BYTE[]	TINYBLOB	INFINITA	752	0,2659574	5,55	0,01
IMAGE	BYTE[]	BLOB	INFINITA	752	61,17021	5,55	3,39
IMAGE	BYTE[]	MEDIUMBLOB	INFINITA	752	100	5,55	5,55
IMAGE	BYTE[]	LONGBLOB	INFINITA	752	100	10,00	10,00
% PS₄			83,74				

Tabla III.IX. Pruebas de inserción de datos (Publicador SQL)

Tipo de dato SQL	Tipo de dato .NET	Tipo de dato MYSQL	Población	Tamaño de la muestra	Porcentaje Conseguido	Porcentaje Propuesto	Sincronización Lograda
BIGINT	INT64	BIGINT	INFINITA	752	100	5,55	5,55
INT	INT32	INT	INFINITA	752	100	10,00	10,00
SMALLINT	INT16	SMALLINT	65535	743	100	5,55	5,55
TINYINT	BYTE	TINYINT	255	191	48,16754	5,55	2,67
DECIMAL	DECIMAL	DECIMAL	INFINITA	752	100	10,00	10,00
BIT	BOOLEAN	BOOL	2	2	100	5,55	5,55
FLOAT	DOUBLE	FLOAT	INFINITA	752	41,93174	5,55	2,33
DATETIME	DATETIME	DATETIME	INFINITA	752	100	10,00	10,00
CHAR	STRING	CHAR	255	191	50,28846	5,60	2,82
VARCHAR	STRING	VARCHAR	65455	687	100	10,00	10,00
IMAGE	BYTE[]	TINYBLOB	INFINITA	752	0,1329787	5,55	0,01
IMAGE	BYTE[]	BLOB	INFINITA	752	65,0266	5,55	3,61
IMAGE	BYTE[]	MEDIUMBLOB	INFINITA	752	100	5,55	5,55
IMAGE	BYTE[]	LONGBLOB	INFINITA	752	100	10,00	10,00
% PS₅			83,63				

Tabla III.X. Pruebas de actualización de datos (Publicador SQL)

Tipo de dato SQL	Tipo de dato .NET	Tipo de dato MYSQL	Población	Tamaño de la muestra	Porcentaje Conseguido	Porcentaje Propuesto	Sincronización Lograda
BIGINT	INT64	BIGINT	INFINITA	752	100	5,55	5,55
INT	INT32	INT	INFINITA	752	100	10,00	10,00
SMALLINT	INT16	SMALLINT	65535	743	100	5,55	5,55
TINYINT	BYTE	TINYINT	255	191	100	5,55	5,55
DECIMAL	DECIMAL	DECIMAL	INFINITA	752	100	10,00	10,00
BIT	BOOLEAN	BOOL	2	2	100	5,55	5,55
FLOAT	DOUBLE	FLOAT	INFINITA	752	100	5,55	5,55
DATETIME	DATETIME	DATETIME	INFINITA	752	100	10,00	10,00
CHAR	STRING	CHAR	255	191	100	5,60	5,60
VARCHAR	STRING	VARCHAR	65455	687	100	10,00	10,00
IMAGE	BYTE[]	TINYBLOB	INFINITA	752	100	5,55	5,55
IMAGE	BYTE[]	BLOB	INFINITA	752	100	5,55	5,55
IMAGE	BYTE[]	MEDIUMBLOB	INFINITA	752	100	5,55	5,55
IMAGE	BYTE[]	LONGBLOB	INFINITA	752	100	10,00	10,00
% PS₆			100,00				

Tabla III.XI. Pruebas de eliminación de datos (Publicador SQL)

Una vez finalizadas las pruebas se procedió a contabilizar el porcentaje de sincronización de la siguiente manera:

Porcentaje de Sincronización con el servidor MYSQL como Publicador:

$$PSM = (PS_1 + PS_2 + PS_3) / 3$$

$$PSM = (93.35 + 92.34 + 100) / 3$$

$$\mathbf{PS = 95.33\%}$$

Porcentaje de Sincronización con el servidor SQL como Publicador:

$$PSS = (PSI + PSA + PSE) / 3$$

$$PSS = (83.64 + 83.63 + 100) / 3$$

$$\mathbf{PSS = 89.09\%}$$

Para obtener el porcentaje de sincronización total se sacará un promedio tomando en cuenta los dos resultados anteriores.

$$PST = (PSM + PSS) / 2$$

$$PST = (95.33 + 89.09) / 2$$

$$\mathbf{PST = 92.23 \%}$$

Procedimiento de análisis de resultados

El proceso de replica se realizo sin mayores inconvenientes a excepción de algunos tipos de datos que presentaron inconvenientes en el proceso de replicación, a continuación se muestran las novedades que se presentaron en este proceso, en la siguiente tabla un resumen:

Tipo de Dato Sql Server	Tipo de Dato – MySQL	Descripción
TINYINT	TINYINT	<i>Rango en MySQL:</i> entero de 8 bits (128 a +127) <i>Rango en Sql:</i> entero de 8 bits sin signo (0 a +255)
FLOAT	FLOAT	<i>Rango en MySQL:</i> numero de coma flotante de 4 bytes <i>Rango en Sql:</i> numero de punto flotante comprendido entre 1,79E 308 y 1,79E +308
DATETIME	DATETIME	<i>Rango en MySQL:</i> entre 01/01/0001 a 31/12/9999 <i>Rango en Sql:</i> entre 01/01/1973 a 31/12/9999
CHAR	CHAR	<i>Carateristica en MySQL:</i> tipo de dato de longitud variable. <i>Carateristica en Sql:</i> tipo de dato de longitud fija.
IMAGE	TINYBLOB	<i>Rango en MySQL:</i> máximo 255 bytes. <i>Rango en Sql:</i> hasta máximo 2 GB.
	BLOB	<i>Rango en MySQL:</i> $2^{16} - 1$ bytes <i>Rango en Sql:</i> hasta máximo 2 GB.
	MEDIUMBLOB	<i>Rango en MySQL:</i> $2^{24} - 1$ bytes <i>Rango en Sql:</i> hasta máximo 2 GB.
	LOB	<i>Rango en MySQL:</i> $2^{32} - 1$ bytes <i>Rango en Sql:</i> hasta máximo 2 GB.

Tabla III.XII. Resumen de inconvenientes en el proceso de replica

Una vez que se ha determinado el porcentaje de sincronización global (PST), este parámetro demostrará la hipótesis. El valor de PST será comparado con la escala de sincronización de la figura.

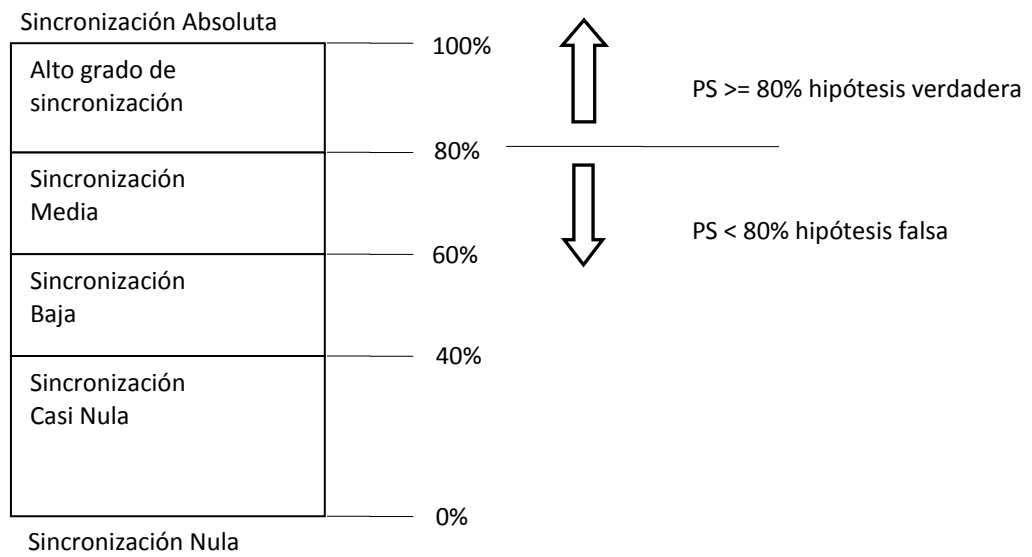


Figura III.24. Escala para medir la sincronización

Al comparar el resultado obtenido (92.23%) con la escala de medición propuesta se observa que se ha alcanzado un nivel de sincronización mayor al 80% por lo que la hipótesis queda demostrada.

CAPITULO IV DESARROLLO DE LA HERRAMIENTA DE SINCRONIZACIÓN

DATOS GENERALES

IDENTIFICACION DEL TEMA

DENOMINACIÓN

- TÍTULO: "Herramienta para la sincronización de bases de datos heterogéneas"
- Nombre del Producto: "Sync Contilibro"
- Propietario: Contilibro.

CLASIFICACIÓN

- Área: Informática
- Sector: Desarrollo de Aplicaciones

4.1 Introducción

El presente capítulo se genera debido a la imperiosa necesidad de documentar técnicamente la herramienta de sincronización de bases de datos heterogéneas, que

permite mantener sincronizadas las bases de datos SQL y MYSQL, para lograr este cometido la herramienta de sincronización se desarrolló bajo la metodología MSF.

Para empezar se documenta la fase de Definición donde se delimita la visión y el ámbito del proyecto que cumplirá los objetivos del cliente, información obtenida de las entrevistas realizadas al Señor Héctor Tello gerente general de Contilibro, así como al personal que ahí labora, lo que ayudará a organizar el proyecto. A su vez en esta etapa se habla acerca de un calendario maestro el mismo que nos indicará cuando y que tiempo nos tomaremos en realizar cada actividad o tarea.

En la segunda fase Planificación, se realizará las especificaciones funcionales, describiendo lo que se va a desarrollar, incluyendo contenido como metas de diseño, requerimientos, características y dependencias. Esto es que aquí se establecen los conceptos que especifican las necesidades de los usuarios. Organiza los componentes de la solución. Así como también se especifica las restricciones tecnológicas de la solución.

En la fase de Desarrollo, se refiere al desarrollo del proyecto, es donde se documenta el código, nomenclatura, interfaces y estrategias utilizadas en el desarrollo del proyecto.

En la cuarta fase de Estabilización trata de la liberación del producto, se refiere al material de capacitación con que debe contar el proyecto.

La quinta fase Instalación, describe el plan que se debe poner en marcha en el momento de la implementación del sistema.

4.2 Planificación y Análisis: Documento SRS

Definición del problema

La empresa Contilibro cuenta con una aplicación satélite que le permite llevar el control de inventarios y facturación, además cuenta con una página web que promociona los libros con los que cuenta; la página web toma los datos de una base de datos MYSQL mientras que la aplicación satélite consume la información de una base de datos SQL Server, por lo que se necesita que las dos bases de datos estén sincronizadas, es decir que en algún momento los dos motores deben tener los mismos datos.

El no contar con una herramienta que mantenga sincronizada la información ha dado lugar a que la información de los dos motores de base de datos que posee Contilibro no se encuentre debidamente actualizada ni sincronizada.

4.3 Visión del Proyecto

La solución que se plantea para la empresa Contilibro, es la creación de una herramienta que permita mantener sincronizada la información de los motores de base de datos SQL Server y MYSQL, por lo que la herramienta ofrece resolver el problema de sincronización para lo cual realizará las siguientes tareas:

- La herramienta mantendrá sincronizada la información de las bases de datos SQL Server y MYSQL.
- La herramienta deberá capturar los cambios que se generen en las dos bases de datos.
- La herramienta deberá mezclar los cambios de los dos motores para finalmente completar el proceso de sincronización.

- La herramienta permitirá llevar una agenda de programación, la misma que se cumplirá según se la haya configurado.

4.4 Perfiles de Usuarios

A continuación se define brevemente una descripción de cada uno de ellos:

No.	Nombre	Perfil	Tipo de Acceso	Descripción
1	Usuario Administrador	Administrador	Total	Es el usuario que tiene acceso a todo el sistema pero de forma única a la gestión de usuario del sistema.

Tabla IV.XIII. Tabla Perfiles de usuarios del sistema

4.5 Ámbito del Proyecto

4.5.1 Alcance

La herramienta de sincronización permitirá combinar las bases de datos SQL Server y MYSQL, para esto la herramienta hará uso de servicios web para extraer los cambios, mezclar la información y sincronizar las bases de datos.

4.5.2 Requerimientos Generales

Para definir los requerimientos de los usuarios, se ha clasificado a los mismos en funcionales y no funcionales, a los cuales se les aplicó técnicas de investigación científica como entrevistas.

4.5.3 Entrevista

Se realizó una entrevista al Gerente General de Contilibro, con el fin de recoger la información necesaria para obtener una visión amplia del funcionamiento de los procesos que se realizan en Contilibro.

Resultados de las Entrevistas

Contilibro necesita mantener sincronizada la información de los dos motores de base de datos.

4.5.4 Requerimientos

Los requerimientos planteados para la creación de la herramienta se describen a continuación:

REQ 1. El sistema deberá permitir registrar los servidores que van a participar en el proceso de sincronización.

REQ 2. El sistema deberá permitir capturar los cambios de cada servidor que se ha registrado.

REQ 3. El sistema deberá permitir subir la información de los cambios de cada uno de los servidores registrados.

REQ 4. El sistema deberá permitir mezclar los cambios de cada uno de los servidores.

REQ 5. El sistema deberá permitir sincronizar los servidores de bases de datos.

REQ 6. El sistema deberá permitir llevar una agenda de programación para la sincronización de datos.

4.5.5 Concepto de la Solución

El modelo de arquitectura que se escogió para el desarrollo de la aplicación, es Cliente – Servidor.

Esta arquitectura utiliza capa de lógica de negocios, en donde se comunica con la de acceso a datos utilizando el Data Access Application Block que es una clase de la Enterprise Library que contiene métodos para acceder a los datos y llamando a procedimientos almacenados.

4.5.6 Uso de Herramientas

Las herramientas software utilizadas en el desarrollo de la herramienta de sincronización se describen en la tabla IV.2.

SOFTWARE	Descripción
Plataforma Operativa de Funcionamiento	Sistema Operativo Windows XP Profesional que nos servirá para la implementación de los servidores de aplicación y base de datos.
Motor de Base de Datos	Se utilizará el motor de base de datos SQL Server 2000 el cual brinda la posibilidad de elaborar Bases de Datos Relacionales y el manejo de procedimientos almacenados para las diferentes transacciones. También se utilizará el motor de base de datos MYSQL.
Herramientas de Desarrollo	Entre las Herramienta de Desarrollo se ha seleccionado el Microsoft Visual Studio 2005 que permite el trabajo sobre la plataforma .net.
Herramientas de	Para la elaboración de ciertos tipos de diagramas que

Modelado UML	permitirán el análisis se utilizará Rational Rose.
Herramientas de oficina	Se utilizará la suite OFFICE 2007 para la elaboración de la documentación básica sobre el desarrollo del proyecto.
Herramientas para Administración y diseño de base de Datos	Se utilizará la suite de SQL Server 2000 y MYSQL.

Tabla IV.XIV. Tabla de Herramientas utilizadas en el desarrollo de la herramienta de sincronización

4.5.7 Planteamiento de la Arquitectura de la Aplicación

Se propone una arquitectura que consta de 5 capas, descritas en la siguiente tabla.

Nombre de las Capas utilizadas en el Sistema Web	Descripción
INTERFAZ DE USUARIO (ESCRITORIO)	Se efectuará la alimentación de la información a través de esta capa hacia la base de datos.
CAPA DE SERVICIOS	Los servicios web permitirán realizar el proceso de sincronización de las bases de datos SQL Server y MYSQL.
REGLAS DE NEGOCIO (CAPA 3)	Las reglas de negocio utilizarán los servicios de la capa de acceso para efectuar los procesos requeridos y presentar la información al usuario final.
ACCESO A DATOS (CAPA 2)	La capa de acceso a datos representará a todas las clases propuestas para el sistema.

<p>BASE DE DATOS (CAPA 1)</p>	<p>En el que se elaborará una base de datos relacional, para soportar la lógica del negocio. Aquí se brindará acceso a la información a través de procedimientos almacenados, debido a su rapidez de ejecución.</p>
--	---

Tabla IV.XV. Tabla de la Arquitectura de la herramienta de sincronización

4.5.8 Objetivos del Proyecto

Objetivos del Negocio

Mantener la información debidamente actualizada y lista en cualquier momento.

Objetivos del Diseño

- Lograr que la información que se maneje dentro de la organización pueda ser consultada por el personal autorizado.
- Brindar interfaces de usuario amigables e intuitivas que faciliten el trabajo cotidiano de los encargados en los diferentes procesos de la empresa.

4.6 Factores Críticos

4.6.1 Análisis de Riesgos

El análisis de riesgos se presenta una identificación preliminar en busca de la eliminación de trastornos durante el proyecto, una mayor habilidad de seguir y controlar los mismos y la confianza de la Planificación de elementos de riesgo.

4.6.2 Listado de Riesgos

Lista de Riesgos	
ID	Descripción
RI1	El equipo de desarrollo puede no adaptarse rápidamente a las tecnologías de .net.
RI2	Los desarrolladores pueden tener retrasos en el cronograma de desarrollo debido a un análisis inadecuado del sistema para Contilibro.
RI3	Al estar conectados los equipos de desarrollo al internet su vulnerabilidad es expuesta a los diferentes tipos de peligro existentes en la gran red de información.
RI4	En los equipos de desarrollo se puede presentar daños imprevistos en el hardware.
RI5	Cambios en los requerimientos por parte del Gerente General de Contilibro.
RI6	Falta de información y requerimientos mal planteados por parte del personal y el gerente general de Contilibro.
RI7	Resistencia de los usuarios finales al uso del sistema una vez finalizado.
RI8	Desacuerdos con Contilibro., con respecto al sistema satélite.

Tabla IV.XVI. Tabla de Riesgos

4.6.3 Tabla de Valores

Los valores con los cuales se representaran a los riesgos de la tabla IV.4., se definen en la tabla IV.5.

Porcentaje	Descripción	Valor
1% 33%	Alto	3
34% 67%	Medio	2
68% 99%	Bajo	1

Tabla IV.XVII. Tabla de Valores de Riesgos

4.6.4 Análisis de Riesgos.

ID	Probabilidad			Impacto	
	%	Valor	Probabilidad	Valor	Impacto
RI1	50	2	Medio	1	Bajo

RI2	80	3	Alto	2	Medio
RI3	40	2	Medio	1	Bajo
RI4	30	2	Medio	1	Bajo
RI5	50	2	Medio	2	Medio
RI6	90	3	Alto	2	Medio
RI7	40	2	Medio	2	Medio
RI8	30	1	Bajo	1	Bajo

Tabla IV.XVIII. Tabla de Análisis de Riesgos

4.6.5 Resultados

Exposición al Riesgo		
ID	Valor	Exposición
RI2	6	Alto
RI6	6	Alto
RI5	4	Medio
RI7	4	Medio
RI1	2	Bajo
RI3	2	Bajo
RI4	2	Bajo
RI8	1	Bajo

Tabla IV.XIX. Tabla de Resultados del análisis de Riesgos

Como resultado de éste análisis se toma la decisión de gestionar los riesgos con exposición Media y Alta.

4.6.6 Restricciones

Listado de Restricciones	
ID	Descripción
RI1	El equipo de desarrollo puede no adaptarse rápidamente a las tecnologías de .net.
RI3	Al estar conectados los equipos de desarrollo al internet su vulnerabilidad es expuesta a los diferentes tipos de peligro existentes en la gran red de información.
RI4	En los equipos de desarrollo se puede presentar daños imprevistos en el hardware.
RI8	Desacuerdos con Contilibro., con respecto al sistema satélite.

Tabla IV.XX. Tabla de Restricciones

4.6.7 Planificación Inicial

Recursos Humanos

Integrantes y sus Funciones	
Nombre	Función

Ángel Apina, Richard Robles	Administrador del Producto
Ángel Apina, Richard Robles	Administrador del Programa
Ángel Apina, Richard Robles	Equipo de desarrollo
Personal de Contilibro.	Equipo analista de Pruebas y Calidad
Ángel Apina, Richard Robles	Equipo encargado del despliegue del Proyecto

Tabla IV.XXI. Tabla del Recurso Humano – Integrantes y Funciones

Recursos Físicos

Numero	Recursos Físicos (Sw / Hw)
	Detalle
2	Computadora Personal Intel core 2 Duo 2.13 GHZ
2	Impreso / Scanners Lexmark
1	Sistema operativo Windows XP
1	Microsoft Visual Studio 2005
1	Servidor de Aplicaciones IIS
1	Servidor de Base de Datos SQL Server 2000
1	Suite de Oficina Office 2007.

Tabla IV.XXII. Tabla del Recurso Físico

4.7 Planificación de Actividades

El tiempo que se tomo para la realización de la herramienta de sincronización desde su planificación hasta las pruebas finales fue un estimado de 6 meses, las actividades

generales desarrolladas en el transcurso del tiempo mencionado se encuentran detalladas de la siguiente manera:

- Ingeniería de la información.
- Definición del problema.
- Concepto de la solución.
- Definir los requerimientos del sistema.
- Definición de los casos de uso.
- Definición del modelo lógico (clases y datos).
- Definición del físico (componentes y datos).
- Desarrollo (interfaz, acceso a datos, negocios).
- Pruebas.

4.7.1 Planificación

Especificación Funcional

Diseño Conceptual

Requerimientos

Requerimientos Funcionales

En base al estudio preliminar, en este apartado se exhiben los requerimientos que el sistema deberá satisfacer, a continuación se describe dichos requerimientos.

REQ 1. El sistema deberá permitir registrar los servidores que van a participar en el proceso de sincronización.

REQ 2. El sistema deberá permitir capturar los cambios de cada servidor que se ha registrado.

REQ 3. El sistema deberá permitir subir la información de los cambios de cada uno de los servidores registrados.

REQ 4. El sistema deberá permitir mezclar los cambios de cada uno de los servidores.

REQ 5. El sistema deberá permitir sincronizar los servidores de bases de datos.

REQ 6. El sistema deberá permitir llevar una agenda de programación para la sincronización de datos.

Requerimientos No Funcionales

Aquí se muestran los requerimientos no funcionales, que a su vez son las características principales de la herramienta de sincronización.

REQ F1. Fácil de Manejar. La interfaz de usuario debe ser intuitiva para que de esta forma los usuarios nuevos del sistema se adapten rápidamente.

REQ F2. Fiabilidad. El sistema deberá manejar soluciones de respaldo en caso de desperfectos, permitiéndole estar siempre disponible.

REQ F3. Estabilidad. El sistema deberá estar configurado de tal forma que no presente situaciones de desperfecto continuas que afecten al desarrollo del trabajo continuo.

REQ F5. Flexibilidad. La aplicación deberá permitir diferentes entornos de configuración presentando la facilidad de migración de plataforma y escalabilidad.

Requerimientos de Interfaz de Usuario

Interfaz de Usuario

REQ I1. La aplicación contará con una interfaz de usuario intuitiva, fácil de aprender y manejar; para las mismas se puede utilizar el mouse y el teclado y además un manejo por ventanas.

Interfaz Hardware

REQ I2. La aplicación desarrollada deberá ser desplegada sobre la infraestructura de red que actualmente posee Contilibro.

Interfaz Software

REQ I3. El lenguaje de Programación en el cual se desarrollara la aplicación es Visual Studio .Net 2005.

REQ I4. Se utilizara como Base de Datos SQL Server 2000 para los sistemas satélite y MYSQL para la página web.

Protocolos de Comunicación

REQ I5. La comunicación entre el cliente y el servidor se realiza a través del protocolo http

Requerimientos del Sistema

REQ S1. El tiempo de respuesta a cualquier petición no debe superar los 7 segundos.

REQ S2. La aplicación deberá poder ser accesible las 24 horas del día, los 365 días del año.

REQ S3. El tiempo de respuesta del sitio web dependerá del hardware facilitado por Contilibro.

Actores

Nombre	Perfil	Tipo de Acceso	Descripción
UsuarioAdministrador	Administrador	Total	Es el usuario que tiene acceso a todo el sistema pero de forma única a la gestión de usuario del sistema.

Tabla IV.XXIII. Usuarios del Sistema

4.8 Casos de Uso y Escenarios

Diagramas de Casos de Uso

- Diagrama de Caso uso General

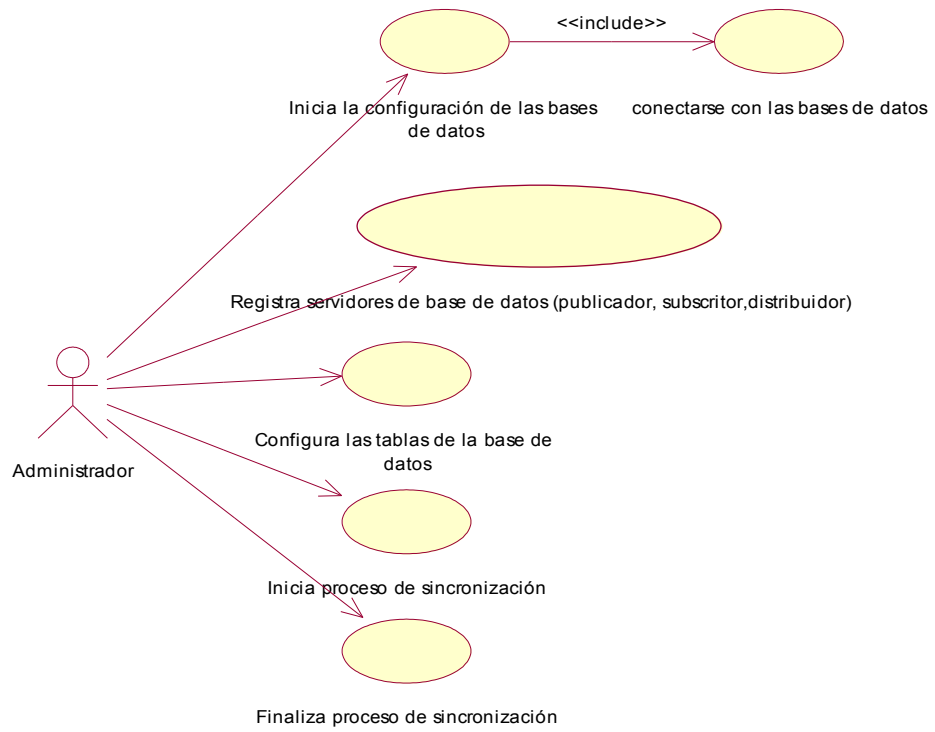


Figura IV.25. Diagrama de Caso de Uso General

- Diagrama de Caso de Uso Sincronización

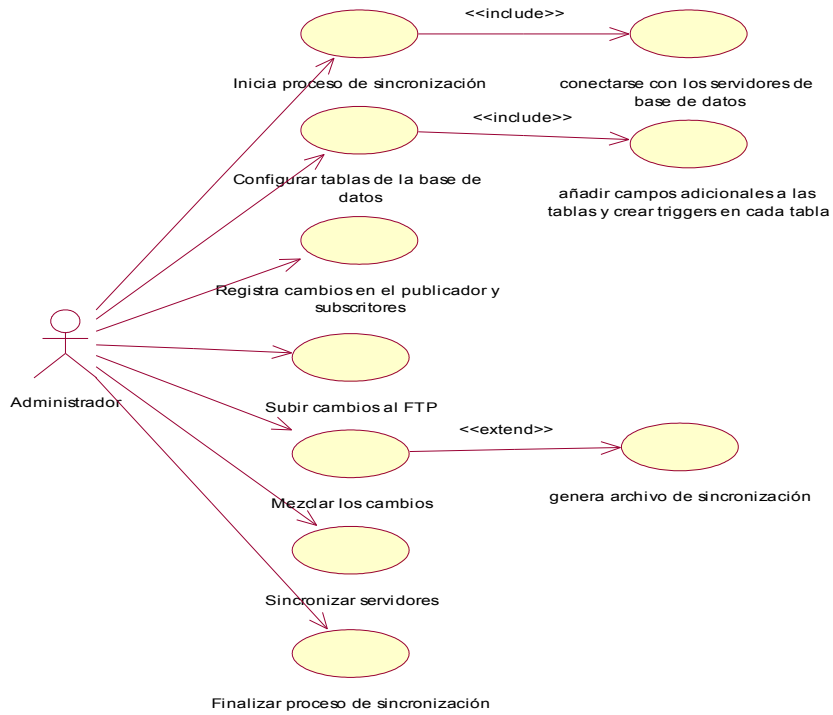


Figura IV.26. Diagrama de Caso Sincronización

- Diagrama de Caso de Uso Configurar Agenda

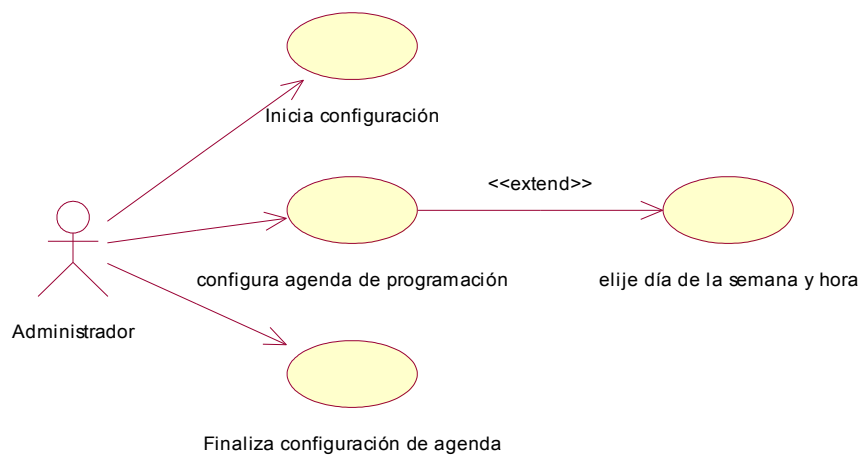


Figura IV.27. Diagrama de Caso de Uso Configurar Agenda

Casos de Uso

Caso de Uso: Iniciar Sesión

Identificador Caso de Uso	CU1
Nombre del Caso de Uso	Caso de Uso General
Actores	Usuario Administrador
Propósito	Configurar sincronización.
Visión General	El administrador ejecuta la herramienta que permite sincronizar bases de datos heterogéneas.
Tipo	Primario
Referencias	
Curso típico de eventos	
Acción del Actor	Respuesta del Sistema
1. Ingresa a la aplicación	2. Presenta el menú principal
	3. Solicita configurar base de datos
4. Configura base de datos	
Cursos Alternativos	
Línea 4: No se puede conectar con las bases de datos	

Tabla IV.XXIV. Caso de Uso General

Caso de Uso: Configurar Sincronización

Identificador Caso de Uso	CU2
Nombre del Caso de Uso	Configurar Sincronización
Actores	Usuario Administrador

Propósito	Configurar proceso de sincronización
Visión General	El usuario configura los servidores de base de datos para registrar el publicador, distribuidor y suscriptores y sincroniza las bases de datos
Tipo	Primario
Referencias	
Curso típico de eventos	
Acción del Actor	Respuesta del Sistema
1. Inicia proceso de sincronización.	2. Muestra menú de sincronización.
3. Registra servidores de base de datos	4. Registra publicador, distribuidor y suscriptores.
5. Elije tablas a publicar.	6. Añade campos adicionales a las tablas y crea triggers en cada tabla.
7. Solicita capturar cambios de las bases de datos	8. Crea archivos y sube al FTP.
9. Solicita mezclar cambios.	10. Crea archivo de configuración y sube al FTP.
11. Solicita sincronizar servidores.	12. Sincroniza los servidores.

Tabla IV.XXV. Caso de Uso Sincronización

Caso de Uso: Configurar Agenda

Identificador Caso de Uso	CU3
Nombre del Caso de Uso	Configurar agenda
Actores	Usuario Administrador
Propósito	Configurar agenda de sincronización

Visión General	El usuario ADMINISTRADOR solicita configurar agenda de sincronización, para esto se elije el día de la semana y la hora en el que se realizará el proceso de sincronización.	
Tipo	Primario	
Referencias		
Curso típico de eventos		
Acción del Actor	Respuesta del Sistema	
1. Solicita configurar agenda	2. Muestra opciones de configuración	
3. Elije opciones de configuración de la agenda.	4. Registra configuración	
Cursos Alternativos		
Línea 4: No se registro correctamente las opciones de configuración.		

Tabla IV.XXVI. Configurar Agenda

4.9 Glosario y Términos

Definiciones

- Gerente General de Contilibro. Persona que se encarga de la parte organizacional de la organización.
- Usuario. Persona (ente) a la cual se le brinda un servicio.

- Casos de Uso. Representación del flujo de información mediante los actores y los casos de uso correspondientes.
- MSF. Microsoft Solution Framework, metodología ágil empleada para el desarrollo de la aplicación.
- Servidor de Base de Datos. El servidor de la base de datos es un programa de computadora que proporciona servicios de la base de datos a otros programas o computadoras, puede también referirse a una computadora dedicada a funcionar tal programa.
- Base de Datos. Cualquier conjunto de datos organizados para su almacenamiento en la memoria de un ordenador o computadora, diseñado para facilitar su mantenimiento y acceso de una forma estándar.
- Interfaz. Punto en el que se establece una conexión entre dos elementos, que les permite trabajar juntos. La interfaz es el medio que permite la interacción entre esos elementos.
- Sistema Operativo. Software básico de una computadora. Tiene tres grandes funciones: coordina y manipula el hardware del ordenador o computadora, organiza los archivos en diversos dispositivos de almacenamiento y gestiona los errores de hardware y la pérdida de datos.

Acrónimos

- HW. Hardware.
- SW. Software.

4.10 Diseño Lógico

Tecnologías a Utilizar en el Proyecto

Herramientas

De acuerdo a la naturaleza de los requerimientos y la solución planteada para estos, se ha elegido a la tecnología Microsoft Visual Studio .NET para la implementación del proyecto. Además como metodología de desarrollo se ha elegido MSF, por cuanto es adaptable a cualquier tipo de proyecto, sea este grande, pequeño, y también se adapta al tiempo disponible para el desarrollo.

Aplicación Multicapas

Almacenes de Datos

SQL Server 2000 nos permitirá la implementación de los objetos necesarios para el soporte de la solución. Principalmente nos permite la elaboración de procedimientos almacenados que brindan mayor rapidez en la ejecución y facilidad de mantenimiento.

Capa de acceso a Datos

El acceso a datos se implementó utilizando ODBC, ya que nos permite la abstracción de la mayoría de transacciones que representan la ejecución de sentencias y la devolución de conjuntos de datos para el procesamiento interno de la aplicación.

Capa de lógica de Negocio

La lógica de negocio se implementará utilizando clases de .net elaborando componentes que permitan realizar las tareas funcionales de la empresa.

Capa de Presentación o Interfaz de Usuario

La aplicación se implementará utilizando c# de .NET.

Capa de Servicios Web

La herramienta hace uso de los servicios web para realizar todo el proceso de configuración de los servidores de base de datos que se han registrado.

4.10.1 Diagrama de Secuencia

A continuación se presenta el diagrama de secuencia de la herramienta de sincronización.

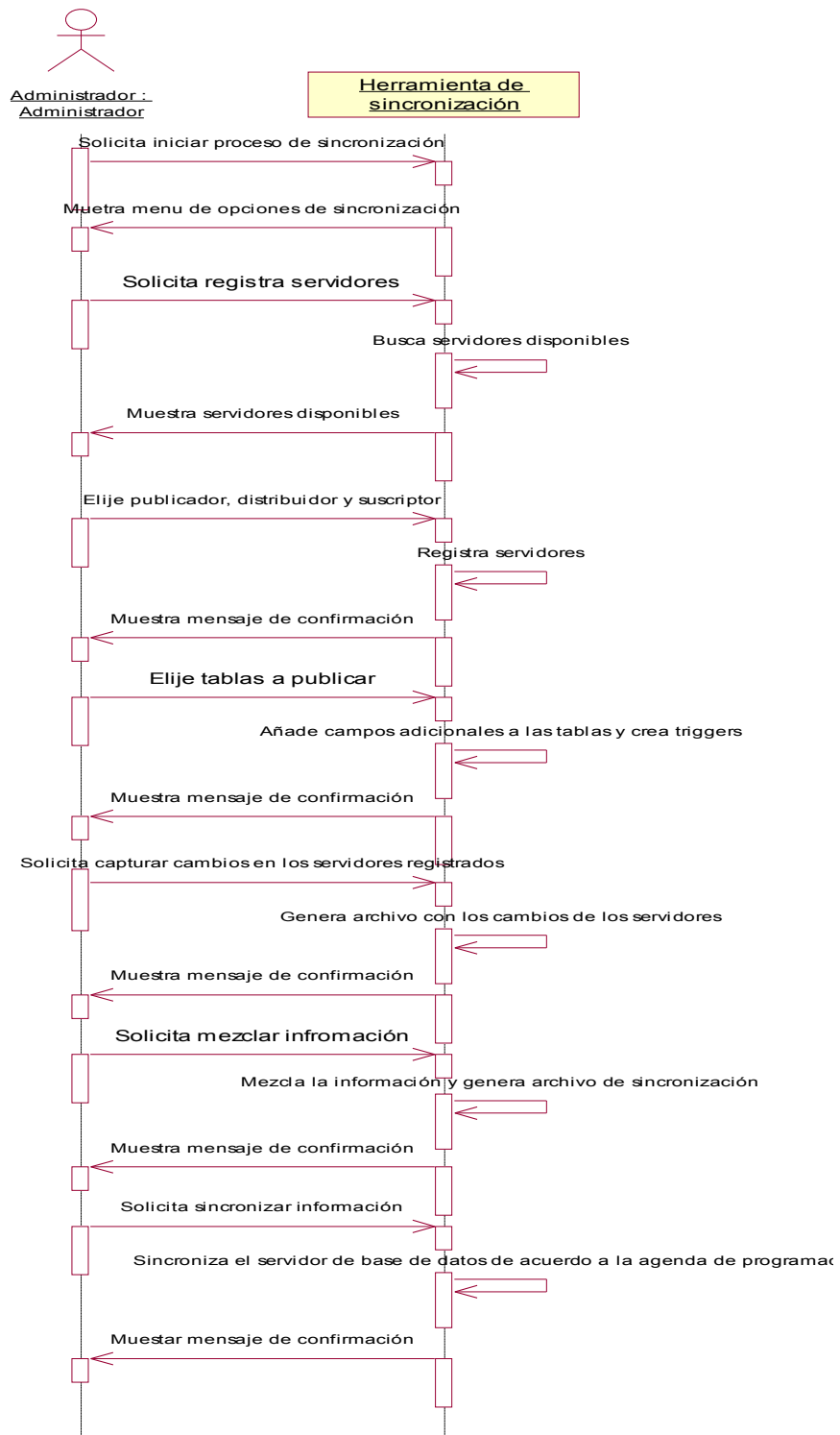


Figura IV.28. Diagrama de Secuencia Proceso de Sincronización

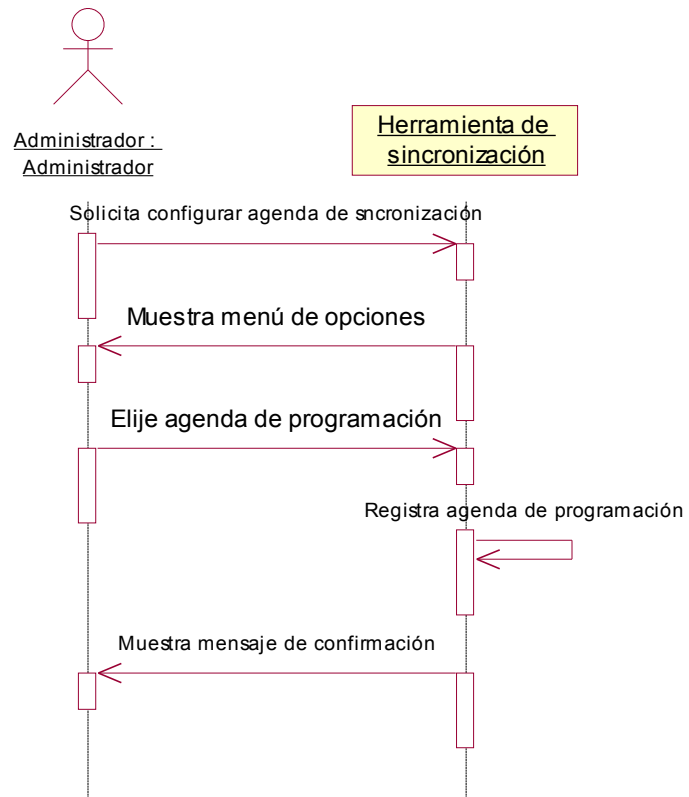


Figura IV.29. Diagrama de Secuencia Configurar Agenda

4.10.2 Diagrama de Componentes

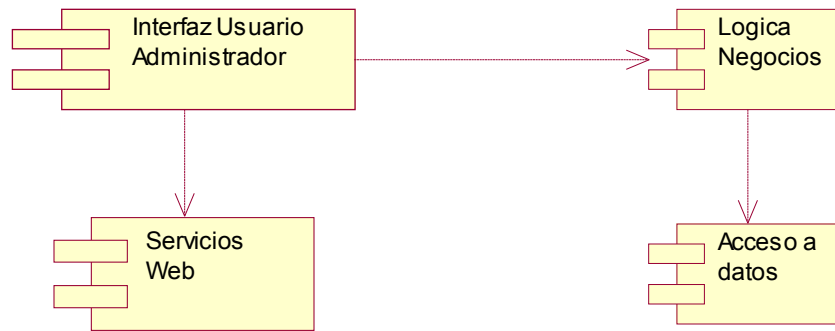


Figura IV.30. Diagrama de Componentes

4.11 Desarrollo

Nomenclatura y Estándares para el Desarrollo

Controles	AMBITO	
	Local o a nivel de procedimiento	Nivel de modulo o Form
Classes	Cls< NombreClase>	cls< NombreClase>
Formularios Win	frm< NombreFormulario>	IFrm< NombreFormulario>
Combobox	cmb< NombreCombo>	ICmb< NombreCombo>
Datagrid	dgv<NombreGrid>	IGrd<NombreGrid>
Buttons	Btn< NombreBoton>	IBtn< NombreBoton>
CheckBoxes	Chk< NombreCheck>	IChk< NombreCheck>
Textboxes	Txt< NombreText>	ITxt< NombreText>
Labels	Lb< NombreEtiqueta>	lB< NombreEtiqueta>
Panel	Pnl< NombreEtiqueta>	IPnl< NombreEtiqueta>
LinkButton	Lbtn< NombreBoton>	lIBtn< NombreBoton>

Tipos primitivos		
Integer	int< Nombre >	lInt< Nombre >
Long	lng< Nombre >	lLng< Nombre >
Boolean	bln< Nombre >	lBln< Nombre >
String	str< Nombre >	lStr< Nombre >
Double	dbl< Nombre >	lDbl< Nombre >
Constantes	C_< Nombre>	LC_< Nombre >

Tabla IV.XXVII. Tabla Nomenclatura de Programación

4.12 Diseño de Interfaces de Usuario

Las pantallas de la herramienta de sincronización de Contilibro están diseñadas de una manera intuitiva y fácil de entender.

Por lo que a continuación de muestran las capturas de pantallas de las funciones del sistema más representativas para cada acción descrita.

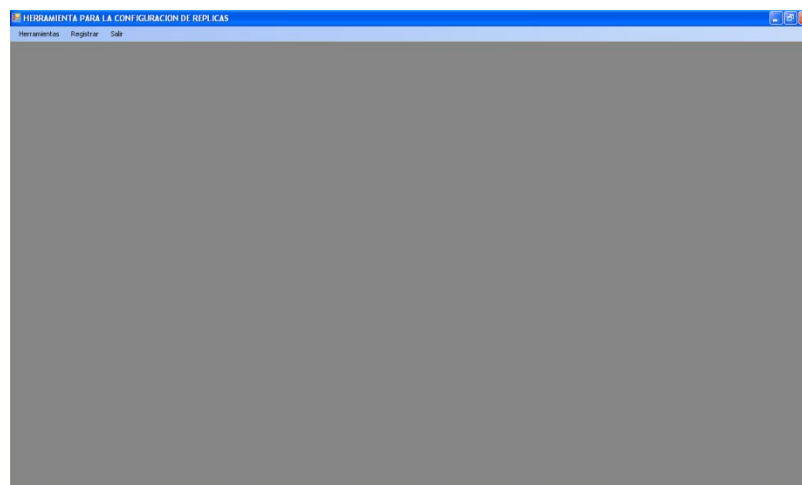


Figura IV.31. Pantalla menú principal

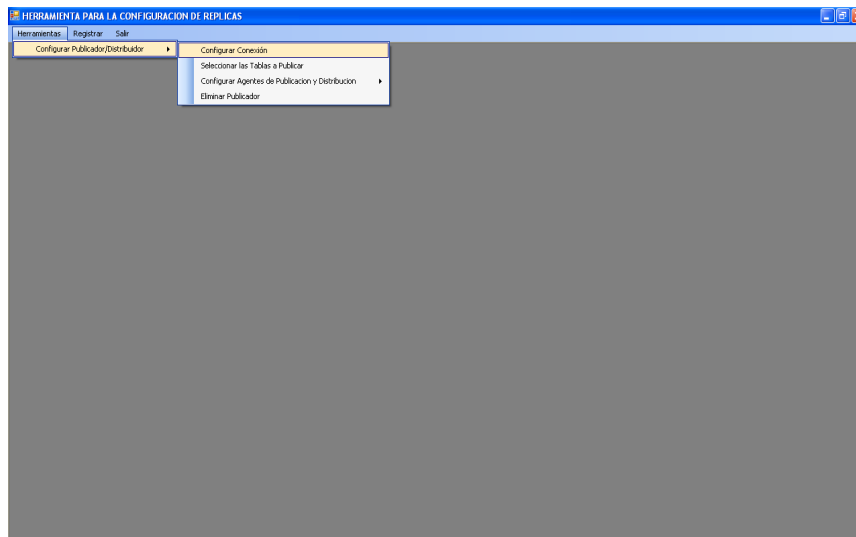


Figura IV.32. Pantalla menú publicador

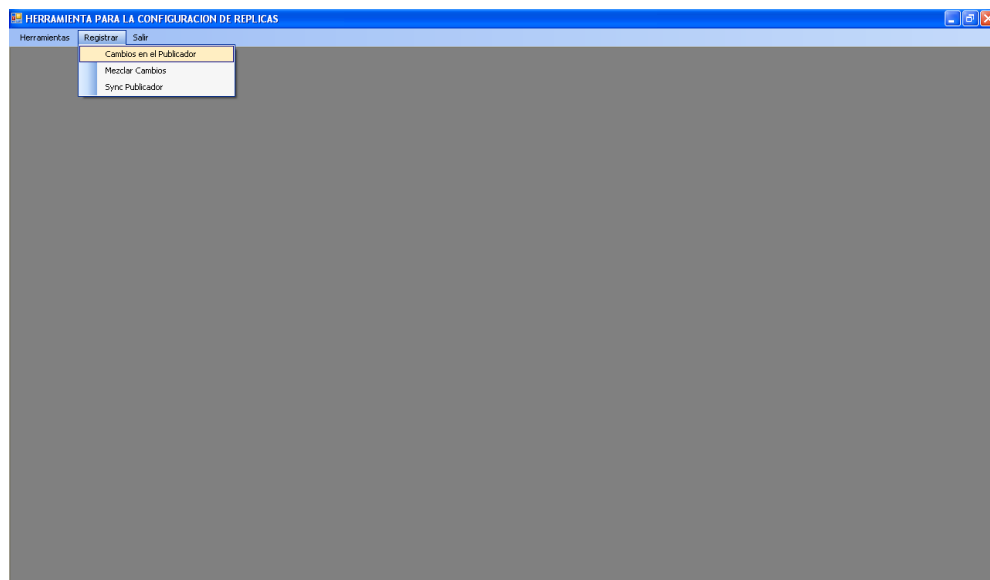


Figura IV.33. Pantalla menú registro servidores

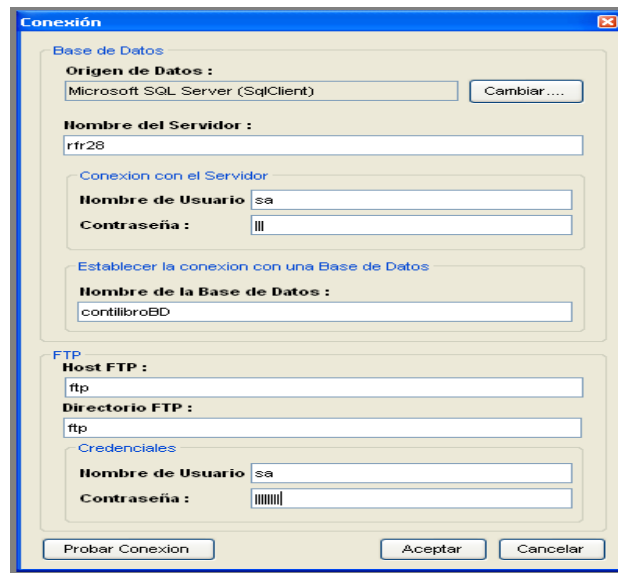


Figura IV.34. Pantalla configurar conexión.

4.12.1 Implementación del Acceso a Datos

El acceso a datos se lo realiza en la capa de acceso de la aplicación realizada en Microsoft Visual Studio 2005.

Se debe aclarar que para la conexión a la base de datos se utilizó la herramienta Enterprise Library, lo que me permite hacer procedimientos de manera general y no solo para una base de datos en específico.

Los procedimientos que aquí se muestran sirven para poder realizar cualquier tipo de tarea con la base de datos.

```
public void CrearComandoPA (String NombreStoredProcedure)
{
    this.comando = factory.CreateCommand();
    this.comando.Connection = this.conexion;
    this.comando.CommandType = CommandType.StoredProcedure;
    this.comando.CommandText = NombreStoredProcedure;
}
```

```

        if (this.transaccion != null)
        {
            this.comando.Transaction=this.transaccion;
        }
    }

    /// <summary>
    /// Obtiene la dirección del parámetro
    /// </summary>
    /// <param name="opcion">1: input, 2:output,
3:Input/Output</param>
    /// <returns>dirección del parámetro</returns>
    private ParameterDirection ObtenerDireccion(int opcion)
    {
        ParameterDirection direccion;

        switch (opcion)
        {
            case 1:
            {
                direccion = ParameterDirection.Input;
                break;
            }
            case 2:
            {
                direccion = ParameterDirection.Output;
                break;
            }
            case 3:
            {
                direccion = ParameterDirection.InputOutput;
                break;
            }
            default:
            {
                throw new BaseDatosException("El parámetro
dirección no es correcto recuerde que debe estar entre 1 y 3");
                break;
            }
        }
    }

```



```

    }
    return direccion;
}

public void AsignarParametroNuloAlPA(string nombreparametro, int
direccion)
{
    DbParameter parametro = factory.CreateParameter();
    parametro.ParameterName = nombreparametro;
    parametro.DbType = DbType.AnsiString;
    parametro.Value = DBNull.Value;
    parametro.Direction = ObtenerDireccion(direccion);
    this.comando.Parameters.Add(parametro);
}

public DataSet EjecutarConsulta(String nombreDataSet, String
nombreTabla)
{
    DataSet ds = new DataSet(nombreDataSet);
    this.adapter = factory.CreateDataAdapter();
    try
    {
        this.adapter.SelectCommand = this.comando;
        this.adapter.Fill(ds, nombreTabla);
    }
    catch (Exception)
    {
        throw new BaseDatosException("Error al llenar el
conjunto de Datos");
    }

    return ds;
}

```

La línea donde se realiza la conexión a la base de datos se lo realiza en el archivo App.config.

```

<connectionStrings>

    <add
name="WindowsAppContiLibro.Properties.Settings.contilibroBDConnectionStr
ing1"
        connectionString="Data Source=RFR28;Initial
Catalog=contilibroBD;User ID=sa;Password=sql1"
        providerName="System.Data.SqlClient" />
    </connectionStrings>

```

4.12.2 Diseño de Esquemas

Capa de Negocios

En la capa de negocios tenemos las clases con las que se trabajará para implementar el proceso de sincronización.

```

public class EncriptarBytes
{
    public String key;
    public PasswordDeriveBytes pass;

    public EncriptarBytes()
    {
        key = "$n%p@s5d3c#yq9t&x?|";
        pass = new PasswordDeriveBytes(key,
            new byte[] {0x49, 0x76, 0x61, 0x6e, 0x20, 0x4d,
                0x65, 0x64, 0x76, 0x65, 0x64, 0x65, 0x76});
    }
    public byte[] Encriptar(byte[] Datos, byte[] Key, byte[] IV)
    {
        MemoryStream ms = new MemoryStream();
        Rijndael alg = Rijndael.Create();
        alg.Key = Key;
        alg.IV = IV;
        CryptoStream cs = new CryptoStream(ms,
            alg.CreateEncryptor(), CryptoStreamMode.Write);
        try

```

```

    {
        cs.Write(Datos, 0, Datos.Length);
        cs.Close();
        byte[] encryptedData = ms.ToArray();
        return encryptedData;
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
}

public byte[] Desencriptar(byte[] Datos, byte[] Key, byte[] IV)
{
    MemoryStream ms = new MemoryStream();
    Rijndael alg = Rijndael.Create();
    alg.Key = Key;
    alg.IV = IV;
    CryptoStream cs = new CryptoStream(ms,
alg.CreateDecryptor(), CryptoStreamMode.Write);
    try
    {
        cs.Write(Datos, 0, Datos.Length);
        cs.Close();
        byte[] decryptedData = ms.ToArray();
        ms.Close();
        return decryptedData;
    }
    catch (Exception ex)
    {
        throw new Exception(ex.Message);
    }
}
}
}

```

Capa de Aplicaciones

Para esta capa se presentara partes de código de manera resumida ya que la mayoría del código es reutilizado.

Configurar Agenda

```

public partial class frmConfigAgenteSync : Form
{
    private ServiceController servicio;
    private ServiceController[] servicios;

    public frmConfigAgenteSync()
    {
        InitializeComponent();
    }

    private void frmConfigAgenteSync_Load(object sender, EventArgs
e)
    {
        BuscarServicio();
    }

    private void BuscarServicio()
    {
        int band = 0;
        int pos = 0;
        servicios =
(ServiceController[])ServiceController.GetServices();

        try
        {
            for (int i = 0; i < servicios.Length; i++)
            {
                if (servicios[i].ServiceName == "ServSync")
                {
                    band = 1;
                    pos = i;
                }
            }
            if (band == 1)
            {
                servicio = servicios[pos];
            }
        }
    }
}

```

```

        CargarInformacion();
    }
    else
        throw new Exception("El Agente de Sincronizacion no
esta instalado en el equipo y no se puede continuar con la configuración
de este servicio...");
    }
    catch (Exception ex)
    {
        //btnAceptar.Enabled = false;
        MessageBox.Show(ex.Message, "ADVERTENCIA",
MessageBoxButtons.OK);
    }
}

private void CargarInformacion()
{
    Config configuracion = new Config();
    txtOrigenDatos.Text = configuracion.Origen;
    txtServidor.Text = configuracion.Servidor;
    txtBaseDatos.Text = configuracion.BaseDatos;
    txtUsuario.Text = configuracion.Login;
    txtContrasena.Text = configuracion.Pass;
}

private void btnCancelar_Click(object sender, EventArgs e)
{
    Close();
}

private void btnAceptar_Click(object sender, EventArgs e)
{
    Config agenda = new Config();
    StringCollection Dias = new StringCollection();
    Dias.Clear();
    try
    {
        if (cmbSucede.SelectedIndex != 1)

```

```

{
    if (cmbSucede.SelectedIndex == 0)
    {
        agenda.SucedeSync = cmbSucede.Text;
        agenda.DiasSync = Dias;
    }
    else
    {
        agenda.SucedeSync = cmbSucede.Text;

        if (chkLunes.Checked)
            Dias.Add("1");
        if (chkMartes.Checked)
            Dias.Add("2");
        if (chkMiercoles.Checked)
            Dias.Add("3");
        if (chkJueves.Checked)
            Dias.Add("4");
        if (chkViernes.Checked)
            Dias.Add("5");
        if (chkSabado.Checked)
            Dias.Add("6");
        if (chkDomingo.Checked)
            Dias.Add("7");

        agenda.DiasSync = Dias;
    }
}
else
{
    cmbSucede.Focus();
    throw new Exception("Seleccione la frecuencia...");
}

if (agenda.SucedeSync == "Semanalmente" && !
chkLunes.Checked && !chkMartes.Checked && !chkMiercoles.Checked
    && !chkJueves.Checked && !chkViernes.Checked && !
chkSabado.Checked && !chkDomingo.Checked)
{

```

```

        chkLunes.Focus();
        throw new Exception("Seleccione uno o más días de la
semana.....");
    }

    agenda.FrecuenciaSync =
Convert.ToInt32(numFrecuencia.Value);
    agenda.FechaInicioSync =
Convert.ToDateTime(dtFechaInicio.Value.ToLongDateString());
    agenda.HoraInicioSync = dtHoraInicio.Value;
    agenda.FechaUltEjecSync = new
DateTime(agenda.FechaInicioSync.Year, agenda.FechaInicioSync.Month,
agenda.FechaInicioSync.Day,
                                agenda.HoraInicioSync.Hour,
agenda.HoraInicioSync.Minute, 0);
    agenda.FechaSigEjecSync = SiguienteEjecucion(agenda);
    agenda.Save();

    if (servicio != null)
    {
        servicio.Refresh();
        if (servicio.CanStop && servicio.Status ==
ServiceControllerStatus.Running)
        {
            servicio.Stop();
        }

        int numD = 0;
        if (agenda.DiasSync.Count > 0)
        {
            numD = agenda.DiasSync.Count;
        }
        string[] arg = new string[16 + numD];
        arg[0] = agenda.Proveedor;
        arg[1] = agenda.CadenaConexion;
        arg[2] = agenda.id_servidor.ToString();
        arg[3] = agenda.Servidor;
        arg[4] = agenda.DirectorioPC;
        arg[5] = agenda.FTP;
        arg[6] = agenda.UsuarioFTP;
        arg[7] = agenda.ClaveFTP;
    }

```

```

arg[8] = agenda.dirFTP_raiz;
arg[9] = agenda.dirFTP_serv;
arg[10] = agenda.SucedeSync;
arg[11] = agenda.FrecuenciaSync.ToString();
arg[12] = agenda.FechaInicioSync.ToString();
arg[13] = agenda.FechaSigEjecSync.ToString();
arg[14] = agenda.FechaUltEjecSync.ToString();
arg[15] = agenda.HoraInicioSync.ToString();

if (numD > 0)
{
    for (int i = 0; i < agenda.DiasSync.Count; i++)
    {
        arg[i + 16] = agenda.DiasSync[i];
    }
}

servicio.Start(arg);

MessageBox.Show("El Servicio de Sincronizacion se ha
configurado correctamente...", "AVISO", MessageBoxButtons.OK,
MessageBoxIcon.Information);

this.Close();
}
else
    throw new Exception("El Servicio de Sincronizacion
no esta instalado en el equipo y no se puede continuar con la
configuración de este servicio...");
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "ADVERTENCIA",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
}
}

private void cmbSucede_SelectedIndexChanged(object sender,
EventArgs e)
{
    if (cmbSucede.SelectedIndex == 0)
    {

```



```
lblFrecuencia.Text = "días";
numFrecuencia.Maximum = 364;

chkLunes.Enabled = false;
chkMartes.Enabled = false;
chkMiercoles.Enabled = false;
chkJueves.Enabled = false;
chkViernes.Enabled = false;
chkSabado.Enabled = false;
chkDomingo.Enabled = false;

chkLunes.Checked = false;
chkMartes.Checked = false;
chkMiercoles.Checked = false;
chkJueves.Checked = false;
chkViernes.Checked = false;
chkSabado.Checked = false;
chkDomingo.Checked = false;
}
else
{
    lblFrecuencia.Text = "semanas";
    numFrecuencia.Maximum = 52;

    chkLunes.Enabled = true;
    chkMartes.Enabled = true;
    chkMiercoles.Enabled = true;
    chkJueves.Enabled = true;
    chkViernes.Enabled = true;
    chkSabado.Enabled = true;
    chkDomingo.Enabled = true;
}
numFrecuencia.Enabled = true;
}

private DateTime SiguienteEjecucion(Config _agenda)
{
    Config agenda = _agenda;
```

```

DateTime dtUlt = agenda.FechaUltEjecSync;
DateTime dtSig = new DateTime();
DateTime dtActual = DateTime.Now;

dtSig = dtUlt;
/**/
if (agenda.SucedeSync == "Diariamente")
{
    while (dtActual >= dtSig)
    {
        dtSig = dtSig.AddDays(agenda.FrecuenciaSync);
    }
}

int dActual = 0;
int dSiguiente = 0;
DateTime dtFecha = new DateTime();
/**/
if (agenda.SucedeSync == "Semanalmente")
{
    dActual = DiaSemanaEntero(dtActual);

    int band = 0;

    for (int i = 0; i < agenda.DiasSync.Count; i++)
    {
        if (band == 0 &&
            (dActual < Convert.ToInt32(agenda.DiasSync[i])
            || (dActual == Convert.ToInt32(agenda.DiasSync[i]) && dtActual.TimeOfDay
            <= agenda.HoraInicioSync.TimeOfDay))
        {
            band = 1;
            dSiguiente =
Convert.ToInt32(agenda.DiasSync[i]);
        }
    }
    if (band == 0)
    {

```

```
        dSiguiente = Convert.ToInt32(agenda.DiasSync[0]);
    }

    dtFecha = dtActual.AddDays(dSiguiente dActual);

    dSiguiente = DiaSemanaEntero(dtFecha);
    dActual = DiaSemanaEntero(dtSig);
    dtSig = dtSig.AddDays(dSiguiente dActual);

    while (dtActual >= dtSig)
    {
        dtSig = dtSig.AddDays(agenda.FrecuenciaSync * 7);
    }
}
return dtSig;
}
private int DiaSemanaEntero(DateTime dt)
{
    int dia = 0;
    switch (dt.DayOfWeek.ToString())
    {
        case "Monday":
            dia = 1;
            break;
        case "Tuesday":
            dia = 2;
            break;
        case "Wednesday":
            dia = 3;
            break;
        case "Thursday":
            dia = 4;
            break;
        case "Friday":
            dia = 5;
            break;
        case "Saturday":
            dia = 6;
    }
}
```

```

        break;
    case "Sunday":
        dia = 7;
        break;
    }
    return dia;
}

private void btnAtras_Click(object sender, EventArgs e)
{
    this.Close();
    configuracionP.frmMostrarDatosSync miForm = new
WinFormSyncContilibro.matenimiento.configuracionP.frmMostrarDatosSync();
    miForm.MdiParent = ActiveForm;
    miForm.StartPosition = FormStartPosition.CenterScreen;
    miForm.Show();
}
}

```

4.13 Especificaciones de Seguridad

Diseño de Estrategias de Autorización, Autenticación

El diseño de la solución propuesta siempre mantendrá presente el perfil del usuario que se encuentra utilizando el sistema, cuidando que se le presenten únicamente las opciones necesarias y suficientes para el cumplimiento de sus actividades, cuidando de esta forma la integridad de la información que se almacene y por ende la integridad del sistema.

4.13.1 Estabilización

Las fases de Estabilización, se refieren a la puesta en marcha del sistema, lo que tiene que ver con el código fuente del sistema y base de datos todo con sus respectivos manuales.

Material de Capacitación

El sistema propuesto contempla distintos perfiles de usuario, con tareas y funciones claramente definidas, para las cuales siempre es necesario el respectivo material de capacitación. Se han propuesto los siguientes manuales, de acuerdo al ámbito operacional de cada actor del sistema:

Manual de Usuario.

Contiene la descripción de las funcionalidades generales del sistema, un manual de usuario debería poder ser entendido por cualquier usuario principiante, así como también serles útil a usuarios avanzados.

Manual de Instalación

El cual especifica los requerimientos y la configuración necesaria para la instalación del sistema, empezando por los recursos hardware, la plataforma operativa indicada para el despliegue, la configuración del entorno software necesario y el levantamiento de los servicios necesarios para el consumo desde los clientes definidos.

4.14 Instalación

4.14.1 Procedimientos y Procesos

El manual de procedimientos y procesos que posee la empresa debe ser actualizado tomando en cuenta a la nueva solución informática incorporada, para que de esta forma el personal operativo tenga una base firme de consulta sobre el nuevo curso de actividades a desarrollarse en la empresa.

4.14.2 Historial del Soporte Técnico

Todo aspecto importante efectuado durante ésta fase debe ser debidamente registrada, ayudando a manejar de forma centralizada las actividades que se han ido cumpliendo en base al tiempo previsto para la presente etapa.

CONCLUSIONES

Después de haber finalizado la investigación se ha llegado a las siguientes conclusiones:

- Se logró desarrollar e implementar una herramienta de replicación de mezcla sobre los motores de base de datos SQL Server y MySql, basado en cuatro componentes principales: un conjunto de Servicios Web, agentes (Servicios Windows), un catálogo de réplica y una consola que permite la administración y trabaja de manera independiente a las aplicaciones que deseen implementar réplica de datos.
- MySql y Sql Server utilizan el lenguaje de consulta estructurado o SQL lo que permite efectuar consultas con el fin de recuperar información de interés de una base de datos así como también hacer cambios sobre ella. El Servicio Web

encargado de generar estas sentencias SQL se basa en el estándar SQL99 para asegurar un correcto mapeo de los datos a replicar.

- La implementación de estos Servicios Web permiten el desarrollo de sistemas más robustos e interoperables y facilitan la tarea de integración de datos con los motores MySQL y Sql Server, independizando a las aplicaciones de los servicios de distribución que no están incluidos en versiones estándar de motores de base de datos, normalmente estos servicios *avanzados* tienen un costo adicional muy alto.
- La solución desarrollada cuenta con un conjunto de agentes que junto a los Servicios Web permiten realizar tareas que dependen del rol que cumpla el servidor. El agente de Control de Cambio permite recuperar la información que ha creado, alterado o eliminado de la base de datos, este servicio está presente en el Publicador y Suscriptor. El agente de Sincronización permite mantener consistente los datos entre servidores remotos, este servicio está presente en el Publicador y Suscriptor. Finalmente el agente de mezcla permite solucionar los conflictos que se presenten en el proceso de réplica, este servicio está presente en el Distribuidor. Todos estos servicios permiten que el proceso de réplica se desarrolle de manera automática siguiendo un calendario que es configurado en la consola de administración de réplica.
- La aplicación desarrollada para la facturación de libros en la empresa “Contilibro” ha permitido la automatización de este proceso y la inclusión de la herramienta de réplica desarrollada ha permitido mantener sincronizada la información que se

maneja en las distintas sucursales de esta empresa. El almacenamiento de la información en varios sitios ha aumentado la disponibilidad de datos y mejorado el rendimiento de las consultas globales.

- El almacenamiento de la información que se recupera de los servidores está a cargo de el Distribuidor que es este caso es el servidor FTP. Toda la información (excepto el catalogo de réplica) es encriptada y solo podrá ser desencriptada por la consola de administración manteniendo segura la información de los servidores.

RECOMENDACIONES

- La aplicación desarrollada cuenta con los controles necesarios para evitar inconvenientes en el proceso de replicación pero es conveniente configurar esta herramienta de forma adecuada especialmente en el calendario que deben seguir los agentes evitando cruces de tareas al momento de que los agentes realicen las tareas a las que fueron asignadas para que el proceso de réplica cumpla con las expectativas planteadas.
- Para poder hacer uso del modelo y su implementación se debe seguir un conjunto de lineamientos que permitan el modelado de tablas de fácil replicación con tipos de datos soportados por el estándar SQL 99 y se debe tomar en cuenta que el modelo desarrollado no soporta la replicación de campos auto incrementales ni de tablas que no posean una clave primaria declarada.

- El modelo de Servicios Web a pesar de mantener un alto grado de sincronización entre los motores de base de datos que se han incluido en este estudio se recomienda no usar esta herramienta si la mayor parte de las columnas de las tablas a replicarse almacenan datos de tipo tinyint, blob, tinyblob, mediumblob para evitar pérdida de información ocasionada porque los mencionados motores de base de datos manejan distintos rangos para estos tipos de datos.

RESUMEN

En el presente trabajo se implementó un modelo de agentes en servicios web para la replicación de bases de datos heterogéneas con el objetivo de transportar (réplica de datos) y mantener la misma información (sincronización de datos) en todas las sucursales de venta que posee la empresa “Contilibro”.

Para realizar este trabajo se utilizó el método deductivo directo–inferencia o conclusión inmediata ya que con este método podremos descubrir consecuencias desconocidas, de principios conocidos. Para la recopilación de la información necesaria que sustente el presente trabajo de investigación, se ha realizado entrevistas en el lugar donde se desarrollo la aplicación y se usaron fuentes de información como libros, revistas, páginas web, etc.

Este modelo está compuesto por un conjunto de Servicios Web desarrollados en Visual Studio .Net que se encuentran alojados en un Servidor Web que cumplen con la tarea de generar sentencias SQL, servidores de base de datos que cumplen el rol de Publicador y Suscriptor, un servidor FTP que actúa como Distribuidor y un catálogo que almacena información generada en el proceso de replicación; además cuenta con una aplicación de escritorio que permite realizar tareas administrativas, independientemente del motor de base de datos que se esté usando.

El modelo propuesto alcanza un 95.33% de sincronización usando MySql como publicador y un 89.09% de sincronización usando SQL Server como publicador obteniendo un 92.23% de

sincronización global indistintamente del motor de base de datos usado. Los tipos de datos usados para estas pruebas fueron: numéricos exactos, numéricos aproximados, cadena de caracteres, datos binarios, fecha y hora.

La implementación de este modelo permite el desarrollo de aplicaciones más robustas e interoperables que facilita la integración de datos cuando existen diferentes motores de base de datos involucrados aumentando la disponibilidad y mejorando el rendimiento de consultas de datos en cada sucursal de venta.

Finalmente para el mejor rendimiento de este modelo se recomienda usar los tipo de datos que alcanzaron un mayor porcentaje de sincronización estos son: int, varchar, image, decimal y datetime.

SUMMARY

In this paper we implemented a model of agents in Web services for the replication of heterogeneous databases in order to transport (data replication) and keep the same information (data synchronization) on all sales branches having the company "Contilibro".

To make this work we used the direct-deductive inference or conclusion immediately and that this method can discover unknown consequences of known principles. For the collection of information necessary to substantiate this research, has conducted interviews at the place where the application was developed and used information sources such as books, magazines, websites, etc..

This model consists of a set of Web services developed in Visual Studio. Net that are hosted on a Web server that meet the task of generating SQL statements, database servers to fulfill the role of Publisher and Subscriber, a server FTP acts as Distributor and catalog stores information generated in the replication process, also has a desktop application that allows to perform administrative tasks, regardless of database engine being used.

The proposed model achieves a 95.33% synchronization using MySql as a Publisher and a 89.09% synchronization using SQL Server as a publisher to obtain a global

synchronization 92.23% regardless of the database engine used. The types of data used for these tests were accurate numerical, approximate numeric, string, binary data, date and time.

The implementation of this model allows the development of more robust and interoperable applications that facilitates the integration of data when different database engines involved increasing the availability and improving query performance data in each branch of sale.

Finally, for the best performance of this model is recommended to use the data type that reached a higher percentage of sync they are: int, varchar, image, decimal and datetime.

ANEXOS

ANEXO A

DESARROLLO DEL SISTEMA SATÉLITE DE CONTILIBRO

DATOS GENERALES

IDENTIFICACION DEL TEMA

✓ DENOMINACIÓN

TÍTULO: "Aplicación Satélite para llevar el control de inventarios y facturación de Contilibro "

Nombre del Producto: "Sistema Satélite Contilibro"

Propietario: Contilibro.

Área Automatizado: Área de Facturación e inventarios de Contilibro.

✓ CLASIFICACIÓN

Área: Informática

Sector: Desarrollo de Aplicaciones de Escritorio

Introducción

El presente capítulo se genera debido a la imperiosa necesidad de documentar técnicamente la aplicación satélite de Contilibro que permite llevar un control de inventarios y de facturación la misma que se desarrolló bajo la metodología MSF.

Para empezar se documenta la fase de Definición donde se delimita la visión y el ámbito del proyecto que cumplirá los objetivos del cliente, información obtenida de las entrevistas realizadas al Señor Héctor Tello gerente general de Contilibro, así como al personal que ahí labora, lo que ayudará a organizar el proyecto. A su vez en esta etapa se habla acerca de un calendario maestro el mismo que nos indicará cuando y que tiempo nos tomaremos en realizar cada actividad o tarea.

En la segunda fase Planificación, se realizará las especificaciones funcionales, describiendo lo que se va a desarrollar, incluyendo contenido como metas de diseño, requerimientos, características y dependencias. Esto es que aquí se establecen los conceptos que especifican las necesidades de los usuarios. Organiza los componentes de la solución. Así como también se especifica las restricciones tecnológicas de la solución.

En la fase de Desarrollo, se refiere al desarrollo del proyecto, es donde se documenta el código, nomenclatura, interfaces y estrategias utilizadas en el desarrollo del proyecto.

En la cuarta fase de Estabilización trata de la liberación del producto, se refiere al material de capacitación con que debe contar el proyecto.

La quinta fase Instalación, describe el plan que se debe poner en marcha en el momento de la implementación del sistema.

Planificación y Análisis: Documento SRS

Definición

Definición del problema

La empresa Contilibro, no cuenta con una aplicación que le permita llevar el control de inventarios y al mismo tiempo que ayude también a la facturación.

El problema principal radica en el manejo de la información de los procesos que realizan en el trabajo diario, es decir, que no tiene ningún tipo de sistema que maneje este tipo de información.

El método que utiliza el personal que labora en la empresa para el manejo de la información, es llevar a mano la facturación y el control de inventarios por lo que consideraron necesario automatizar estos procesos para aligerar el control de los mismos.

El no contar con un sistema que automatice los procesos anteriormente mencionados ha dado lugar a la aparición de problemas tales como:

- ✓ La información acerca de los libros que posee Contilibro no se encuentra debidamente actualizada, por ejemplo no se tiene con exactitud el stock o la existencia de libros en cada una de las sucursales.
- ✓ La información de los clientes y proveedores tampoco se encuentra debidamente actualizada.
- ✓ No se lleva un control adecuado en lo que se refiere a la facturación y ventas de los libros.
- ✓ Dificultades para crear reportes solicitados por los clientes, por ejemplo generar proformas de los libros que ofrece Contilibro.

Visión del Proyecto

La solución que se plantea para la empresa Contilibro, es la creación de un sistema satélite que permita llevar el control de inventarios y facturación; así como también permita llevar un control de proveedores y clientes de la empresa, el sistema ofrece resolver los problemas antes mencionados con los siguientes beneficios.

- ✓ El proceso de facturación y de inventario se llevará de manera automática.
- ✓ La información de clientes, productos(libros) y proveedores estará debidamente actualizada.

- ✓ Consultas y Reportes de cualquier tipo de información que se encuentre ingresada en el sistema.
- ✓ El sistema tendrá la capacidad de crear tipos de usuarios, lo que permitirá administrar de manera correcta el sistema.
- ✓ La información que se maneje en la empresa será más real y con un porcentaje muy bajo de errores.

Perfiles de Usuarios

A continuación se define brevemente una descripción de cada uno de ellos:

No.	Nombre	Perfil	Tipo de Acceso	Descripción
1	Usuario Administrador	Administrador	Total	Es el usuario que tiene acceso a todo el sistema pero de forma única a la gestión de usuario del sistema.
2	Usuarios vendedores	Ventas, facturación	Parcial	Tiene los permisos para generar reportes y realizar ventas de los libros.

Ámbito del Proyecto

Alcance

El sistema satélite de Contilibro permitirá el ingreso de datos de libros, clientes, proveedores además de llevar un control de inventario y de facturación, obteniendo del sistema los siguientes resultados:

- ✓ Ingreso de todos los datos necesarios para la creación de un libro, cliente, proveedor.
- ✓ Reportes de libros en existencia, clientes y proveedores.
- ✓ Facturación de las ventas realizadas por los clientes.

Requerimientos Generales

Para definir los requerimientos de los usuarios, se ha clasificado a los mismos en funcionales y no funcionales, a los cuales se les aplicó técnicas de investigación científica como entrevistas.

Entrevista

Se realizó una entrevista al Gerente General de Contilibro, con el fin de recoger la información necesaria para obtener una visión amplia del funcionamiento de los procesos que se realizan en Contilibro.

Resultados de las Entrevistas

La incidencia de los problemas que existen en la realización de los trámites se detalla a continuación.

- ✓ Para mantener toda la información organizada de las ventas que se realizan en la empresa se realizan procesos manualmente.
- ✓ No tienen una manera automática de generara reportes solicitados por el cliente.

Observando los problemas detallados anteriormente se puede concluir que no existe ningún sistema para el control de inventarios y facturación y por lo tanto es necesario el desarrollo de un sistema capaz de resolver dichos problemas.

Requerimientos

Los requerimientos planteados para la creación del sistema satélite de Contilibro, se describen a continuación:

REQ 1. El sistema deberá permitir gestionar la información de los usuarios que podrán administrar y usar el mismo.

REQ 2. El sistema deberá permitir gestionar la información de las sucursales de Contilibro.

REQ 3. El sistema deberá permitir gestionar la información de las áreas académicas de los libros.

REQ 4. El sistema deberá permitir gestionar la información de las categorías de los libros.

REQ 5. El sistema deberá permitir gestionar la información de un cliente.

REQ 6. El sistema deberá permitir el gestionar la información de un proveedor.

REQ 7. El sistema deberá permitir gestionar la información de las ventas directas de los libros.

REQ 8. El sistema deberá permitir gestionar la información de las ventas a consignación de los libros.

REQ 9. El sistema deberá permitir gestionar la información de las proformas que se realicen en cada una de las Sucursales de Contilibro.

REQ 10. El sistema deberá permitir gestionar la información de las devoluciones que se realicen.

REQ 11. El sistema deberá gestionar la información de una factura.

REQ 12. El sistema deberá permitir generar proformas de los libros por cada sucursal.

Concepto de la Solución

El modelo de arquitectura que se escogió para el desarrollo de la aplicación, es Cliente – Servidor.

Esta arquitectura utiliza capa de lógica de negocios, en donde se comunica con la de acceso a datos utilizando el Data Access Application Block que es una clase de

la Enterprise Library que contiene métodos para acceder a los datos y llamando a procedimientos almacenados.

Uso de Herramientas

Las herramientas software utilizadas en el desarrollo del sistema satélite se describen en la tabla V.II.

SOFTWARE	Descripción
Plataforma Operativa de Funcionamiento	Sistema Operativo Windows XP Profesional que nos servirá para la implementación de los servidores de aplicación y base de datos.
Motor de Base de Datos	Se utilizará el motor de base de datos SQL Server 2000 el cual brinda la posibilidad de elaborar Bases de Datos Relacionales y el manejo de procedimientos almacenados para las diferentes transacciones.
Herramientas de Desarrollo	Entre las Herramienta de Desarrollo se ha seleccionado el Microsoft Visual Studio 2005 que permite el trabajo sobre la plataforma .net.
Manejador de Reportes	Los reportes se crearan con el gestor de reportes propio de Microsoft Visual Studio, Microsoft Crystal Reports.
Herramientas de Modelado UML	Para la elaboración de ciertos tipos de diagramas que permitirán el análisis se utilizará Rational Rose.
Herramientas de oficina	Se utilizará la suite OFFICE 2007 para la elaboración de la documentación básica sobre el desarrollo del proyecto.
Herramientas para Administración y diseño de base de Datos	Se utilizará la suite de SQL Server 2000.

Planteamiento de la Arquitectura de la Aplicación

Se propone una arquitectura que consta de 5 capas, descritas en la siguiente tabla.

Nombre de las Capas utilizadas en el Sistema Web	Descripción
INTERFAZ DE USUARIO (ESCRITORIO)	Se efectuará la alimentación de la información a través de esta capa hacia la base de datos.
REGLAS DE NEGOCIO (CAPA 3)	Las reglas de negocio utilizarán los servicios de la capa de acceso para efectuar los procesos requeridos y presentar la información al usuario final.
ACCESO A DATOS (CAPA 2)	La capa de acceso a datos representará a todas las clases propuestas para el sistema.
BASE DE DATOS (CAPA 1)	En el que se elaborará una base de datos relacional, para soportar la lógica del negocio. Aquí se brindará acceso a la información a través de procedimientos almacenados, debido a su rapidez de ejecución.

Objetivos del Proyecto

Objetivos del Negocio

- ✓ Disminuir el tiempo de respuesta, al acceder a la información a través del sistema.
- ✓ Aumenta el tiempo y la funcionalidad del usuario, con la generación de cálculos y reportes automáticos de la información ingresada al sistema.
- ✓ Mantener la información debidamente actualizada y lista en cualquier momento.

Objetivos del Diseño

- ✓ Lograr que la información que se maneje dentro de la organización pueda ser consultada por el personal autorizado.
- ✓ Disminuir el tiempo de operación y respuesta en la ejecución de los procesos.

- ✓ Brindar interfaces de usuario amigables e intuitivas que faciliten el trabajo cotidiano de los encargados en los diferentes procesos de la empresa.

Factores Críticos

Análisis de Riesgos

El análisis de riesgos se presenta una identificación preliminar en busca de la eliminación de trastornos durante el proyecto, una mayor habilidad de seguir y controlar los mismos y la confianza de la Planificación de elementos de riesgo.

Listado de Riesgos

Lista de Riesgos	
ID	Descripción
RI-1	El equipo de desarrollo puede no adaptarse rápidamente a las tecnologías de .net.
RI-2	Los desarrolladores pueden tener retrasos en el cronograma de desarrollo debido a un análisis inadecuado del sistema para Contilibro.
RI-3	Al estar conectados los equipos de desarrollo al internet su vulnerabilidad es expuesta a los diferentes tipos de peligro existentes en la gran red de información.
RI-4	En los equipos de desarrollo se puede presentar daños imprevistos en el hardware.
RI-5	Cambios en los requerimientos por parte del Gerente General de Contilibro.
RI-6	Falta de información y requerimientos mal planteados por parte del personal y el gerente general de Contilibro.
RI-7	Resistencia de los usuarios finales al uso del sistema una vez finalizado.
RI-8	Desacuerdos con Contilibro., con respecto al sistema satélite.

Tabla de Valores

Los valores con los cuales se representaran a los riesgos de la tabla V.4., se definen en la tabla V.5.

Porcentaje	Descripción	Valor
1% - 33%	Alto	3
34% - 67%	Medio	2
68% - 99%	Bajo	1

Análisis de Riesgos

ID	Probabilidad			Impacto	
	%	Valor	Probabilidad	Valor	Impacto
RI-1	50	2	Medio	1	Bajo
RI-2	80	3	Alto	2	Medio
RI-3	40	2	Medio	1	Bajo
RI-4	30	2	Medio	1	Bajo
RI-5	50	2	Medio	2	Medio
RI-6	90	3	Alto	2	Medio
RI-7	40	2	Medio	2	Medio
RI-8	30	1	Bajo	1	Bajo

Resultados

Exposición al Riesgo		
ID	Valor	Exposición
RI-2	6	Alto
RI-6	6	Alto
RI-5	4	Medio
RI-7	4	Medio
RI-1	2	Bajo
RI-3	2	Bajo
RI-4	2	Bajo
RI-8	1	Bajo

Como resultado de éste análisis se toma la decisión de gestionar los riesgos con exposición Media y Alta.

Restricciones

Listado de Restricciones	
ID	Descripción
RI-1	El equipo de desarrollo puede no adaptarse rápidamente a las tecnologías de .net.
RI	Al estar conectados los equipos de desarrollo al internet su vulnerabilidad es expuesta a los diferentes tipos de peligro existentes en la gran red de

-3	información.
RI -4	En los equipos de desarrollo se puede presentar daños imprevistos en el hardware.
RI -8	Desacuerdos con Contilibro., con respecto al sistema satélite.

Planificación Inicial

Recursos Humanos

Integrantes y sus Funciones	
Nombre	Función
Ángel Apina, Richard Robles	Administrador del Producto
Ángel Apina, Richard Robles	Administrador del Programa
Ángel Apina, Richard Robles	Equipo de desarrollo
Personal de Contilibro.	Equipo analista de Pruebas y Calidad
Ángel Apina, Richard Robles	Equipo encargado del despliegue del Proyecto

Recursos Físicos

Numero	Recursos Físicos (Sw / Hw)
	Detalle
2	Computadora Personal Intel core 2 Duo 2.13 GHZ
2	Impreso / Scanners Lexmark

1	Sistema operativo Windows XP
1	Microsoft Visual Studio 2005
1	Servidor de Aplicaciones IIS
1	Servidor de Base de Datos SQL Server 2000
1	Suite de Oficina Office 2007.

Planificación de Actividades

El tiempo que se tomo para la realización del sistema Satélite de Contilibro desde su planificación hasta las pruebas finales fue un estimado de un año y 5 meses, las actividades generales desarrolladas en el transcurso del tiempo mencionado se encuentran detalladas de la siguiente manera:

- ✓ Ingeniería de la información.
- ✓ Definición del problema.
- ✓ Concepto de la solución.
- ✓ Definir los requerimientos del sistema.
- ✓ Definición de los casos de uso.
- ✓ Definición del modelo lógico (clases y datos).
- ✓ Definición del físico (componentes y datos).
- ✓ Desarrollo (interfaz, acceso a datos, negocios).
- ✓ Pruebas.

Planificación

Especificación Funcional

Diseño Conceptual

Requerimientos

Requerimientos Funcionales

En base al estudio preliminar, en este apartado se exhiben los requerimientos que el sistema deberá satisfacer, a continuación se describe dichos requerimientos.

Para el mejor análisis de requisitos se les ha dividido en módulos que serán explicados a continuación.

Modulo de Mantenimiento

En este modulo estarán todos y cada uno de los ítems necesarios para poder llevar el control de inventarios y de facturación de Contilibro. Por lo que a este modulo se lo ha dividido en 2 partes como:

Mantenimiento de Usuarios

REQ 1. El sistema deberá permitir el ingreso, modificación y eliminación de Usuarios.

Mantenimiento de Libros, Clientes, Proveedores, Areas, Categorías

Que son los ítems que se necesitan para la creación de una factura y proforma.

REQ 2. El sistema deberá permitir el ingreso, modificación una área.

REQ 3. El sistema deberá permitir el ingreso, modificación de una categoría.

REQ 4. El sistema deberá permitir el ingreso, modificación de un libro.

REQ 5. El sistema deberá permitir el ingreso, modificación de un cliente.

REQ 6. El sistema deberá permitir el ingreso, modificación de un proveedor

Modulo de Facturación-Ventas

Aquí se llevará el control de la facturación y ventas de las sucursales de Contilibro.

REQ 7. El sistema deberá permitir el registro de una venta directa de libros.

REQ 8. El sistema deberá permitir el registro de una venta a consignación de libros.

Modulo de Consultas

En el modulo de consultas se encuentran todas y cada una de las consultas requeridas por el usuario final, las cuales presentan reportes con la información necesaria.

REQ 9. El sistema deberá permitir mostrar consultas de los libros con los que cuenta Contilibro.

REQ 10. El sistema deberá permitir mostrar consultas de los clientes de Contilibro.

REQ 11. El sistema deberá permitir mostrar consultas de los proveedores de Contilibro.

REQ 12. El sistema deberá permitir realizar proformas.

Requerimientos No Funcionales

Aquí se muestran los requerimientos no funcionales, que a su vez son las características principales del sistema a desarrollar.

REQ F1. Fácil de Manejar.- La interfaz de usuario debe ser intuitiva para que de esta forma los usuarios nuevos del sistema se adapten rápidamente.

REQ F2. Fiabilidad.- El sistema deberá manejar soluciones de respaldo en caso de desperfectos, permitiéndole estar siempre disponible.

REQ F3. Estabilidad.- El sistema deberá estar configurado de tal forma que no presente situaciones de desperfecto continuas que afecten al desarrollo del trabajo continuo.

REQ F4. Seguridad.- El acceso a la información pertinente de cada perfil de usuario se debe manejar utilizando sesiones y sistemas de autenticación y autorización.

REQ F5. Integridad.- La información debe ser alimentada a la base de datos pasando por procesos de validación.

REQ F6. Flexibilidad.- La aplicación deberá permitir diferentes entornos de configuración presentando la facilidad de migración de plataforma y escalabilidad.

Requerimientos de Interfaz de Usuario

Interfaz de Usuario

REQ I1. La aplicación contará con una interfaz de usuario intuitiva, fácil de aprender y manejar; para las mismas se puede utilizar el mouse y el teclado y además un manejo por ventanas.

Interfaz Hardware

REQ I2. La aplicación desarrollada deberá ser desplegada sobre la infraestructura de red que actualmente posee Contilibro.

Interfaz Software

REQ I3. El lenguaje de Programación en el cual se desarrollara la aplicación es Visual Studio .Net 2005.

REQ I4. Se utilizara como Base de Datos SQL Server 2000.

Protocolos de Comunicación

REQ I5. La comunicación entre el cliente y el servidor se realiza a través del protocolo http

Requerimientos del Sistema

REQ S1. El tiempo de respuesta a cualquier petición no debe superar los 7 segundos.

REQ S2. La aplicación deberá poder ser accesible las 24 horas del día, los 365 días del año.

REQ S3. El tiempo de respuesta del sitio web dependerá del hardware facilitado por Contilibro.

Actores

Nombre	Perfil	Tipo de	Descripción
--------	--------	---------	-------------

		Acceso	
Usuario Administrador	Administrador	Total	Es el usuario que tiene acceso a todo el sistema pero de forma única a la gestión de usuario del sistema.
Usuarios Vendedores	Administrador	Parcial	Tiene casi todos los permisos que el usuario Administrador a excepción de la gestión de usuario y eliminaciones de ciertos ítems.

Casos de Uso y Escenarios

Diagramas de Casos de Uso

- ✓ Diagrama de Caso de Uso General

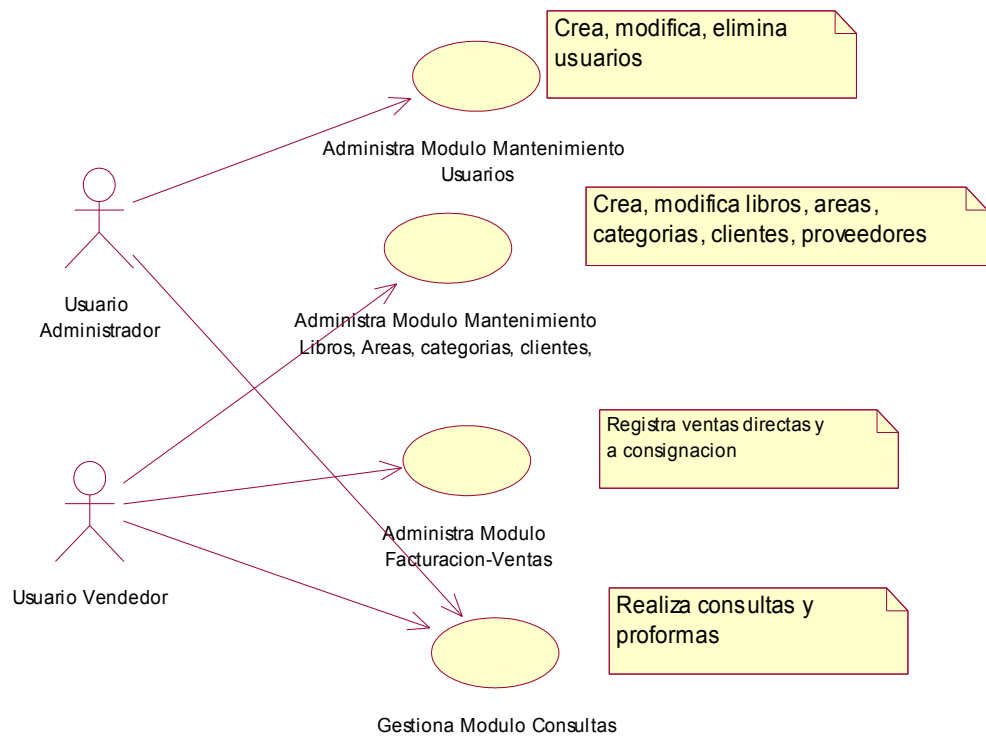
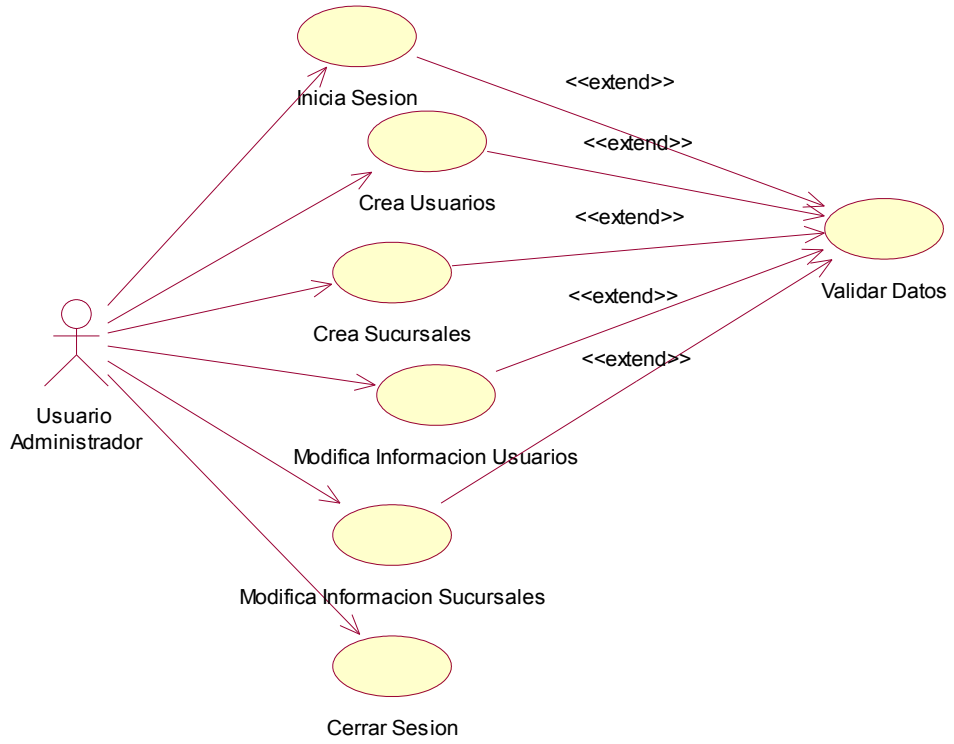


Figura V.1. Diagrama de Caso de Uso General

Elaborado por: Ángel Apina, Richard Robles

- ✓ Diagrama de Caso de Uso Usuario Administrador



✓ Diagrama de Caso de Uso Usuario Vendedor

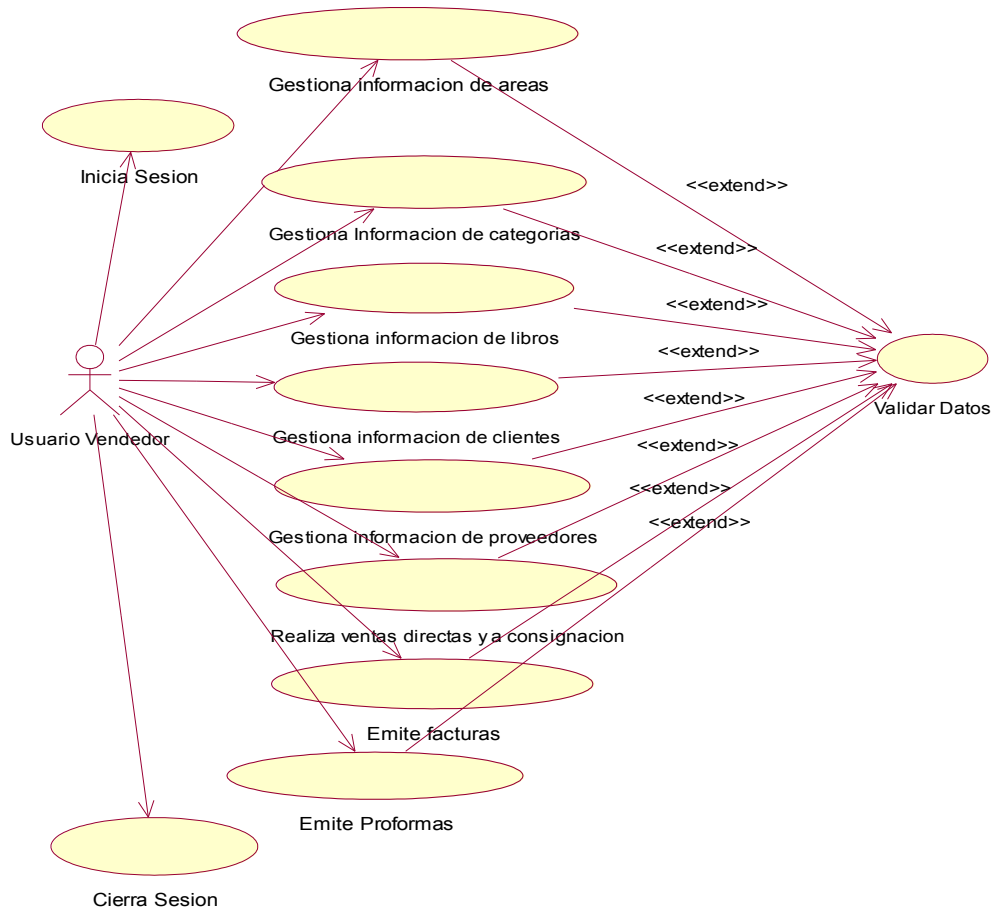
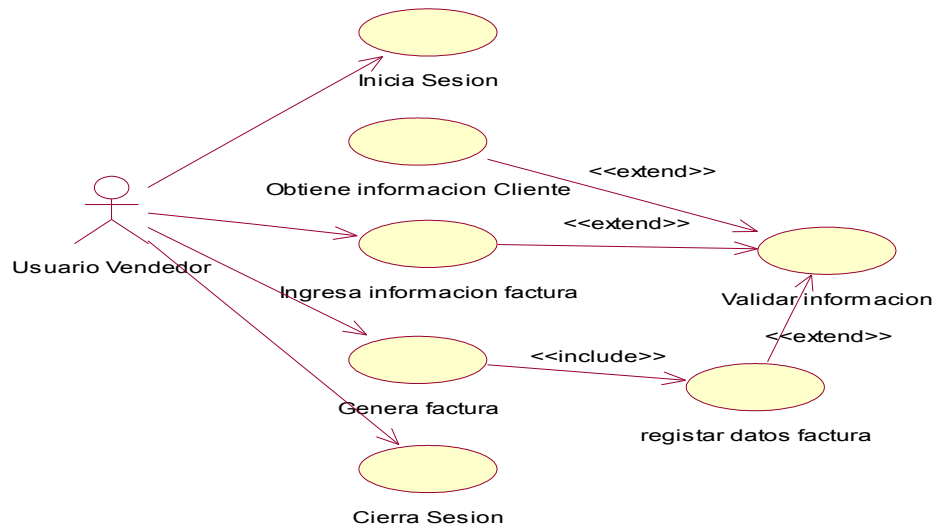


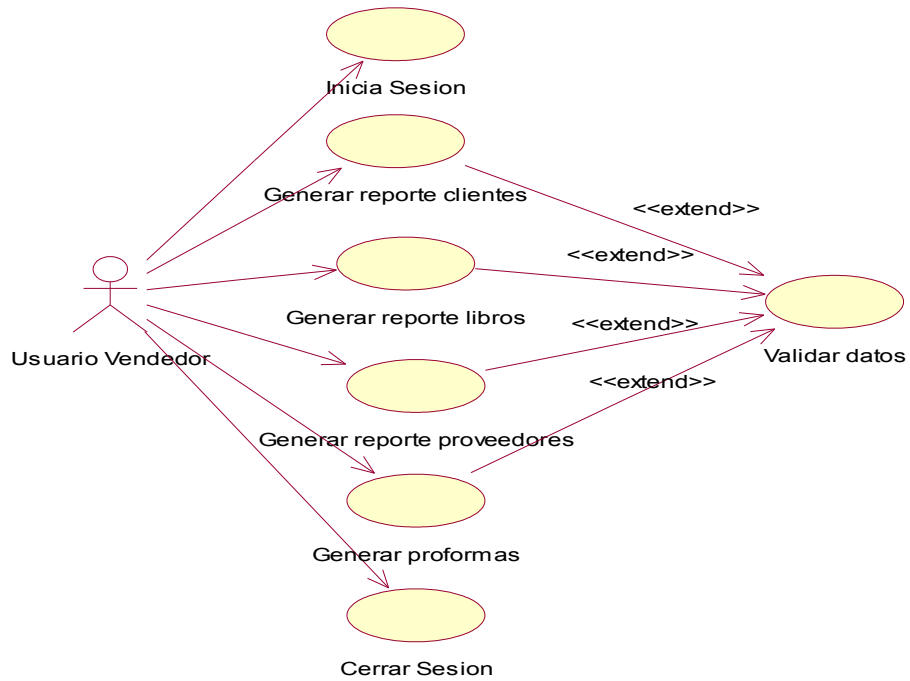
Figura V.3. Diagrama de Caso de Uso Usuario Vendedor

Elaborado por: Ángel Apina, Richard Robles

✓ Diagrama de Caso de Uso Usuario Vendedor Generar Facturas



✓ Diagrama de Caso de Uso Usuario Vendedor Generar Reportes



Casos de Uso

Caso de Uso: Iniciar Sesión

Identificador Caso de Uso	CU-1
Nombre del Caso de Uso	Iniciar sesión
Actores	Usuario Administrador Usuario Vendedor
Propósito	Ingresar sistema con el menú correspondiente según el tipo de usuario.
Visión General	El usuario ingresa al sistema, el sistema le solicita el nombre de usuario, password, sucursal y el tipo de usuario, el usuario ingresa los datos, el sistema verifica los datos e ingresa a la ventana principal en la que se muestra el menú principal que le corresponde al usuario con el cual ingreso.
Tipo	Primario
Referencias	
Curso típico de eventos	
Acción del Actor	Respuesta del Sistema
1. Ingresar a la aplicación	2. Presenta el menú principal
	3. Solicita al usuario el Nombre de Usuario, la contraseña, tipo de usuario y sucursal a la que pertenece
4. Ingresar el nombre de Usuario, contraseña, tipo de usuario y sucursal.	5. Ingresar a la pantalla principal con el menú que le corresponde al usuario con el cual ingreso.
Cursos Alternativos	
Línea 5: Los datos no son correctos, da un mensaje de error y borra los datos ingresados	

Caso de Uso: Cerrar Sesión

Identificador Caso de Uso	CU-2
Nombre del Caso de Uso	Cerrar sesión

Actores	Usuario Administrador Usuario Vendedor
Propósito	Cerrar la sesión antes de cerrar el sistema.
Visión General	El usuario, una vez que haya iniciado una sesión es necesario cerrar
Tipo	Primario
Referencias	
Curso típico de eventos	
Acción del Actor	Respuesta del Sistema
1. Solicita cerrar la sesión	2. Cierra la sesión

Caso de Uso: Crear Usuario

Identificador Caso de Uso	CU-3
Nombre del Caso de Uso	Crear usuario
Actores	Usuario Administrador
Propósito	Crear un usuario del tipo solicitado
Visión General	El usuario ADMINISTRADOR solicita al sistema crear un usuario, el sistema solicita los datos del usuario que se va a crear, el usuario ingresa los datos, el sistema valida y verifica los datos, y luego registra al nuevo usuario.
Tipo	Primario
Referencias	
Curso típico de eventos	
Acción del Actor	Respuesta del Sistema
1. Solicita crear un nuevo usuario	2. Solicita ingresar los datos del nuevo usuario.
3. Ingresar los datos del usuario nuevo	4. Valida y verifica los datos, y registra al nuevo usuario en la BD.

Cursos Alternativos

Línea 4: Si los datos son incorrectos muestra un mensaje y no continúa.

Caso de Uso: Buscar Usuario

Identificador Caso de Uso	CU-4
Nombre del Caso de Uso	Buscar usuario
Actores	Usuario Administrador
Propósito	Buscar información del usuario
Visión General	El usuario ADMINISTRADOR solicita la información del usuario identificado, el sistema muestra la información del usuario solicitado.
Tipo	Primario
Referencias	
Curso típico de eventos	
Acción del Actor	Respuesta del Sistema
1. Solicita información usuario dado, según un identificador en especial (ID, Nombre, etc).	2. Muestra la información del usuario solicitado
Cursos Alternativos	
Línea 2: El sistema muestra la tabla de usuarios vacía si no existe el usuario buscado por su identificador.	

Caso de Uso: Modificar Usuario

Identificador Caso de Uso	CU-5
Nombre del Caso de Uso	Modificar usuario
Actores	Usuario Administrador
Propósito	Actualiza los datos del usuario

Visión General	El usuario ADMINISTRADOR solicita actualizar los datos de un usuario en particular, el sistema devuelve los datos del usuario, el usuario ADMINISTRADOR actualiza los datos permitidos, el sistema valida y verifica los datos, para actualizar en la BD.	
Tipo	Primario	
Referencias	Buscar Elemento Usuario	
Curso típico de eventos		
Acción del Actor	Respuesta del Sistema	
1. Solicita actualizar los datos de un usuario dado.	2. Presenta los datos actuales del usuario.	
3. Modifica los datos permitidos.	4. Valida y verifica los datos, y luego los registra en la BD.	
Cursos Alternativos		
Línea 4: Si los datos son incorrectos muestra un mensaje de error y no continúa.		

Caso de Uso: Eliminar Usuario

Identificador Caso de Uso	CU-6
Nombre del Caso de Uso	Eliminar usuario
Actores	Usuario Administrador
Propósito	Elimina un usuario
Visión General	El usuario ADMINISTRADOR solicita eliminar un usuario del sistema, el sistema muestra un mensaje de confirmación, luego de la confirmación por parte del usuario Administrados el sistema

	eliminar el usuario, y actualiza en la BD.	
Tipo	Primario	
Referencias		
Curso típico de eventos		
Acción del Actor	Respuesta del Sistema	
1. Solicita eliminar usuario dado.	2. Muestra un mensaje de confirmación	
3. Confirma la solicitud.	4. Elimina el usuario y refresca la vista de todos los usuarios que no fueron eliminados.	

Caso de Uso: Generar Reportes

Identificador Caso de Uso	CU-7	
Nombre del Caso de Uso	Generar Reportes	
Actores	Usuario Administrador, Usuario vendedor	
Propósito	Ingresa a Reportes	
Visión General	El usuario ADMINISTRADOR o vendedor ingresa al menú de reportes y elije el reporte que desea generar.	
Tipo	Primario	
Referencias		
Curso típico de eventos		
Acción del Actor	Respuesta del Sistema	
1. Solicita ingresar a reportes.	2. Muestra todas las reportes del sistema	
3. Selecciona un reporte en particular	4. Muestra la información del reporte	

Caso de Uso: Crear Cliente, proveedor, libro

Identificador Caso de Uso	CU-8	
Nombre del Caso de Uso	Crear cliente, proveedor, libro	

Actores	Usuario vendedor
Propósito	Crear un nuevo cliente, proveedor, libro
Visión General	El usuario vendedor solicita crear un nuevo cliente, proveedor, libro el sistema pide que se ingrese los datos del nuevo cliente y los registra en la base de datos
Tipo	Primario
Referencias	
Curso típico de eventos	
Acción del Actor	Respuesta del Sistema
1. Solicita crear un cliente, proveedor, libro	2. Muestra ventana de creación del nuevo cliente, proveedor, libro.
3. El usuario ingresa los datos del nuevo cliente, proveedor, libro	4. Registra los datos en la base de datos.
	5. Muestra un mensaje de confirmación.
Cursos Alternativos	
Línea 4: Si los datos están mal ingresados se presenta un mensaje de error.	

Caso de Uso: Registrar ventas directas o ventas a consignación

Identificador Caso de Uso	CU-9
Nombre del Caso de Uso	Registrar Ventas
Actores	Usuario vendedor
Propósito	Registrar ventas
Visión General	El usuario vendedor solicita registrar una venta realizada.
Tipo	Primario
Referencias	
Curso típico de eventos	
Acción del Actor	Respuesta del Sistema
1. Solicita registrar una venta.	2. Muestra ventana de registro de ventas.
3. El usuario elije el tipo de venta que desea	4. Muestra la ventana para elegir los libros que

registrar (directa o a consignación).	desea vender.
5. El usuario elije los libros y registra la venta.	6. Muestra un mensaje de confirmación.
Cursos Alternativos	
Línea 5: Si no hay suficientes libros en la sucursal se muestra un mensaje de error.	

Caso de Uso: Modificar un cliente, libro, proveedor

Identificador Caso de Uso	CU-10
Nombre del Caso de Uso	Modificar cliente, libro, proveedor
Actores	Usuario vendedor
Propósito	Modificar un cliente, libro, proveedor.
Visión General	El usuario solicita la modificación de un cliente, libro, proveedor.
Tipo	Primario
Referencias	
Curso típico de eventos	
Acción del Actor	Respuesta del Sistema
1. Solicita modificar un cliente, libro, proveedor.	2. Solicita modificar los datos.
3. Modifica los datos	4. Valida, modifica los datos y registra elemento en la BD.
Cursos Alternativos	
Línea 4: Si los datos son incorrectos muestra un mensaje y no continúa.	
Línea 4: Si los datos no cumplen con las restricciones impuestas, el sistema muestra un mensaje y no continúa.	

Caso de Uso: Registrar una devolución

Identificador Caso de Uso	CU-11
Nombre del Caso de Uso	Registrar una devolución
Actores	Usuario vendedor

Propósito	Registrar una devolución	
Visión General	El usuario vendedor solicita registrar una devolución.	
Tipo	Primario	
Referencias	Buscar libros	
Curso típico de eventos		
Acción del Actor	Respuesta del Sistema	
1. Solicita registrar una devolución.	2. Presenta los datos de la compra que se ha realizado.	
3. Elige los libros a devolver.	4. Valida la devolución y registra en la base de datos.	
Cursos Alternativos		
Línea 4.: Si los datos son incorrectos muestra un mensaje de error.		

Glosario de Términos y Abreviaturas

Definiciones

- ✓ **Gerente General de Contilibro.**- Persona que se encarga de la parte organizacional de la organización.
- ✓ **Usuario.**- Persona (ente) a la cual se le brinda un servicio.
- ✓ **Casos de Uso.**- Representación del flujo de información mediante los actores y los casos de uso correspondientes.
- ✓ **MSF.**- Microsoft Solution Framework, metodología ágil empleada para el desarrollo de la aplicación.
- ✓ **Servidor de Base de Datos.**- El servidor de la base de datos es un programa de computadora que proporciona servicios de la base de datos a otros programas o computadoras, puede también referirse a una computadora dedicada a funcionar tal programa.
- ✓ **Base de Datos.**- Cualquier conjunto de datos organizados para su almacenamiento en la memoria de un ordenador o computadora, diseñado para facilitar su mantenimiento y acceso de una forma estándar.
- ✓ **Interfaz.**- Punto en el que se establece una conexión entre dos elementos, que les permite trabajar juntos. La interfaz es el medio que permite la interacción entre esos elementos.

- ✓ **Sistema Operativo.**- Software básico de una computadora. Tiene tres grandes funciones: coordina y manipula el hardware del ordenador o computadora, organiza los archivos en diversos dispositivos de almacenamiento y gestiona los errores de hardware y la pérdida de datos.

Acrónimos

- ✓ **HW.**- Hardware.
- ✓ **SW.**- Software.

Diseño Lógico

Tecnologías a Utilizar en el Proyecto

Herramientas

De acuerdo a la naturaleza de los requerimientos y la solución planteada para estos, se ha elegido a la tecnología Microsoft Visual Studio .NET para la implementación del proyecto. Además como metodología de desarrollo se ha elegido MSF, por cuanto es adaptable a cualquier tipo de proyecto, sea este grande, pequeño, y también se adapta al tiempo disponible para el desarrollo.

Aplicación Multicapas

Almacenes de Datos

SQL Server 2000 nos permitirá la implementación de los objetos necesarios para el soporte de la solución. Principalmente nos permite la elaboración de procedimientos almacenados que brindan mayor rapidez en la ejecución y facilidad de mantenimiento.

Capa de acceso a Datos

El acceso a datos se implementó utilizando ODBC, ya que nos permite la abstracción de la mayoría de transacciones que representan la ejecución de sentencias y la devolución de conjuntos de datos para el procesamiento interno de la aplicación.

Capa de lógica de Negocio

La lógica de negocio se implementará utilizando clases de .net elaborando componentes que permitan realizar las tareas funcionales de la empresa.

Capa de Presentación o Interfaz de Usuario

La aplicación se implementará utilizando c# de .NET.

Diagrama de Secuencia

A continuación se presenta un diagrama de secuencia por cada tipo de usuario definido en el sistema.



Figura V.6. Diagrama de Secuencia Usuario de Administrador

Elaborado por: Ángel Apina, Richard Robles

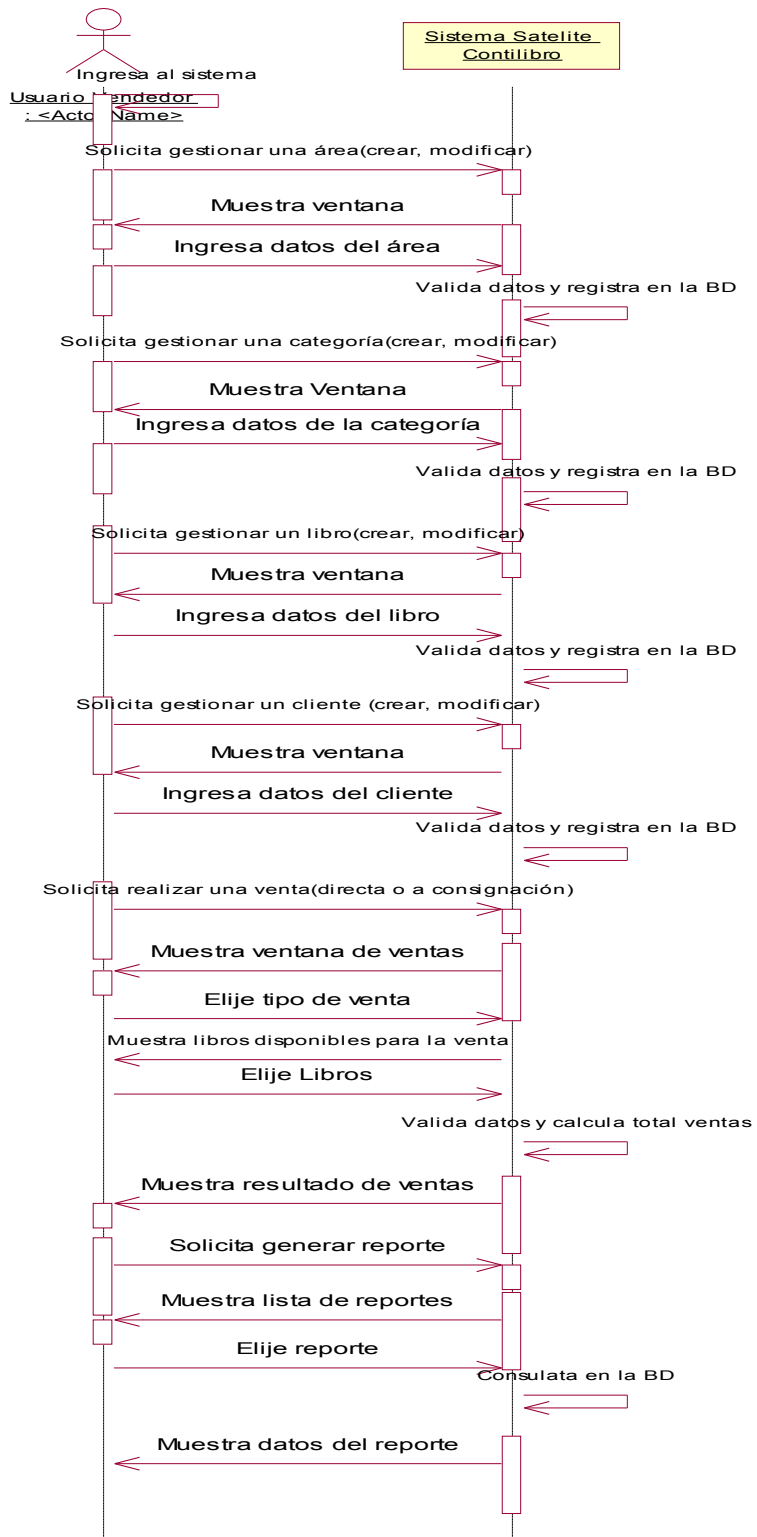


Diagrama de Clases

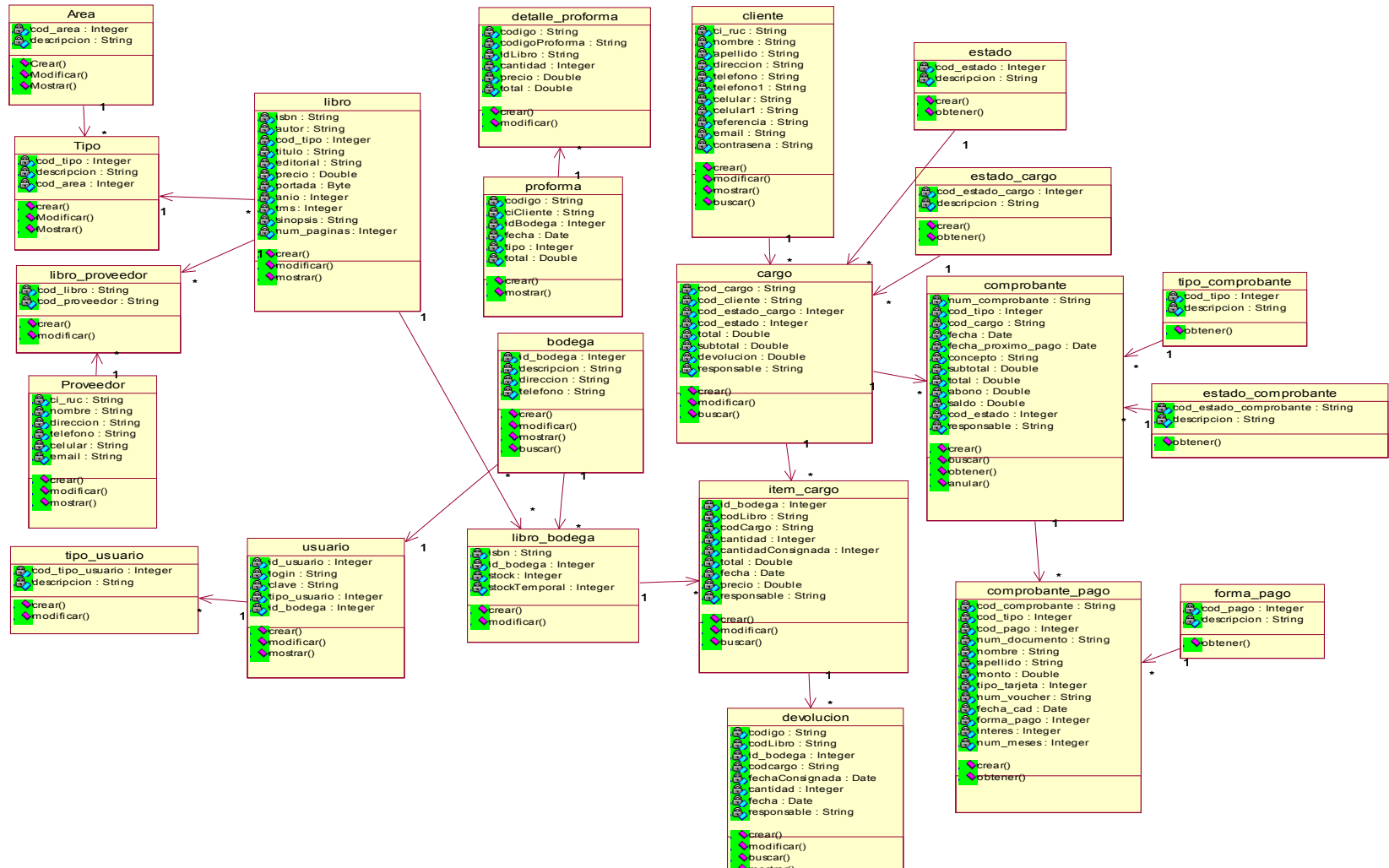


Diagrama de Componentes

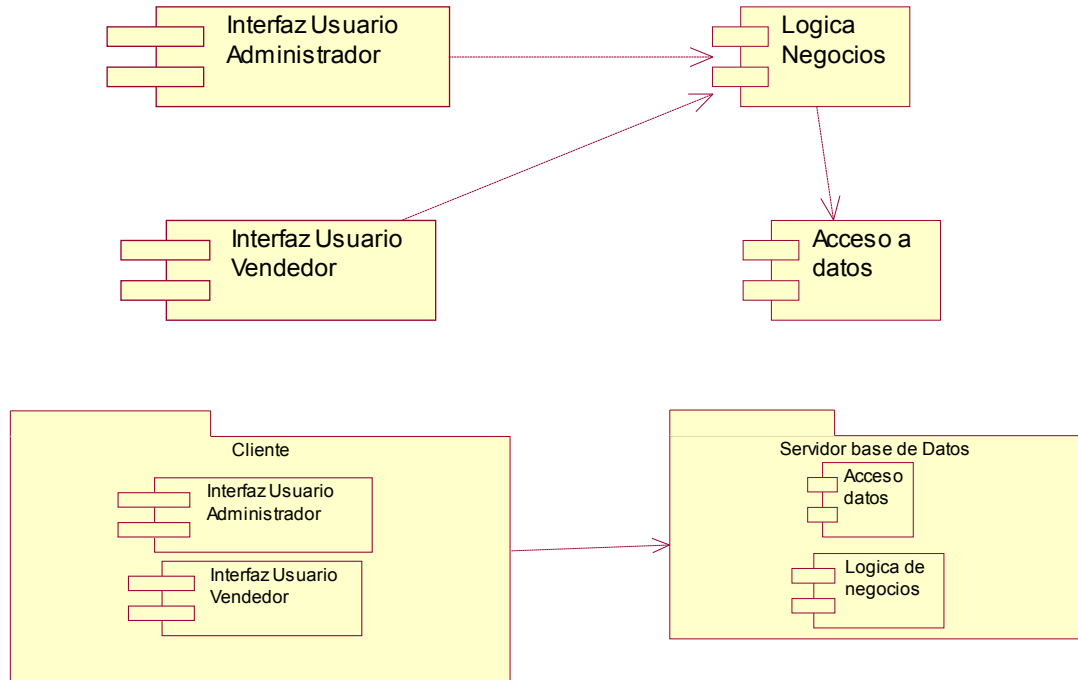
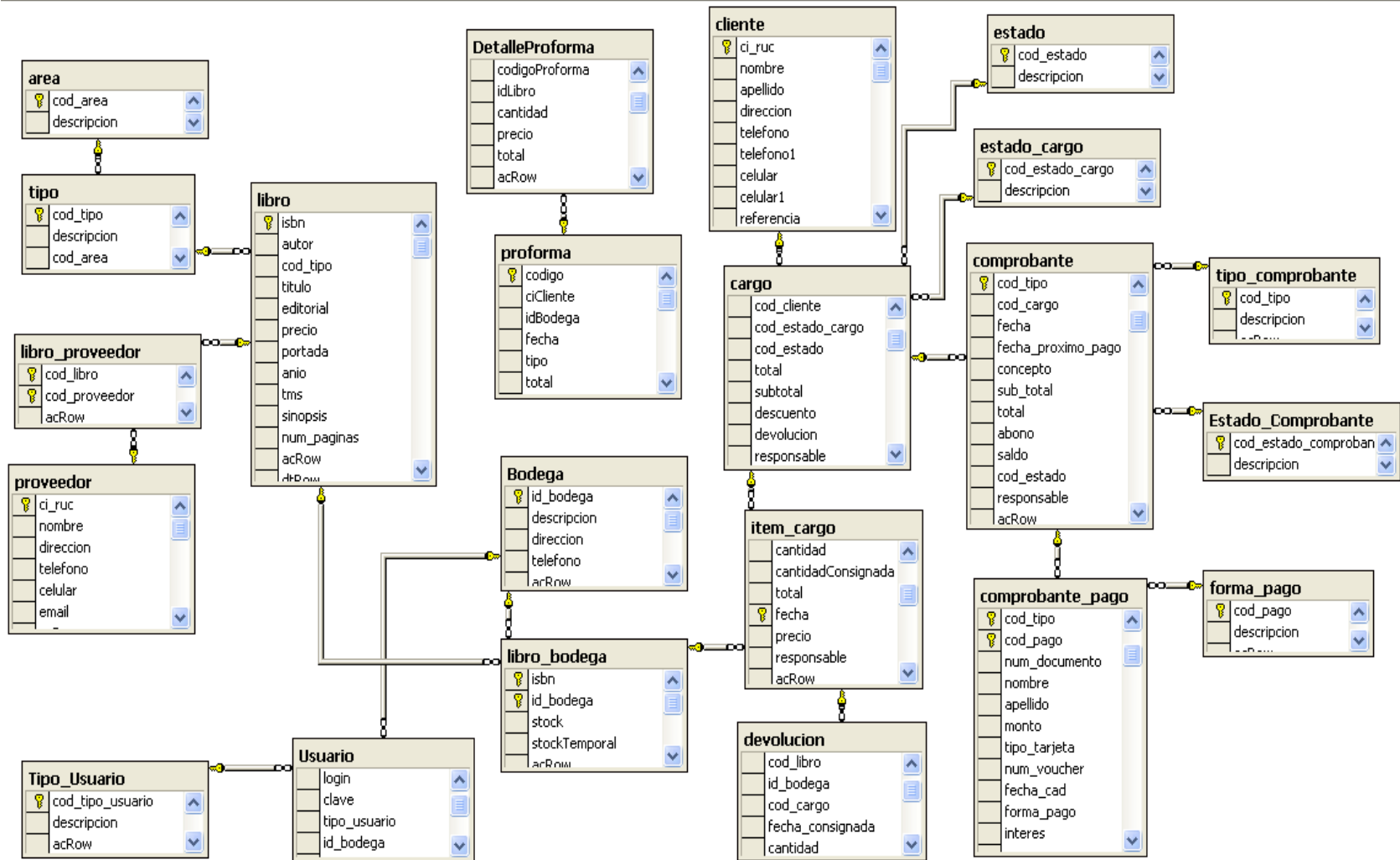


Diagrama Físico de la Base de Datos



Desarrollo

Nomenclatura y Estándares para el Desarrollo

Controles	AMBITO	
	Local o a nivel de procedimiento	Nivel de modulo o Form
Classes	Cls< NombreClase>	cls< NombreClase>
Formularios Win	frm< NombreFormulario>	IFrm< NombreFormulario>
Combobox	cmb< NombreCombo>	ICmb< NombreCombo>
Datagrid	dgv<NombreGrid>	IGrd<NombreGrid>
Buttons	Btn< NombreBoton>	IBtn< NombreBoton>
CheckBoxes	Chk< NombreCheck>	IChk< NombreCheck>
Textboxes	Txt< NombreText>	ITxt< NombreText>
Labels	Lb< NombreEtiqueta>	ILb< NombreEtiqueta>
Panel	Pnl< NombreEtiqueta>	IPnl< NombreEtiqueta>
LinkButton	Lbtn< NombreBoton>	IIBtn< NombreBoton>
Tipos primitivos		
Integer	int< Nombre >	IInt< Nombre >
Long	lng< Nombre >	ILng< Nombre >
Boolean	bln< Nombre >	IBln< Nombre >
String	str< Nombre >	IStr< Nombre >
Double	dbl< Nombre >	IDbl< Nombre >
Constantes	C_ < Nombre>	LC_ < Nombre >

Capa de Presentación

Diseño de Interfaces de Usuario

Las pantallas del sistema satélite de Contilibro son muy parecidas ya que todas cuentan con la misma lógica de ingreso, modificación y eliminación, de igual manera las consultas y reportes.

Por lo que a continuación de muestran las capturas de pantallas de las funciones del sistema más representativas para cada acción descrita.

Inicio de Sesión

Ingrese Usuario y Contraseña

Usuario:

Contraseña:

Tipo Usuario:

Sucursal:

MENU PRINCIPAL

Reportes Sucursales Usuarios Salir

Facturación

GENERAL

- Agrega Nuevo Cliente
- Editar Información del Cliente
- Agrega Nueva Área
- Modificar Área
- Agrega Nueva Categoría
- Modificar Categoría
- Elabora Proforma
- Administra Proformas

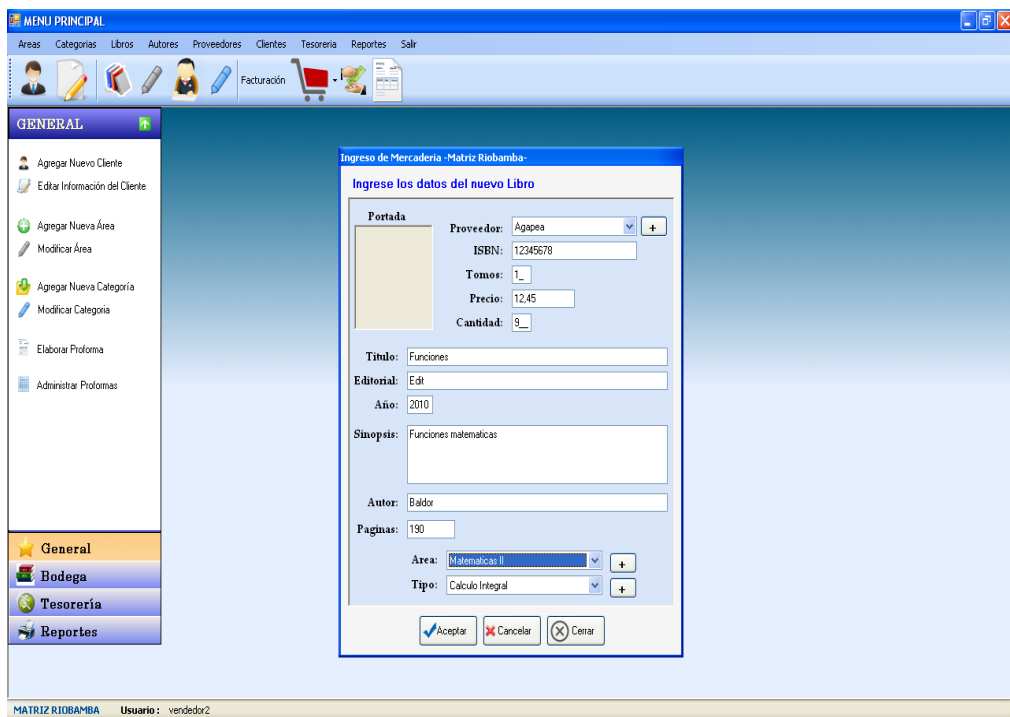
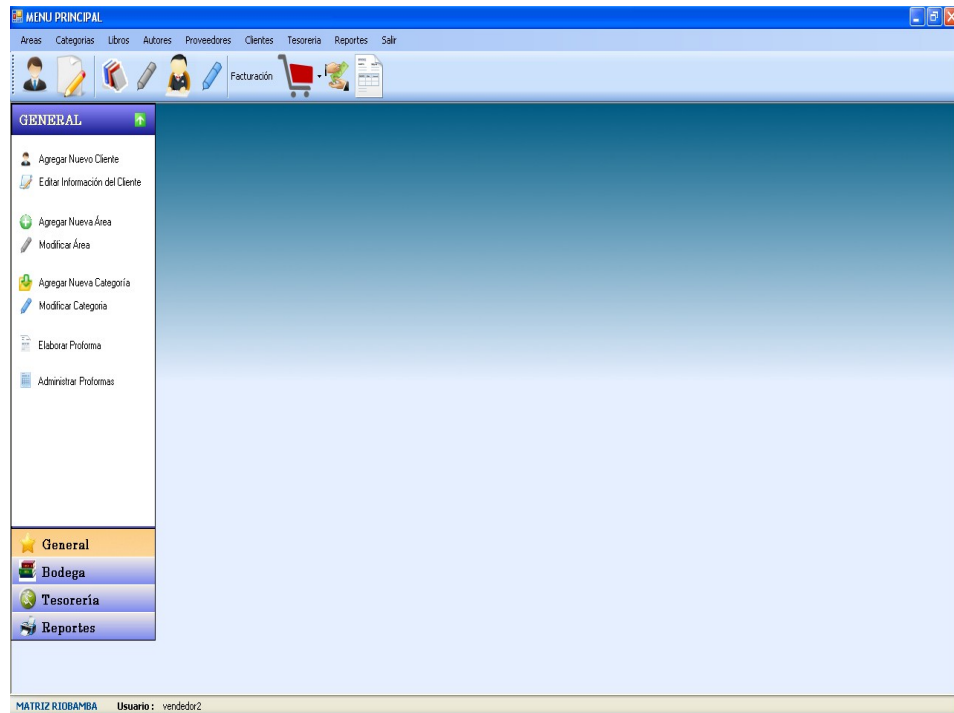
General

Bodega

Tesorería

Reportes

MATRIZ RIOBAMBA Usuario: administrador



MENU PRINCIPAL

Areas Categorías Libros Autores Proveedores Clientes Tesorería Reportes Salir

Facturación

GENERAL

- Agregar Nuevo Cliente
- Editar Información del Cliente
- Agregar Nueva Área
- Modificar Área
- Agregar Nueva Categoría
- Modificar Categoría
- Elaborar Proforma
- Administrar Proformas

General

Bodega

Tesorería

Reportes

MATRIZ RIOBAMBA Usuario: vendedor2

Ingreso Cliente

Ingrese los Datos del Nuevo Cliente

CI-RUC: 06032933-9 Contraseña: *****

Nombre: Ana

Apellido: Alban

Dirección: Estacion

Teléfono: 0329877987 Teléfono de Referencia: 0329877987

Celular: 098098098 Celular de Referencia: 098098098

Nombre de la Referencia: Teresa Gomez

Email: aslabr@hotmail.com

MENU PRINCIPAL

Areas Categorías Libros Autores Proveedores Clientes Tesorería Reportes Salir

Facturación

GENERAL

- Agregar Nuevo Cliente
- Editar Información del Cliente
- Agregar Nueva Área
- Modificar Área
- Agregar Nueva Categoría
- Modificar Categoría
- Elaborar Proforma
- Administrar Proformas

General

Bodega

Tesorería

Reportes

MATRIZ RIOBAMBA Usuario: vendedor2

Facturación Libros Consignados

Facturas Forma Pago

Cabecera

Tipo de venta: Contado Credito Cédula/RUC: 0400493169001

Número de Factura: Cliente: Giovanni Rodriguez

Fecha de Emisión: martes, 02 de agosto de 2011 Teléfono: 0329877989

Fecha de Vencimiento: martes, 02 de agosto de 2011 Dirección: La Doloresa

Detalle

ISBN	Detalle / Obra	Precio	Cantidad	Total
121212	paginas dinamicas	23.89	1	23.89
12345	ASP	24.50	2	49.00
1234567891234	Finanzas Internacionales	42.00	1	42.00

Sub Total	114.89
IVA	0.00
Descuento	---
TOTAL	114.89

MENU PRINCIPAL

Áreas Categorías Libros Autores Proveedores Clientes Tesorería Reportes Salir

Facturación

GENERAL

- Agregar Nuevo Cliente
- Editar Información del Cliente
- Agregar Nueva Área
- Modificar Área
- Agregar Nueva Categoría
- Modificar Categoría
- Elaborar Proforma
- Administrar Proformas

General Bodega Tesorería Reportes

Facturación

Factura | Forma de Pago

Cabecera

Tipo de venta: Contado Crédito Cédula RUC: 060333521-7

Número de Factura: 12345 Cliente: Angel Eduardo Apina Prieto

Fecha de Emisión: martes , 02 de agosto de 2011 Teléfono: 2948695

Fecha de Vencimiento: martes , 02 de agosto de 2011 Dirección: Orocco 15-39 y Morona

Detalle

CODIGO	TITULO	CANTIDAD	PRECIO	TOTAL
001	libro1	1	32,2	32,20
				Sub Total 32,20
				IVA 0,00
				Descuento ---
				TOTAL 32,20

Aceptar Cancelar

MATRIZ RIOBAMBA Usuario: vendedor2

MENU PRINCIPAL

Áreas Categorías Libros Autores Proveedores Clientes Tesorería Reportes Salir

Facturación

GENERAL

- Agregar Nuevo Cliente
- Editar Información del Cliente
- Agregar Nueva Área
- Modificar Área
- Agregar Nueva Categoría
- Modificar Categoría
- Elaborar Proforma
- Administrar Proformas

General Bodega Tesorería Reportes

Devolución de Libros (Bodega: 1 "Matriz Riobamba")

Cabecera

Fecha de Devolución: martes , 02 de agosto de 2011 Teléfono: 032989989

Cédula RUC: 060332933-4 Dirección: la estación

Cliente: Juan Lopez

Detalle

Editar Deshacer Todo

Fecha de Consignación	Id. Cargo	ISBN	Detalle / Obra	Precio	Total	Cantidad Consignada	Cantidad a Devolver	Bodega
19/05/2011	cat-5	121212	paginas dinamicas	23,89	23,89	1	0	1

Aceptar Cancelar

MATRIZ RIOBAMBA Usuario: vendedor2

MENU PRINCIPAL

Áreas Categorías Libros Autores Proveedores Clientes Tesorería Reportes Salir

Facturación

GENERAL

- Agregar Nuevo Cliente
- Editar Información del Cliente
- Agregar Nueva Área
- Modificar Área
- Agregar Nueva Categoría
- Modificar Categoría
- Elaborar Profoma
- Administrar Profomas

General Bodega Tesorería Reportes

Consignación

Cabeceza

Fecha de Consignación: martes ,02 de agosto de 2011 Teléfono: 2948595
 Cédula RUC: 060339521-7 Dirección: Orozco 15-39 y Morona
 Cliente: Angel Eduardo Apina Reino

Detalle

CANT.	CODIGO	DETALLE/OBRAS	TMS	PRECIO	TOTAL	
1	001	Titulo1	2	32,20	32,20	
1	12345	ASP	1	24,50	24,50	
					Sub Total	56,70

MATRIZ RIOBAMBA Usuario : vendedor2 (contlibro (Running) - Microsoft Visual Studio)

MENU PRINCIPAL

Áreas Categorías Libros Autores Proveedores Clientes Tesorería Reportes Salir

Facturación

GENERAL

- Agregar Nuevo Cliente
- Editar Información del Cliente
- Agregar Nueva Área
- Modificar Área
- Agregar Nueva Categoría
- Modificar Categoría
- Elaborar Profoma
- Administrar Profomas

General Bodega Tesorería Reportes

Añadir Recibo

Factura | Forma de Pago

Datos del Comprobante de Pago

Num. Comprobante: 12 Cédula RUC: 06044566-9
 Fecha de Emisión: martes ,02 de agosto de 2011 Cliente: Patricia Machado
 Fecha Proximo Pago: martes ,02 de agosto de 2011 Teléfono: 032628006
 Número de Factura: 123954858 Dirección: La Estacion
 Por Concepto de: pago

Detalle de Libros Facturados

ISBN	Detalle / Obra	Cantidad	Precio	Total
12345	ASP	1	24,50	24,50

Saldo Anterior: 22,50 Abono: 20,00 Saldo Deudor: 2,50

Aceptar Cancelar

MATRIZ RIOBAMBA Usuario : vendedor2

MENU PRINCIPAL

Áreas Categorías Libros Autores Proveedores Clientes Tesorería Reportes Salir

Facturación

GENERAL

Agregar Nuevo Cliente
 Editar Información del Cliente
 Agregar Nueva Área
 Modificar Área
 Agregar Nueva Categoría
 Modificar Categoría
 Elaborar Profoma
 Administrar Profomas

General
 Bodega
 Tesorería
 Reportes

Cedula: 060333521-7 Dirección: Orozco 15-39 y Morona Fecha: martes, 02 de agosto de 2011
 Nombres: Angel Eduardo Apina Reino
 Telefono: 2948595

Buscar: ISBN Área: Economía II Tipo: Blanca Autor: Título: Buscar

Seleccionar Todos:

ISBN	Título	Autor	Área	Categoría	Precio	Stock	Cantidad
001	Titulo1		Economía II	Blanca	32,2	4	1
002	Titulo2		Economía II	Blanca	41,2	6	1
003	tit		Economía II	Blanca	32,4	6	1
121212	paginas dinamicas		Economía II	Blanca	23,89	1	1

Lista de Libros Consignados

ISBN	Título	Precio	Cantidad	Total
001	Titulo1	32,2	1	32,2
002	Titulo2	41,2	1	41,2
003	tit	32,4	1	32,4
121212	paginas dinamicas	23,89	1	23,89

Agregar
 Limpiar Lista
 Eliminar de la Lista
 Imprimir Profoma
 Centar

TOTAL 129,69

MATRIZ RIOBAMBA Usuario: vendedor2

MENU PRINCIPAL

Áreas Categorías Libros Autores Proveedores Clientes Tesorería Reportes Salir

Facturación

GENERAL

Agregar Nuevo Cliente
 Editar Información del Cliente
 Agregar Nueva Área
 Modificar Área
 Agregar Nueva Categoría
 Modificar Categoría
 Elaborar Profoma
 Administrar Profomas

General
 Bodega
 Tesorería
 Reportes

Reporte Libros

Man Report

CONTILIBRO
REPORTE DE LIBROS

ISBN	AUTOR	AREA	CATEGORIA	TITULO	PRECIO	STOCK
001	autor1	Economía II	Blanca	Titulo1	32,20	4
002	autor2	Economía II	Blanca	Titulo2	41,20	6
003	autor	Economía II	Blanca	tit	32,40	6
005	sq	Informática	Base de Datos	TituloNuevo	5,00	3
007	Advanced	Informática	AES	encriptacion	10,25	6
1	angel	Economía II	Economía Doméstica	iiiiiii	12,00	15
111111	autor1	Informática	Multimedia	Diseño Web	34,00	-1
121212	autor2	Economía II	Blanca	paginas dinamicas	23,89	1

Current Page No.: 1 Total Page No.: 1 Zoom Factor: 100%

MATRIZ RIOBAMBA Usuario: vendedor2

MENU PRINCIPAL

Áreas Categorías Libros Autores Proveedores Clientes Tesorería Reportes Salir

Facturación

GENERAL

- Agregar Nuevo Cliente
- Editar Información del Cliente
- Agregar Nueva Área
- Modificar Área
- Agregar Nueva Categoría
- Modificar Categoría
- Elaborar Profomas
- Administrar Profomas

General
Bodega
Tesorería
Reportes

Reporte Clientes

Main Report

CONTILIBRO
REPORTE DE CLIENTES

CI-RUC	NOMBRE	APELLIDO	DIRECCION	CELULAR	EMAIL
06033321-7	Angel Eduardo	Xpina Reino	Orocco 15-39 y Worona	095280188	angel_xpina284@hotmail.com
06032255-8	dede	ded	dede	098098098	ded
06032293-4	Juan	lopez	la estacion	098890000	jlopez@hotmail.com
06044566-8	Patricia	Itachado	La Estacion	084009341	pmashado@gmail.com
14065076-9	Miriam Susana	Riera Idrovo	Brasil y Orocco	084009341	miriam_susana15@hotmail.com
06032293-5	Richard Fernando	Robles Carrillo	10 de Agosto 0915 y Av. Eloy Alfaro	061882884	rfric2@gmail.com
040485163001	Giovani	Rodriguez	La Dolorosa	098098098	grodri@guac@gmail.com

Current Page No.: 1 Total Page No.: 1 Zoom Factor: 100%

MATRIZ RIOBAMBA Usuario: vendedor2

MENU PRINCIPAL

Áreas Categorías Libros Autores Proveedores Clientes Tesorería Reportes Salir

Facturación

GENERAL

- Agregar Nuevo Cliente
- Editar Información del Cliente
- Agregar Nueva Área
- Modificar Área
- Agregar Nueva Categoría
- Modificar Categoría
- Elaborar Profomas
- Administrar Profomas

General
Bodega
Tesorería
Reportes

Reporte Proveedores

Main Report

CONTILIBRO
REPORTE DE PROVEEDORES

CI-RUC	NOMBRE PROV	DIRECCION	TELEFONO	CELULAR	EMAIL
060251254-9	Agapea	Calle 1 entre Calle 2 y Calle 3	302795236	095669478	ventaLibros@agapea.com
140896512-8	Proveedor1	Calle1 y calle2	28874563	085123854	proveedor1@hotmail.com

Current Page No.: 1 Total Page No.: 1 Zoom Factor: 100%

MATRIZ RIOBAMBA Usuario: vendedor2

Capa de Datos

Implementación de la Base de Datos

Identificar Clases y Objetos

CLASES Y OBJETO
Area
Categoría
Cliente
Usuario
Tipo_usuario
Bodega
Libro
Libro_bodega
Cargo
Ítem_cargo
Devolución
Comprobante
Comprobante_pago
forma_pago
Tipo_comprobante
Estado
Estado_comprobante
Proforma
Detalle_proforma
Estado_cargo
Libro_proveedor
Tipo_usuario

Identificar las Relaciones

Las relaciones que tienen los objetos de la tabla V.24, son las que se detalla a continuación.

- ✓ Un cliente puede generar uno o varios cargos.
- ✓ Un libro puede venir de varios proveedores.
- ✓ Un área puede tener varias categorías.
- ✓ Un libro puede tener varias categorías.
- ✓ Un cargo puede generar varios comprobantes.
- ✓ Un comprobante puede generar varios comprobantes de pago.
- ✓ Un comprobante puede tener varios estados.
- ✓ Un comprobante de pago puede cancelarse de varias formas.
- ✓ Un usuario puede ser de varios tipos.
- ✓ Un libro puede estar en varias bodegas.
- ✓ Una proforma puede generar varios detalles de proforma.
- ✓ Un cargo puede generar varios ítem cargo.
- ✓ Un ítem cargo puede generar varias devoluciones.

Script de la Base de Datos

La implementación de la base de datos consiste en la creación de tablas con sus atributos respectivos, que ayudan al funcionamiento del sistema.

Existe un script de generación de tabla, por cada una de las tablas de la base de datos del sistema, a continuación se presenta un script para la creación de una de las tablas de la base de datos del sistema, para la creación de las demás tablas solo varía la cantidad y el nombre de los atributos que posean cada tabla.

```

/***** Objeto: tabla [dbo].[item_cargo]   fecha de la secuencia de comandos:
02/08/2011 14:16:16 *****/

```

```

CREATE TABLE [dbo].[item_cargo] (
    [id_bodega] [intCodigo] NOT NULL ,

```

```

[cod_libro] [txtIsbn] NOT NULL ,
[cod_cargo] [txtCadenaPequena] NOT NULL ,
[cantidad] [intCantidad] NULL ,
[cantidadConsignada] [intCantidad] NULL ,
[total] [dcDinero1] NULL ,
[fecha] [dtFecha] NOT NULL ,
[precio] [dcDinero1] NULL ,
[responsable] [txtCadenaPequena] NULL ,
[acRow] [int] NULL ,
[dtRow] [datetime] NULL ,
[idRow] [varchar] (60) NULL
) ON [PRIMARY]
GO

```

Implementación de Acceso a Datos

El acceso a datos se lo realiza en la capa de acceso de la aplicación realizada en Microsoft Visual Studio 2005.

Se debe aclarar que para la conexión a la base de datos se utilizó la herramienta Enterprise Library, lo que me permite hacer procedimientos de manera general y no solo para una base de datos en específico.

Los procedimientos que aquí se muestran sirven para poder realizar cualquier tipo de tarea con la base de datos.

```

public void CrearComandoPA(String NombreStoredProcedure)
{
    this.comando = factory.CreateCommand();
    this.comando.Connection = this.conexion;
    this.comando.CommandType = CommandType.StoredProcedure;
    this.comando.CommandText = NombreStoredProcedure;
    if (this.transaccion != null)
    {
        this.comando.Transaction=this.transaccion;
    }
}

```

```

    }
}

/// <summary>
/// Obtiene la dirección del parámetro
/// </summary>
/// <param name="opcion">1: input, 2:output,
3:Input/Output</param>
/// <returns>dirección del parámetro</returns>
private ParameterDirection ObtenerDireccion(int opcion)
{
    ParameterDirection direccion;

    switch (opcion)
    {
        case 1:
        {
            direccion = ParameterDirection.Input;
            break;
        }
        case 2:
        {
            direccion = ParameterDirection.Output;
            break;
        }
        case 3:
        {
            direccion = ParameterDirection.InputOutput;
            break;
        }
        default:
        {
            throw new BaseDatosException("El parámetro
dirección no es correcto recuerde que debe estar entre 1 y 3");
            break;
        }
    }

    return direccion;
}

```

```

    }

    public void AsignarParametroNuloAlPA(string nombreparametro, int
    direccion)
    {
        DbParameter parametro = factory.CreateParameter();
        parametro.ParameterName = nombreparametro;
        parametro.DbType = DbType.AnsiString;
        parametro.Value = DBNull.Value;
        parametro.Direction = ObtenerDireccion(direccion);
        this.comando.Parameters.Add(parametro);
    }

    public DataSet EjecutarConsulta(String nombreDataSet, String
    nombreTabla)
    {
        DataSet ds = new DataSet(nombreDataSet);
        this.adapter = factory.CreateDataAdapter();
        try
        {
            this.adapter.SelectCommand = this.comando;
            this.adapter.Fill(ds, nombreTabla);
        }
        catch (Exception)
        {
            throw new BaseDatosException("Error al llenar el
conjunto de Datos");
        }

        return ds;
    }

```

La línea donde se realiza la conexión a la base de datos se lo realiza en el archivo App.config.

```
<connectionStrings>
```

```

    <add
name="WindowsAppContiLibro.Properties.Settings.contilibroBDConnecti
onString1"
    connectionString="Data Source=RFR28;Initial
Catalog=contilibroBD;User ID=sa;Password=sql1"
    providerName="System.Data.SqlClient" />
</connectionStrings>

```

Diseño de Esquemas

Capa de Negocios

En la capa de negocios tenemos las clases con las que se trabajará para manipular los datos de la base de datos, además también se cuenta con una clase catálogo en donde se llaman a los procedimientos almacenados de la base de datos.

Clase donde se identifican las propiedades de una Tabla, con sus respectivos métodos SET y GET.

```

public class Area
{
    int cod_area;
    String descripcion;

    /// <summary>
    /// Crea una instancia de área
    /// </summary>
    /// <param name="codArea"> código del área</param>
    /// <param name="nomArea">nombre del área</param>
    public Area(int codArea, String descArea)
    {
        cod_area = codArea;
        descripcion = descArea;
    }

    public Area()
    {
        cod_area = 0;
        descripcion = "";
    }
}

```



```
/// <summary>
/// Código del área
/// </summary>
public int Codigo
{
    get
    {
        return cod_area;
    }
    set
    {
        cod_area = value;
    }
}

/// <summary>
/// Nombre del área
/// </summary>
public String Descripcion
{
    get
    {
        return descripcion;
    }
    set
    {
        if (value.Trim() != "")
        {
            descripcion = value;
        }
        else
        {
            throw new Exception("Error, ingrese la
descripcion...");
        }
    }
}
```

Catálogo de la clase área.

```

public class CatalogoAreas
{
    /// <summary>
    /// inserta una nueva area en la BD
    /// </summary>
    /// <param name="miArea">area que se va a ingresar en la
BD</param>
    /// <returns></returns>
    public static Boolean NuevaArea(Area miArea)
    {
        BaseDatos bd = new BaseDatos();
        bd.Conectar();
        try
        {
            bd.CrearComandoPA("sp_insert_Area");
            bd.AsignarParametroEnteroAlPA("@codigo",
miArea.Codigo, 1);
            bd.AsignarParametroCadenaAlPA("@descripcion",
miArea.Descripcion, 1);
            bd.EjecutarComando();
            bd.Desconectar();
            return true;
        }
        catch (BaseDatosException ex)
        {
            throw new BaseDatosException("Error al acceder a la
base de datos", ex);
            return false;
        }
        catch (LogicaNegociosException ex)
        {
            throw new LogicaNegociosException("Error al obtener
los datos del Area", ex);
            return false;
        }
    }
}

```

```
        finally
        {
            bd.Desconectar();
        }
    }

    public static Boolean ModificarArea(Area objArea)
    {
        BaseDatos bd = new BaseDatos();
        bd.Conectar();
        try
        {
            bd.CrearComandoPA("sp_update_area");
            bd.AsignarParametroEnteroAlPA("@cod_area",
objArea.Codigo, 1);
            bd.AsignarParametroCadenaAlPA("@descripcion",
objArea.Descripcion, 1);
            bd.EjecutarComando();
            bd.Desconectar();
            return true;
        }
        catch (BaseDatosException ex)
        {
            throw new BaseDatosException("Error al acceder a la
base de datos", ex);
            return false;
        }
        catch (LogicaNegociosException ex)
        {
            throw new LogicaNegociosException("Error al
modificar los datos del Area", ex);
            return false;
        }
        finally
        {
            bd.Desconectar();
        }
    }
}
```

```

}

/// <summary>
/// obtiene el area segun su codigo
/// </summary>
/// <param name="codArea">codigo del area a buscar</param>
/// <returns></returns>
///
public static Area ObtenerArea(int codArea)
{
    Area miArea = new Area();
    BaseDatos bd = new BaseDatos();
    bd.Conectar();
    try
    {
        bd.CrearComandoPA("sp_obtener_area");
        bd.AsignarParametroEnteroAlPA("@codigo", codArea,
1);

        DbDataReader Datos = bd.EjecutarConsulta();
        while (Datos.Read())
        {
            try
            {
                miArea = new
Area(Convert.ToInt32(Datos.GetValue(0)), Convert.ToString(Datos.GetV
alue(1)));
            }
            catch (InvalidCastException ex)
            {
                throw new LogicaNegociosException("Los
tipos no coinciden.", ex);
            }
            catch (DbException ex)
            {
                throw new LogicaNegociosException("Error de
ADO.NET", ex);
            }
        }
    }
}

```

```

        Datos.Close();
        bd.Desconectar();

    }

    catch (BaseDatosException ex)
    {
        throw new BaseDatosException("Error al acceder a la
base de datos para obtener el Area.");
    }

    catch (LogicaNegociosException ex)
    {
        throw new LogicaNegociosException("Error al obtener
el Area." + ex.Message);
    }

    return miArea;
}

public static List<Area> ObtenerAreas()
{
    List<Area> areas = new List<Area>();
    try
    {
        BaseDatos bd = new BaseDatos();
        bd.Conectar();
        bd.CrearComandoPA("sp_obtener_todas_areas");
        DataSet datos = new DataSet();
        datos = bd.EjecutarConsulta("DS", "Areas");
        Area a;
        try
        {
            int numAreas = datos.Tables[0].Rows.Count;
            for (int i = 0; i < numAreas; i++)
            {
                a = new Area(
                    Convert.ToInt32(datos.Tables
s[0].Rows[i].ItemArray[0]),
                    Convert.ToString(datos.Tabl
es[0].Rows[i].ItemArray[1]));
            }
        }
    }
}

```

```

        areas.Add(a);
    }
}
catch (InvalidCastException ex)
{
    throw new LogicaNegociosException("Los tipos no
coinciden", ex);
}
catch (DataException ex)
{
    throw new LogicaNegociosException("Error de
ADO.NET", ex);
}

    bd.Desconectar();
}
catch (BaseDatosException ex)
{
    throw new BaseDatosException("Error al acceder a la
base de datos", ex);
}
catch (LogicaNegociosException ex)
{
    throw new LogicaNegociosException("Error al obtener
los datos de las áreas", ex);
}
return areas;
}

```

Capa de Aplicaciones

Para esta capa se presentara partes de código de manera resumida ya que la mayoría del código es reutilizado.

Autenticación de Usuarios

```

private void btnAceptar_Click(object sender, EventArgs e)
{

```

```

        DataSet ds =
CatalogoUsuario.ObtenerUsuario(txtLogin.Text, txtClave.Text,
Convert.ToInt32(cmbUsuario.SelectedValue),
Convert.ToInt32(cmbBodega.SelectedValue));

        if (ds.Tables[0].Rows.Count > 0)
        {
            frmMenu.ActiveForm.Hide();

            objBodega =
CatalogoBodega.ObtenerObjetoBodega_Codigo(Convert.ToInt32(cmbBodega
.SelectedValue));

            frmMenu obj = new
frmMenu(CatalogoUsuario.ObtenerUsuario1(txtLogin.Text,
txtClave.Text, Convert.ToInt32(cmbUsuario.SelectedValue),
Convert.ToInt32(cmbBodega.SelectedValue),
Convert.ToInt32(cmbBodega.SelectedValue));

            obj.Show();
        }
        else
        {
            MessageBox.Show("No se encontro el
usuario...", "Contilibro", MessageBoxButtons.OK, MessageBoxIcon.Inform
ation);

            txtClave.Clear();
            txtLogin.Clear();
            txtLogin.Focus();
        }
    }
}

```

Ingreso de Datos

```

private void btnAceptar_Click(object sender, EventArgs e)
{
    try
    {
        Area miArea = new Area();
        if (txtCodigo.Text.Trim() != "")
        {
            miArea.Codigo =
Convert.ToInt32(txtCodigo.Text);
        }
        else
    }
}

```

```

        {
            MessageBox.Show("El codigo del area no puede
estar vacio....");
            txtCodigo.Focus();
        }
        if (txtDescripcion.Text.Trim() != "")
        {
            miArea.Descripcion = txtDescripcion.Text;
        }
        else
        {
            MessageBox.Show("La descripcion del area no
puede estar vacia....");
            txtDescripcion.Focus();
        }
        if (txtDescripcion.Text != "" && txtCodigo.Text !=
"")
        {
            if (CatalogoAreas.NuevaArea(miArea))
            {
                MessageBox.Show("Area almacenada
correctamente!!!", "Contilibro");
                limpiar();
            }
            else
            {
                MessageBox.Show("Area no
ingresada...", "Contilibro", MessageBoxButtons.OK, MessageBoxIcon.Erro
r);
            }
        }
    }

catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}

```



```
}
```

Modificar Datos

Hay que mencionar que tanto el eliminar como el actualizar se lo realiza dentro de la propiedad RowCommand de un GridView, lo que permite:

```
private void btnModificar_Click(object sender, EventArgs e)
{
    try
    {
        Area objArea = new Area();
        objArea.Codigo = Convert.ToInt32(txtCodigo.Text);
        if (txtDescripcion.Text.Trim() != "")
        {
            objArea.Descripcion = txtDescripcion.Text;
        }
        else
        {
            MessageBox.Show("La descripcion del area no
puede estar vacia....");
            txtDescripcion.Focus();
        }
        if (txtCodigo.Text != "" && txtDescripcion.Text !=
"")
        {
            if (CatalogoAreas.ModificarArea(objArea))
            {
                MessageBox.Show("Datos Modificados
correctamente...", "Contilibro", MessageBoxButtons.OK,
MessageBoxIcon.Information);
                limpiar();
            }
            else
            {
                MessageBox.Show("Error al modificar los
datos...", "Contilibro", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }
}
```

```

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

Mostrar Datos extraídos desde la Base de Datos

```

try
    {
        Area objArea = new Area();
        if (txtCodBuscar.Text != "")
        {
            objArea =
CatalogoAreas.ObtenerArea(Convert.ToInt32(txtCodBuscar.Text));
            if (objArea.Codigo.Equals(0))
            {
                MessageBox.Show("No se encontró...",
"Contilibro", MessageBoxButtons.OK, MessageBoxIcon.Error);
                limpiar();
            }
            else
            {
                List<Area> area = new List<Area>();
                area.Add(objArea);
                dgvAreas.DataSource = area;
                txtCodigo.Text =
Convert.ToString(dgvAreas.CurrentRow.Cells[0].Value);
                txtDescripcion.Text =
Convert.ToString(dgvAreas.CurrentRow.Cells[1].Value);
            }
        }
        else
        {
            MessageBox.Show("Ingrese un valor... ",
"Contilibro", MessageBoxButtons.OK, MessageBoxIcon.Error);

```

```
    }  
  }  
  catch (Exception ex)  
  {  
      MessageBox.Show(ex.Message);  
  }
```

Especificaciones de Seguridad

Diseño de Estrategias de Autorización, Autenticación

Para el control de acceso a las diferentes funcionalidades del proyecto se ha implementado la autenticación de usuarios, es decir, para utilizar el sistema primeramente se debe autenticar con una cuanta y una clave válidas.

El diseño de la solución propuesta siempre mantendrá presente el perfil del usuario que se encuentra utilizando el sistema, cuidando que se le presenten únicamente las opciones necesarias y suficientes para el cumplimiento de sus actividades, cuidando de esta forma la integridad de la información que se almacene y por ende la integridad del sistema.

Estabilización

Las fases de Estabilización, se refieren a la puesta en marcha del sistema, lo que tiene que ver con el código fuente del sistema y base de datos todo con sus respectivos manuales.

Material de Capacitación

El sistema propuesto contempla distintos perfiles de usuario, con tareas y funciones claramente definidas, para las cuales siempre es necesario el respectivo material de capacitación. Se han propuesto los siguientes manuales, de acuerdo al ámbito operacional de cada actor del sistema:

Manual de Usuario.

Contiene la descripción de las funcionalidades generales del sistema, un manual de usuario debería poder ser entendido por cualquier usuario principiante, así como también serles útil a usuarios avanzados.

Manual de Instalación

El cual especifica los requerimientos y la configuración necesaria para la instalación del sistema, empezando por los recursos hardware, la plataforma operativa indicada para el despliegue, la configuración del entorno software necesario y el levantamiento de los servicios necesarios para el consumo desde los clientes definidos.

Instalación

Procedimientos y Procesos

El manual de procedimientos y procesos que posee la empresa debe ser actualizado tomando en cuenta a la nueva solución informática incorporada, para que de esta forma el personal operativo tenga una base firme de consulta sobre el nuevo curso de actividades a desarrollarse en la empresa.

Historial del Soporte Técnico

Todo aspecto importante efectuado durante ésta fase debe ser debidamente registrada, ayudando a manejar de forma centralizada las actividades que se han ido cumpliendo en base al tiempo previsto para la presente etapa.

Especificaciones de Seguridad

Diseño de Estrategias de Autorización, Autenticación

El diseño de la solución propuesta siempre mantendrá presente el perfil del usuario que se encuentra utilizando el sistema, cuidando que se le presenten únicamente las opciones necesarias y suficientes para el cumplimiento de sus actividades, cuidando de esta forma la integridad de la información que se almacene y por ende la integridad del sistema.

Estabilización

Las fases de Estabilización, se refieren a la puesta en marcha del sistema, lo que tiene que ver con el código fuente del sistema y base de datos todo con sus respectivos manuales.

Material de Capacitación

El sistema propuesto contempla distintos perfiles de usuario, con tareas y funciones claramente definidas, para las cuales siempre es necesario el respectivo material de capacitación. Se han propuesto los siguientes manuales, de acuerdo al ámbito operacional de cada actor del sistema:

Manual de Usuario.

Contiene la descripción de las funcionalidades generales del sistema, un manual de usuario debería poder ser entendido por cualquier usuario principiante, así como también serles útil a usuarios avanzados.

Manual de Instalación

El cual especifica los requerimientos y la configuración necesaria para la instalación del sistema, empezando por los recursos hardware, la plataforma operativa indicada para el despliegue, la configuración del entorno software

necesario y el levantamiento de los servicios necesarios para el consumo desde los clientes definidos.

Instalación

Procedimientos y Procesos

El manual de procedimientos y procesos que posee la empresa debe ser actualizado tomando en cuenta a la nueva solución informática incorporada, para que de esta forma el personal operativo tenga una base firme de consulta sobre el nuevo curso de actividades a desarrollarse en la empresa.

Historial del Soporte Técnico

Todo aspecto importante efectuado durante ésta fase debe ser debidamente registrada, ayudando a manejar de forma centralizada las actividades que se han ido cumpliendo en base al tiempo previsto para la presente etapa.

ANEXO B

Manual de Usuario Sistema Satélite Contilibro

1. PRESENTACIÓN

El presente manual está destinado al personal que tendrá acceso al sistema satélite de Contilibro. Aquí se detalla el uso del sistema, el mismo que está destinado a automatizar los procesos de inventario y de facturación de la empresa que se han venido haciendo de manera manual para de ésta manera ahorrar tiempo y recursos.

Para garantizar el correcto funcionamiento del sistema se presenta en este documento una guía acerca de su funcionalidad, por lo cual se recomienda que antes de su utilización se lea y analice de forma completa y detenida, con la finalidad de evitar problemas futuros y sacarle el mejor provecho.

2. INTRODUCCIÓN

La empresa Contilibro se encarga de la venta y distribución de libros a nivel nacional por lo que se ha visto en la necesidad de contar con un sistema informático que permita llevar un control de inventarios y facturación y de esta manera dejar de realizar los procesos anteriormente mencionados de manera manual, ya que esto conlleva a pérdida de tiempo y de recursos para la empresa.

La solución que se plantea para la empresa Contilibro, es la creación de un sistema satélite que permita llevar el control de inventarios y facturación; así como también permita llevar un control de proveedores y clientes de la empresa.

3. OBJETIVOS DEL SISTEMA

Con el sistema se logrará conseguir que:

- ✓ Los procesos de inventarios y de facturación de la empresa se realicen de manera automatizada y rápida.
- ✓ La información de clientes, productos(libros) y proveedores estará debidamente actualizada.
- ✓ Consultas y Reportes de cualquier tipo de información que se encuentre ingresada en el sistema.
- ✓ El sistema tendrá la capacidad de crear tipos de usuarios, lo que permitirá administrar de manera correcta el sistema.
- ✓ La información que se maneje en la empresa será más real y con un porcentaje muy bajo de errores.

4. CAPACIDADES DEL SISTEMA

El sistema posee las siguientes capacidades:

- ✓ Permitir el acceso únicamente a personal encargado de manejar el sistema.
- ✓ Llevar un control automatizado de los procesos de inventario y facturación de la empresa.
- ✓ Realizar búsquedas de libros, clientes, proveedores.
- ✓ Controlar campos vacíos
- ✓ Controlar datos erróneos

- ✓ Emitir reportes de libros, clientes, proveedores.
- ✓ Emitir proformas.

5. REQUISITOS MÍNIMOS INDISPENSABLES

Hardware

- ✓ Procesador Pentium IV de 800 Mhz 1GB en RAM.
- ✓ 1GB de espacio libre en el disco
- ✓ Monitor con una resolución de 1280 * 800 pixeles.

Software

- ✓ Sistema Operativo Microsoft Windows XP/2002
- ✓ Microsoft .Net Framework 2.0.

6. ESTRUCTURA DEL SISTEMA

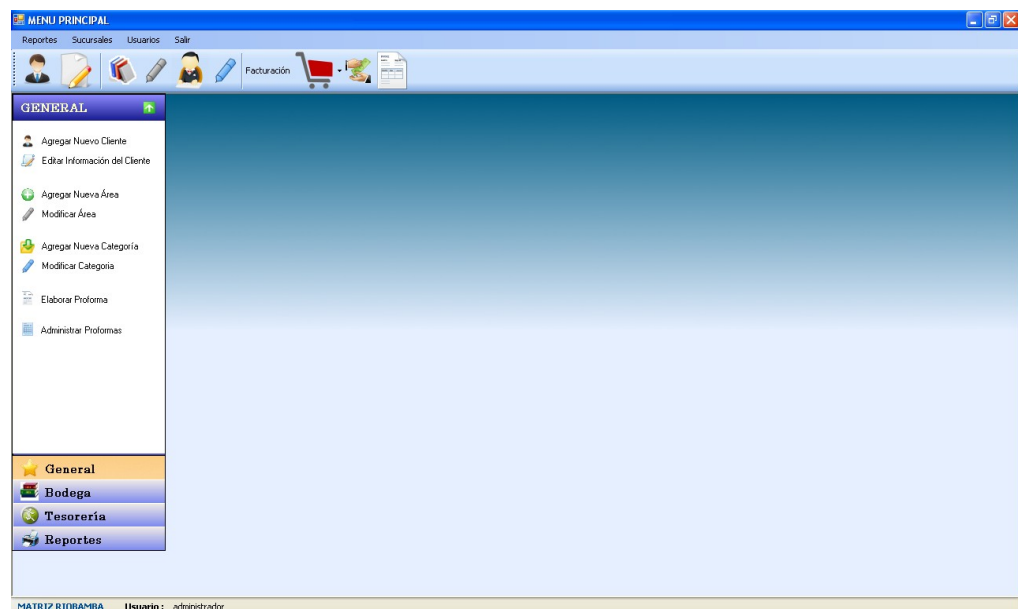
El sistema está desarrollado para manejar dos tipos de usuarios como son el usuario administrador y el usuario vendedor los mismos que podrán acceder al sistema ingresando su nombre de usuario, contraseña, tipo de usuario y sucursal a la que pertenecen, después de haber ingresado los datos de inicio correctamente se desplegará un menú de opciones dependiendo del tipo de usuario que haya ingresado, de esta manera:

6.1 Menú Administrador

Inicio de Sesión Administrador



Al ingresar como administrador se desplegará un menú de opciones de esta manera:

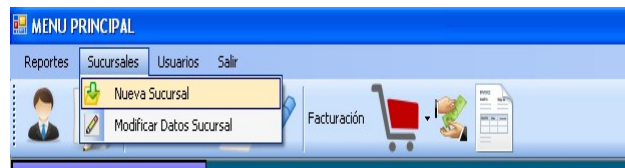


El menú le permitirá al usuario administrador gestionar la información de usuarios, sucursales y también le permitirá generar reportes.

Menú Sucursales:

En este menú se podrá ingresar una nueva sucursal así como también modificar los datos de una sucursal existente:

Ingresar una nueva Sucursal:



Al hacer click en esta opción se desplegará una pantalla en la cual se deben ingresar los datos de la nueva sucursal que se desea ingresar y seguidamente se debe hacer click en el botón aceptar para que se registre en la base de datos.

The image shows a form window titled 'Sucursal' with the subtitle 'Ingrese los Datos de la Nueva Sucursal'. The form has four input fields: 'ID Sucursal' with the value '112', 'Nombre Sucursal' with the value 'Sucursal Ambato', 'Direccion' with the value 'Avenida Cevallos', and 'Telefono' with the value '032987678'. At the bottom of the form, there are three buttons: 'Aceptar' (with a checkmark icon), 'Cancelar' (with a red X icon), and 'Cerrar' (with a close icon).

Modificar datos de una sucursal:

Al hacer click en esta opción se desplegará una pantalla que permitirá buscar la sucursal que se desea editar.

Modificar Sucursal

Buscar por:

ID Nombre

Ingrese el nombre

ID Bodega	Nombre Bodega	Direccion	Telefono
1	Matriz Riobamba	Guayaquil 25-40 y España	032955198
4	Sucursal Ambato	Av. Numero 1 y 9 de Octubre	2948956
3	Sucursal Riobamba Centro	Guayaquil y Garcia Moreno	2689546
2	Sucursal Riobamba Politecnica	Panamericana km 1 1/2	2948000

Modifique los Datos de la Sucursal Seleccionada.

ID:

Nombre Sucursal:

Direccion:

Telefono:

Menú Usuarios:

En este menú se da la opción de ingresar un nuevo usuario, modificar los datos de un usuario y eliminar a un usuario.



Nuevo Usuario:

Al hacer click en esta opción se desplegará una pantalla que permitirá ingresar los datos del nuevo usuario que se desea registrar.

Usuario

Ingrese los Datos del Nuevo Usuario

ID. Usuario:

Login:

Contraseña:

Tipo Usuario:

Sucursal:

Modificar Usuario:

Permitirá modificar la información de un usuario que ya está registrado.

Modificar Usuario

Buscar por:

ID Nombre

Ingrese el nombre

ID_USUARIO	LOGIN	TIPO USUARIO	SUCURSAL
2	vendedor2	Vendedor	Matriz Riobamba
3	vendedor3	Vendedor	Sucursal Riobamba Politecnica

Modifique los Datos del Usuario Seleccionado.

ID:

Login:

Tipo Usuario:

Sucursal:

Contraseña:

¿Desea cambiar de contraseña?

Nueva Contraseña:

Confirmar Contraseña:

Eliminar Usuario:

Permitirá eliminar un usuario que se encuentre registrado en la base de datos.

Eliminar Usuario

Buscar por:

ID Nombre

Ingrese el nombre

ID_USUARIO	NOMBRE	TIPO USUARIO	SUCURSAL
2	vendedor2	Vendedor	Matriz Riobamba
3	vendedor3	Vendedor	Sucursal Riobamba Poli...

Datos del Usuario

ID:

Nombre:

Tipo Usuario:

Menú Reportes:

En este menú se podrá escoger entre distintos tipos de reportes que se generarán según el usuario los escoja.



Al hacer click en cualquiera de las opciones se desplegará una pantalla con los datos del reporte seleccionado.

CIRUC	NOMBRE	APELLIDO	DIRECCION	CELULAR	EMAIL
06033821-7	Angel Eduardo	Apina Reino	Orozo 15-39 y Morona	095280188	angel_apina2884@hotmail.com
060332955-8	dede	ded	dede	098098098	ded
060332933-4	juan	lopez	la estacion	098880000	jlopez@hotmail.com
060445986-9	Patricia	Machado	La Estacion	084009341	pmachado@gmail.com
140050776-9	Mirian Susana	Riera Idrovo	Braail y Orozoo	084009341	mirian_susan15@hotmail.com
060332933-5	Richard Fernando	Robles Carrillo	10 de Agosto 0915 y Av. Eloy Alfaro	087892834	rfr22@gmail.com
040049168001	Giovani	Rodriguez	La Dolorosa	098098098	grodriiguez@gmail.com

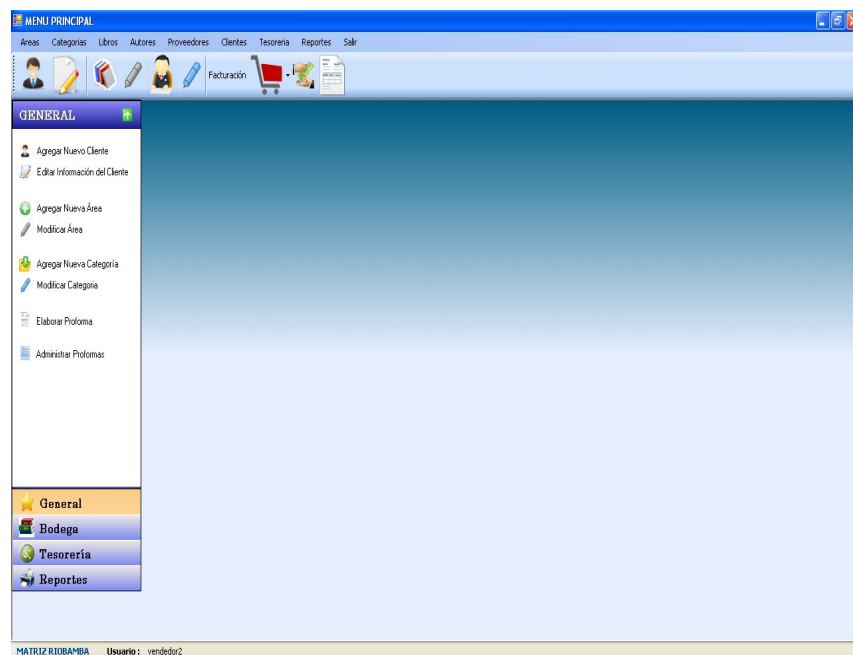
6.2 Menú Vendedor

Al acceder como un usuario vendedor se podrá gestionar la información de áreas, categorías, clientes, proveedores, libros; así como también se podrá realizar ventas y generar reportes.

Para acceder al menú del usuario vendedor primeramente se deberá ingresar como usuario vendedor e introducir la información requerida.

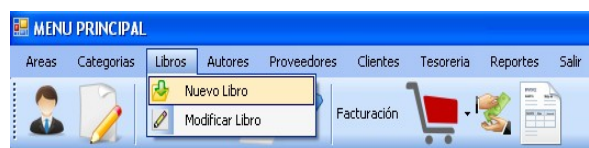


Si los datos fueron ingresados correctamente se mostrará una pantalla con su respectivo menú de opciones.



Aquí se podrá gestionar la información de áreas, categorías, libros, proveedores, clientes, realizar ventas y generar reportes. La manera de gestionar la información es similar lo único que se debe hacer es escoger una opción del menú e ingresar la información necesaria, por tal razón en este manual únicamente se demostrará la utilización del menú libros debido a que el proceso es similar con las opciones de área, categoría, cliente y proveedor.

Menú libros:



Nuevo Libro:

Al hacer click en esta opción se desplegará una pantalla en la cual se deberá ingresar la información necesaria para el registro de un nuevo libro.

Ingreso de Mercadería -Matriz Riobamba-

Ingrese los datos del nuevo Libro

Portada

Proveedor: Proveedor1 +

ISBN: 1234567890

Tomos: 2_

Precio: 12,50

Cantidad: 10_

Título: Aplicaciones WEb

Editorial: Edit sof

Año: 2011

Sinopsis: Se explica de manera concreta todo lo referente a aplicaciones web en la tecnología ASP

Autor: Gates

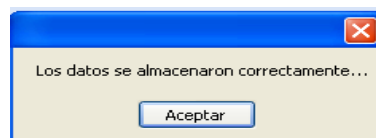
Páginas: 250

Área: Informática +

Tipo: AES +

✓ Aceptar ✗ Cancelar ✕ Cerrar

Si la información se ingresó correctamente se desplegará un mensaje de confirmación.

Modificar Libro:

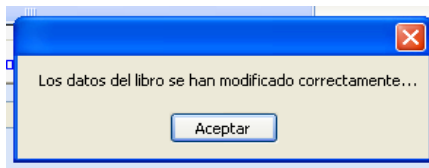
Al hacer click en esta opción se mostrará una pantalla en la que se deberá editar la información del libro que se desea, para esto primeramente se debe realizar una búsqueda del libro deseado.

La búsqueda se la podrá realizar por código, título, autor, área o categoría.

The screenshot shows a software interface for managing books. At the top, there are radio buttons for 'Código', 'Título', 'Autor', and 'Área', with 'Título' selected. Below this is a search bar with the text 'Ingrese el Título' and a search button. A table displays search results with columns for ISBN, AUTOR, CATEGORIA, TITULO, EDITORIAL, PRECIO, and AÑO. The row with ISBN '2' is highlighted. Below the table, there is a section titled 'Modifique los Datos del Libro Seleccionado.' containing a form with fields for ISBN, Tomos, Precio, Cantidad en Bodega, Autor, Título, Editorial, Año, Páginas, Sinopsis, Área, and Categoría. There are 'Modificar' and 'Cerrar' buttons at the bottom of the form.

ISBN	AUTOR	CATEGORIA	TITULO	EDITORIAL	PRECIO	AÑO
12345	autor3	Base de Datos	ASP	EDIT	24,50	201
4444	autor8	Calculo Integral	Calcul Integral	Edit	34,60	201
2	autor8	Economía Domés...	Cuidado Ambiental	WWW	15,00	201
2222	autor	Marketing de Ser...	Economía	e	10,25	201
007	Advanced	AES	encriptacion	edit	10,25	201
1234567891234	autor4	Calculo Integral	Finanzas Internac...	"Editorial Pru...	42,00	200
134567890123	autor5	Multimedia	Joombia	"Editorial de ...	45,00	201

Si los datos fueron editados correctamente se mostrará un mensaje de confirmación.



En esta sección se explicará la manera de realizar ventas, devoluciones y facturación.

Facturación de libros consignados:



Para realizar este proceso se debe hacer click en la opción facturación y escoger libros consignados, al hacer esto se mostrará una pantalla en la que primeramente se deberá escoger el cliente de esta manera.

Buscar Cliente

Buscar por:

Cédula - RUC Apellido

Ingrese el apellido del cliente

CEDULA / RUC	NOMBRE	APELLIDO	TELEFONO	DIRECCION	TELEFONO DE REFERENCIA	EMAIL
0400493169001	Giovani	Rodriguez	032987789	La Dolorosa	022345678	grodriguez@gmail...
060332933-4	Juan	lopez	032989989	la estacion	032999888	jlopez@hotmail.com
060333521-7	Angel Eduardo	Apina Reino	2948595	Orozco 15-39 y Morona	29519440	angel_apina2884...

Una vez escogido el cliente se mostrará la siguiente pantalla en la que aparecerá los libros que el cliente ha adquirido a consignación.

Factura | Forma Pago

Cabecera

Tipo de venta: Contado Credito

Cédula RUC: 0400493169001

Número de Factura: 12344

Cliente: Giovani Rodriguez

Fecha de Emisión: martes, 02 de agosto de 2011

Teléfono: 032987789

Fecha de Vencimiento: martes, 02 de agosto de 2011

Dirección: La Dolorosa

Detalle

ISBN	Detalle / Obra	Precio	Cantidad	Total
121212	paginas dinamicas	23,89	1	23,89
12345	ASP	24,50	2	49,00
1234567891234	Finanzas Internacionales	42,00	1	42,00

Sub Total	114,89
IVA	0,00
Descuento	---
TOTAL	114,89

Para completar con el proceso se deberá especificar la forma de pago, para realizar esto se debe hacer click en la pestaña forma de pago y se mostrará una pantalla en la que se deberá especificar la manera en que se va a cancelar el total de la deuda.

De la misma manera que en la compra de libros a consignación también se debe especificar la forma en la que se va a realizar el pago, esto se lo realiza haciendo click en la pestaña forma pago.

The screenshot shows a web application window titled 'Facturación' with a sub-tab 'Forma de Pago'. It contains three main sections for payment methods:

- Monto en Efectivo:** A text input field containing '7.90'.
- Monto en Cheque:** A text input field containing '40.00'. Below it, a 'Datos' section includes:
 - Banco: Banco Pro Credit
 - # Cheque: 1234567
 - Nombre: Patricia Machado
- Monto en Tarjeta de Crédito:** A text input field containing '50.00'. Below it, a 'Datos' section includes:
 - Numero de Tarjeta: 123456
 - Numero de Voucher: 654321
 - Nombre: Patricia
 - Apellido: Machado
 - Fecha de Caducidad: martes .02 de agosto de 2011

At the bottom, a summary shows: Efectivo: 7.90, Cheque: 40.00, and Tarjeta de Crédito: 50.00. A green checkmark and the text 'El monto de la factura se ha cubierto en su totalidad..' indicate the total amount is covered.

De esta manera se completa el proceso de facturación de los libros cuando se ha realizado una venta directa.

Realizar abonos

Esta opción permite realizar abonos de las compras que se han hecho a crédito la misma que generará un recibo, para realizar esta operación primeramente se debe buscar el cliente y registrar el abono que se va a realizar, de igual manera se tiene la opción de elegir la forma de pagar el abono en la ventana correspondiente.

Añadir Recibo

Factura | Forma de Pago

Datos del Comprobante de Pago

Num. Comprobante: 001 Cédula/RUC: 060445566-9

Fecha de Emisión: martes , 02 de agosto de 2011 Cliente: Patricia Machado

Fecha Proximo Pago: martes , 02 de agosto de 2011 Teléfono: 032628006

Numero de Factura: 123354858 Dirección: La Estacion

Por Concepto de: pago cuota

Detalle de Libros Facturados

ISBN	Detalle / Obra	Cantidad	Precio	Total
12345	ASP	1	24,50	24,50

Saldo Anterior: 22,50 Abono: 22,50 Saldo Deudor: 0,00

Añadir Recibo

Factura | Forma de Pago

Monto en Efectivo: 22,50

Monto en Cheque: _____

Datos:

Banco: _____

Cheque: _____

Nombres: _____

Monto en Tarjeta de Crédito: _____

Datos:

INDIQUE LA TARJETA QUE CORRESPONDA

AMERICAN EXPRESS Diners Club International MASTERCARD VISA

CORRIENTE DIFERIDO PLAN PAGOS OTROS

SI ES DIFERIDO

CON INTERES SIN INTERES

INDIQUE EL NUMERO DE MESES

3 6 9 12 18 24 OTROS

Efectivo: 22,50

Cheque: 0,00 El monto del recibo se ha cubierto en su totalidad..

Tarjeta de Crédito: 0,00

Menú consignación

Esta opción permite realizar compras de libros a consignación para lo cual primeramente se deberá buscar al cliente que desea realizar la compra y después se mostrará un pantalla que permitirá elegir los libros que se desea llevar a consignación.

Fecha de Consignación: martes ,02 de agosto de 2011

Cédula/RUC: 060332933-4

Cliente: juan lopez

Teléfono: 032989989

Dirección: la estacion

Detalle

CANT.	CODIGO	DETALLE/OBRAS	TMS	PRECIO	TOTAL
1	001	título1	2	32,20	32,20
2	002	título2	1	41,20	82,40
3	12345	ASP	1	24,50	73,50
Sub Total					188,10

Menú devoluciones:

En este menú se despliegan dos opciones que son la devolución de libros consignados y la devolución de los libros que se han facturado.

Devolución de libros consignados:


Permite realizar la devolución de los libros que se han dado a consignación, para realizar este proceso se debe primeramente buscar al cliente que ha realizado la consignación y registrar la devolución.

Devolución de Libros (Bodega: 1 "Matriz Riobamba")

Cabecera

Fecha de Devolución: martes , 02 de agosto de 2011
 Cédula RUC: 060332933-4
 Cliente: juan lopez
 Teléfono: 032989989
 Dirección: la estacion

Detalle

 **Deshacer Todo**

Fecha de Consignación	Id Cargo	ISBN	Detalle / Obra	Precio	Total	Cantidad Consignada	Cantidad a Devolver	Bodega
19/05/2011	ca1-5	121212	paginas dinamicas	23.89	23.89	1	0	1

Para ingresar la cantidad de libros que se desea devolver se debe hacer doble click en la fila correspondiente al libro que se desea devolver, al hacer doble click aparecerá una pantalla con la información principal del libro además de una caja de texto en donde se podrá elegir la cantidad de unidades que se desea devolver.

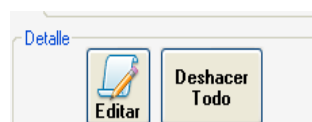
Editar Devolución

Información del Libro

ISBN: 121212
 TITULO: paginas dinamicas
 AUTOR: autor2
 AREA: Economía II
 CATEGORIA: Banca
 PRECIO: 23.89

Cantidad a Devolver

Esa operación también se puede realizar haciendo click en el botón editar.



Devolución de libros facturados:

Esta opción permite realizar la devolución de los libros que ya se han facturado, para realizar este proceso se debe primeramente buscar el cliente, una vez elegido el cliente aparecerá la siguiente pantalla.

Cabecera

Nota de Crédito N°:

Mostrar las facturas de: 6 meses atrás.

Factura N°:

Fecha: martes, 02 de agosto de 2011

Concepto:

Cliente CI / RUC: 060445566-9

Nombre: Patricia Machado

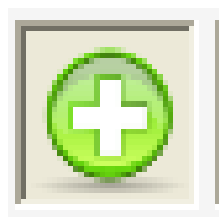
Teléfono: 032628006

Dirección: La Estacion

Detalle

Fecha	ISBN	Detalle / Obra	Cantidad	Precio	Total	Cantidad a Devolver	Bodega

Al realizar la devolución se generará una nota de crédito. Para añadir libros a la nota de crédito se debe hacer click en el botón de buscar.



Al hacer click aparecerá una pantalla que permitirá elegir el modo de búsqueda de los libros.

LIBROS

Seleccione el tipo de Búsqueda

Por Código Lista

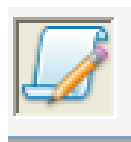
Al seleccionar el tipo de búsqueda se añadirá el libro a la lista.

Fecha	ISBN	Detalle / Obra	Cantidad	Precio	Total	Bodega
20/07/2011 11:35	003	tit	2	32,40	64,80	1

Una vez seleccionado los libros se mostrarán en la pantalla para de esta manera generar la nota de crédito correspondiente.

Fecha	ISBN	Detalle / Obra	Cantidad	Precio	Total	Cantidad a Devolver	Bodega
20/07/2011 11:3...	003	tit	2	32,4	64,8	1	1

Se podrá también editar la cantidad de libros que se desea devolver haciendo click en el siguiente botón.

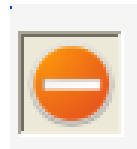


Al hacer click en este botón se mostrará una pantalla en la cual se mostrará la información principal del libro, en esta pantalla se podrá editar la cantidad de libros que se desea devolver.

The screenshot shows a window titled "Editar Devolución" (Edit Return). It is divided into two sections:

- Información del Libro (Book Information):**
 - ISBN: 000
 - TITULO: tit
 - AUTOR: autor
 - AREA: Economía II
 - CATEGORIA: Banca
 - PRECIO: 32,4
- Cantidad a Devolver (Quantity to Return):**
 - An "Aceptar" (Accept) button with a checkmark icon.
 - A dropdown menu showing the number "1".

También se da la opción de eliminar todos los libros de la lista haciendo click en el botón eliminar.



El usuario vendedor también tiene la posibilidad de generar reportes, para esto hay que hacer click en el menú deportes y elegir el tipo de reporte que se desea generar.



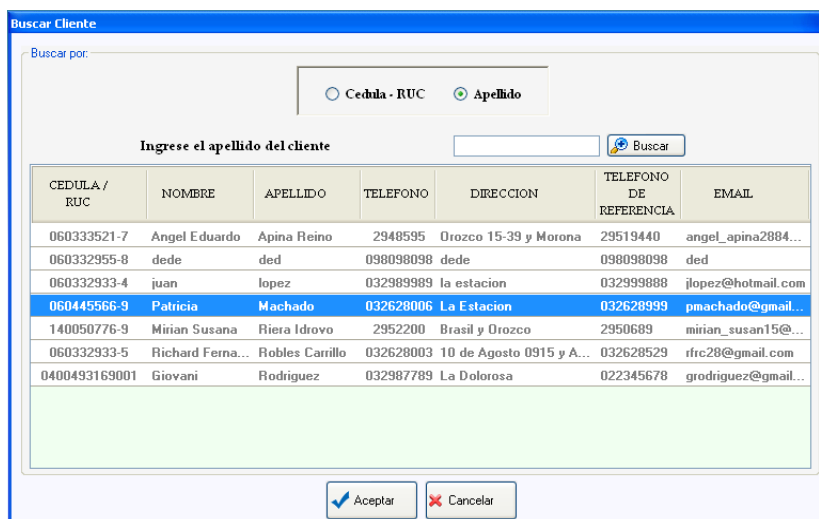
Por ejemplo en esta ocasión se ha escogido el reporte de los proveedores, el cual se generará al hacer click en la opción lista de proveedores.



En el menú de reportes también existe la posibilidad de generar proformas para esto se debe hacer click en la opción proformas.



Al hacer click en la opción proformas se mostrará la siguiente pantalla en donde se deberá escoger el nombre del cliente.



Al hacer click en aceptar se mostrará una pantalla en la cual se deberá buscar los libros que se desean agregar a la proforma, para esto se han incluido distintas formas de búsqueda para los libros.

Formulario de búsqueda de libros. Campos: ISBN (input), Área (dropdown: Informática), Tipo (dropdown: Base de Datos), Autor (input), Título (input). Botón: Buscar.

Una vez que se elija el criterio de búsqueda y dar click en el botón aceptar, aparecerá en una lista los libros que se hayan encontrado según el criterio de búsqueda de esta manera.

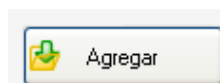
ISBN	Título	Autor	Área	Categoría	Precio	Stock	Cantidad	
005	titulo nuevo	aq	Informática	Base de Datos	5,00	3	0	<input type="checkbox"/>
12345	ASP	autor3	Informática	Base de Datos	24,50	12	0	<input checked="" type="checkbox"/>

Seleccionar Todos:

Para elegir los libros que se desea agregar a la proforma se debe marcar los casilleros correspondientes o también se podrá elegir todos haciendo click en el check seleccionar todos, una vez hecho esto se tendrá que especificar la cantidad de libros que se desea agregar a la lista, para esto se debe hacer click en la celda cantidad y escribir las unidades que se desea agregar.

ISBN	Título	Autor	Área	Categoría	Precio	Stock	Cantidad	
005	titulo nuevo	aq	Informática	Base de Datos	5,00	3	2	<input checked="" type="checkbox"/>
12345	ASP	autor3	Informática	Base de Datos	24,50	12	3	<input checked="" type="checkbox"/>

Para agregar los libros a la proforma se debe hacer click en el botón agregar.



Al hacer click en agregar los libros que han sido seleccionados se pasarán a la lista de libros de la proforma.

Lista de Libros Consignados

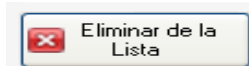
ISBN	Título	Precio	Cantidad	Total
005	titulo nuevo	5	2	10
12345	ASP	24,5	3	73,5

TOTAL 83,50

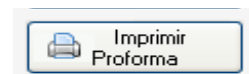
Una vez hecho esto se podrá limpiar toda la lista haciendo click en el botón limpiar lista.



También se da la opción de eliminar solo un libro de la lista, para esto se debe hacer click en el botón eliminar de la lista.



Finalmente se podrá generar la factura haciendo click en el botón imprimir proforma.



Al hacer click en imprimir proforma se generará la proforma y se mostrará en pantalla.



CONTILIBRO

PROFORMA DE LIBROS

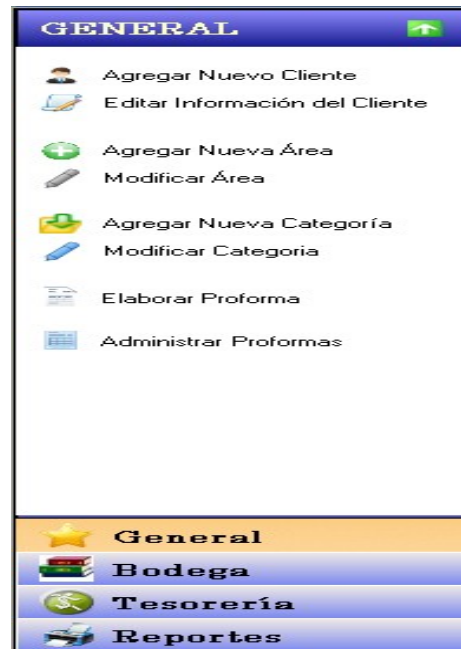
Datos del Cliente:

Nombre Cliente:	Patricia Machado
Cédula/RUC:	060445566-9
Teléfono:	032628006
Dirección:	La Estacion
Fecha:	02-agosto-2011

ISBN	TITULO	PRECIO	CANTIDAD	TOTAL
005	titulo nuevo	5	2	10
12345	ASP	24,5	3	73,5
TOTAL :				83,50

El sistema satélite también cuenta con un menú rápido para acceder a las opciones con mayor facilidad.

Menú que se encuentra en la parte izquierda de la pantalla principal.

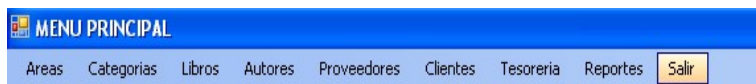


Menú que se encuentra en la parte superior de la pantalla principal.



- ✓ 1: Agregar nuevo cliente.
- ✓ 2: Modificar datos del cliente.
- ✓ 3: Agregar un libro.
- ✓ 4: Modificar un libro.
- ✓ 5: Facturación
- ✓ 6: Abonos
- ✓ 7: Proformas

Para abandonar el sistema únicamente se debe hacer click en la opción salir del menú principal.



ANEXO C

Manual de Usuario Herramienta de Sincronización de Contilibro

1. PRESENTACIÓN

El presente manual está destinado al personal que tendrá acceso a la herramienta de sincronización de Contilibro. Aquí se detalla el uso de la herramienta, la misma que tiene por objetivo mantener sincronizadas las bases

de datos de los sistemas satélite de Contilibro (MSSQL Server) y de la aplicación Web (MYSQL 5.1).

Para garantizar el correcto funcionamiento de la herramienta se presenta en este documento una guía acerca de su funcionalidad, por lo cual se recomienda que antes de su utilización se lea y analice de forma completa y detenida, con la finalidad de evitar problemas futuros y sacarle el mejor provecho.

2. INTRODUCCIÓN

La empresa Contilibro que se encarga de la venta y distribución de libros a nivel nacional cuenta en la actualidad con sistema satélite que lleva el control de inventario y ventas que se realizan, el sistema satélite tiene como repositorio de datos una base de datos que se encuentra en el motor MSSQL Server, Contilibro también cuenta con una página Web que consume los datos de una base de datos en MYSQL 5.1 las dos bases de datos tienen la misma estructura y cuentan con las mismas tablas, el objetivo de la herramienta de sincronización es lograr sincronizar los dos motores de bases de datos de tal manera que en algún instante de tiempo los dos motores de bases de datos deben contar con la misma información.

La solución que se plantea es la utilización de servicios Web los mismos que estarán encargados de realizar el proceso de replicación de datos, además la herramienta cuenta con una agenda de programación que permite elegir el instante en que se realizará la replicación de datos; esta agenda de programación está controlada por servicios Windows los mismos que una vez programados permitirán que la réplica de datos se realice de manera automática.

3. OBJETIVOS DE LA HERRAMIENTA

Los objetivos de la herramienta son:

- ✓ Registrar los servidores de base de datos que estarán presentes en el proceso de sincronización de datos.
- ✓ Capturar los cambios que se realicen en cada una de las bases de datos.
- ✓ Mezclar los cambios que se hayan capturado.
- ✓ Sincronizar las bases de datos, tanto de los sistemas satélite como de la aplicación Web de Contilibro.
- ✓ Configurar una agenda que permita realizar el proceso de sincronización de manera automática.

4. CAPACIDADES DE LA HERRAMIENTA

La herramienta permitirá:

- ✓ Registrar los servidores que participarán en el proceso de sincronización.
- ✓ Capturar los cambios de cada una de las bases de datos, mezclar los cambios.
- ✓ Sincronizar los cambios en cada una de las bases de datos.
- ✓ Elegir una agenda de programación para que se realice el proceso de sincronización.

5. REQUISITOS MÍNIMOS INDISPENSABLES

Hardware

- ✓ Procesador Pentium IV de 800 Mhz 1GB en RAM.
- ✓ 1GB de espacio libre en el disco
- ✓ Monitor con una resolución de 1280 * 800 pixeles.

Software

- ✓ Sistema Operativo Microsoft Windows XP/2002
- ✓ Microsoft .Net Framework 2.0.

6. ESTRUCTURA DE LA HERRAMIENTA

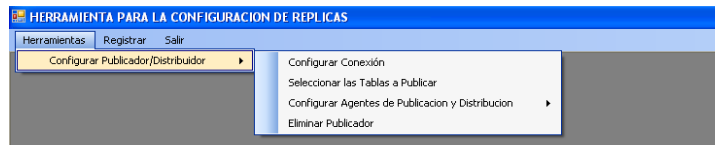
La herramienta está desarrollada para manejar un tipo de usuario Administrador el mismo que podrá acceder a la herramienta que cuenta con un menú de opciones que le permitirá programar la agenda para el proceso de sincronización y también le permitirá configurar las bases de datos, esto es elegir el publicador, distribuidor y suscriptor que participarán en el proceso.

6.3 Inicio del proceso de sincronización

Para iniciar el proceso de sincronización primeramente se debe escoger qué servidor de base de datos va a tener el papel de publicador, distribuidor y suscriptor, para realizar esta tarea la herramienta cuenta con un menú de opciones el cual permitirá configurar los servidores y de esta manera iniciar con el proceso de sincronización.

Configuración del Publicador:

Para configurar el publicador se debe elegir el menú herramientas y escoger configurar publicador.



Al elegir esta opción se desplegará un submenú que permitirá configurar la conexión con el servidor que actuará como publicador y distribuidor al mismo tiempo, también permitirá seleccionar las tablas que se desean publicar, así como también configurar los agentes de publicación y distribución y eliminar la publicación.

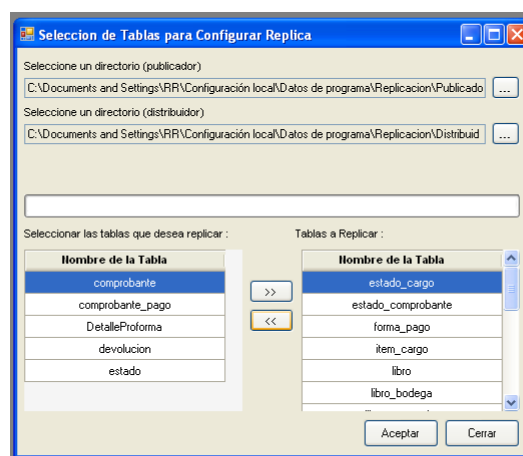
Configurar Conexión:

Esta opción permitirá elegir el servidor que tendrá el papel de publicador y distribuidor al mismo tiempo, para esto se debe ingresar la información requerida y hacer click en el botón aceptar.

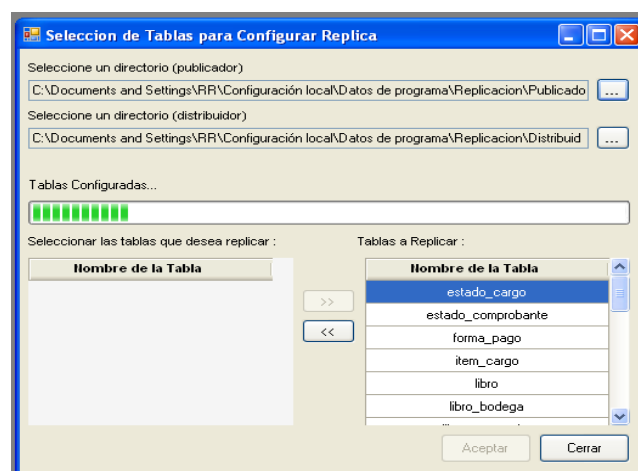
A screenshot of a dialog box titled "Conexión". It is divided into two main sections: "Base de Datos" and "FTP".
In the "Base de Datos" section:
- "Origen de Datos:" is set to "Microsoft SQL Server (SqlClient)" with a "Cambiar...." button.
- "Nombre del Servidor:" is "rfr28".
- "Conexión con el Servidor" section contains:
 - "Nombre de Usuario:" "sa"
 - "Contraseña:" (masked with "|||")
- "Establecer la conexión con una Base de Datos" section contains:
 - "Nombre de la Base de Datos:" "contilibroBD"
In the "FTP" section:
- "Host FTP:" "wscontilibro.somee.com"
- "Directorio FTP:" "ftp://wscontilibro.somee.com"
- "Credenciales" section contains:
 - "Nombre de Usuario:" "basecontilibro"
 - "Contraseña:" (masked with "|||||")
At the bottom are three buttons: "Probar Conexión", "Aceptar", and "Cancelar".

Seleccionar tablas:

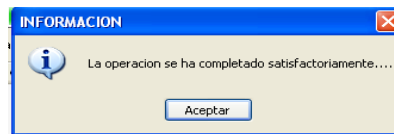
Esta opción permite elegir las tablas que se publicarán, las mismas que participarán en el proceso de sincronización para lo cual la herramienta añadirá campos adicionales los mismos que permitirán saber con certeza que acciones (inserción, actualización o eliminación) se realizaron sobre las tablas; además también se crearán triggers por cada tabla que se publique de manera que cuando se produzca algún cambio este quede capturado y registrado en los campos adicionales que se añadieron.



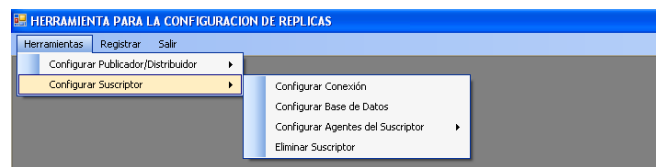
Una vez seleccionadas las tablas se debe hacer click en el botón aceptar para comenzar con el proceso anteriormente descrito.



Al completar el proceso anterior se presenta un mensaje de confirmación.

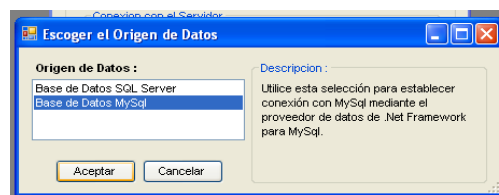


Para seguir con el proceso de sincronización se necesita especificar qué servidor actuará como suscriptor para lo cual se debe elegir en el menú herramientas la opción configurar suscriptor que desplegará un submenú con diferentes opciones de configuración.



De la misma manera que en el publicador se debe primeramente elegir el servidor que actuará como suscriptor, para esto se debe elegir la opción configurar conexión; al realizar esto se mostrará un formulario en el que se debe ingresar los datos del servidor de suscripción.

En este caso el servidor que realizará el papel de suscriptor es un servidor MYSQL, por lo que se debe primeramente especificar el origen de datos.



Una vez terminado con este proceso los servidores ya están en capacidad de registrar los cambios que se realicen en la base de datos de Contilibro; el siguiente proceso que se debe realizar es capturar los cambios de los servidores publicador y suscriptor; para esto se debe hacer click en el menú registrar para capturar los cambios del publicador y del suscriptor respectivamente.

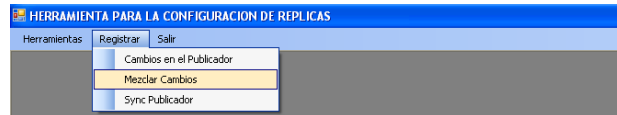


Al completarse este proceso aparecerá un mensaje de confirmación.

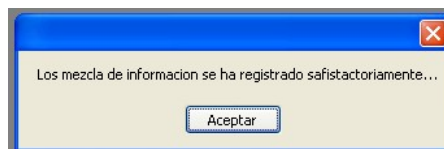


Si se trata de un servidor publicador se deberá escoger la opción de mezclar los cambios del menú; la misma que permitirá realizar la mezcla de

datos de manera que cuando finalice este proceso se tenga un archivo con todos los cambios registrados por todos los servidores.



Si el proceso se realizó correctamente se mostrará un mensaje de confirmación:

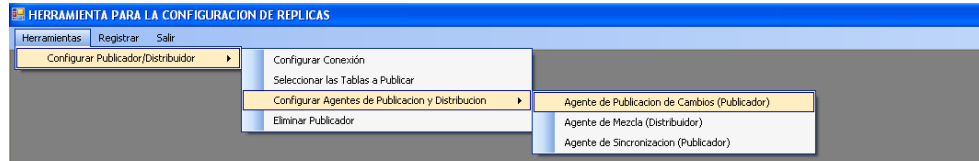


Después se deberá hacer click en la opción Sync Publicador para que de esta manera el servidor pueda sincronizar su información.



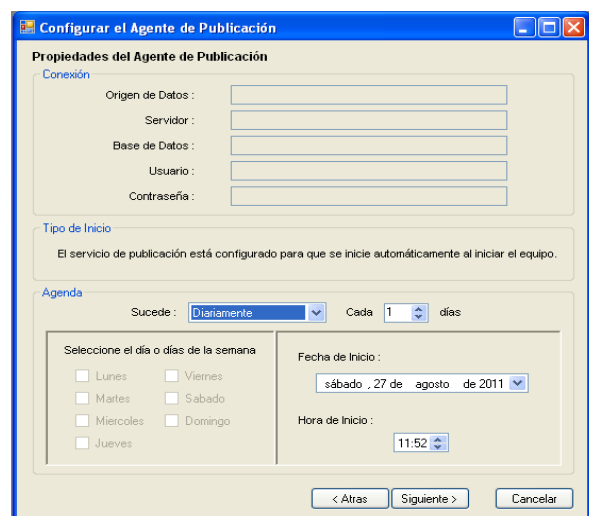
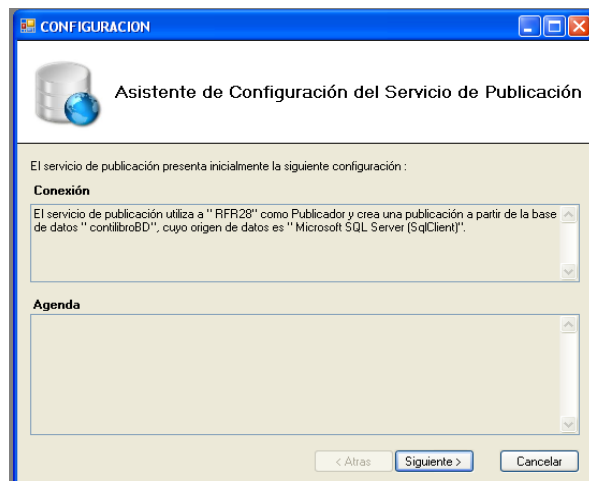
Para realizar el proceso de sincronización del suscriptor se debe hacer el mismo proceso anteriormente mencionado en el menú del suscriptor.

La herramienta también presenta una opción de configuración de agentes de publicación que permitirá programar una agenda de manera que el proceso anteriormente descrito se realice automáticamente dependiendo de los parámetros que se elija. Para esto se debe hacer click en la opción configurar agentes de publicación, suscripción respectivamente.



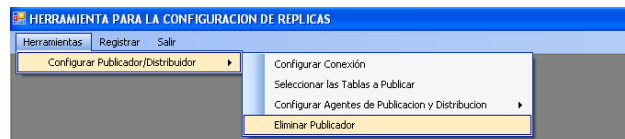
Al realizar esto se desplegará otro submenú que permitirá configurar los agentes de publicación de cambios, de mezcla y de sincronización.

Para programar la agenda se debe seleccionar el agente y elegir la agenda de programación más adecuada.



Una vez completado este proceso la herramienta ya se encuentra correctamente configurada de manera que automáticamente se realizará el proceso de sincronización de los servidores que están registrados.

La herramienta también permite eliminar un servidor ya sea este publicador o suscriptor para esto se debe hacer click en la opción del menú eliminar publicador o suscriptor respectivamente.



De esta manera es como se completará el proceso de sincronización de los servidores.

GLOSARIO

ASP:

También conocido como ASP clásico, es una tecnología de Microsoft del tipo "lado del servidor" para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Services (IIS).

HTML:

Es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

HTTPS:

Protocolo seguro de transferencia de hipertexto), más conocido por sus siglas HTTPS, es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hiper Texto.

W3C:

El World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce recomendaciones para la World Wide Web.

SQL:

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés structured query language) es un lenguaje declarativo de acceso a bases de datos relacionales que

permite especificar diversos tipos de operaciones en estas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar de una forma sencilla información de interés de una base de datos, así como también hacer cambios sobre ella.

AES:

Advanced Encryption Standard (AES), también conocido como Rijndael (pronunciado "Rain Doll" en inglés), es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos.

SOAP:

SOAP (siglas de Simple Object Access Protocol) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de un protocolo creado por David Winer en 1998, llamado XMLRPC. SOAP fue creado por Microsoft, IBM y otros y está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web.

WSDL:

WSDL son las siglas de Web Services Description Language, un formato XML que se utiliza para describir servicios Web. WSDL describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo.

XML:

Son las siglas de Extensible Markup Language, una especificación/lenguaje de programación desarrollada por el W3C. XML es una versión de SGML, diseñado especialmente para los documentos de la web. Permite que los diseñadores creen sus propias etiquetas, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones.

Servicio Web:

Un servicio web (en inglés, Web service) es una pieza de software que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet.

Sincronización:

La sincronización de archivos es utilizada para mantener la misma versión de archivos en múltiples dispositivos. Por ejemplo, sincronizar la libreta de dirección de un teléfono con la libreta de direcciones de una computadora.

BIBLIOGRAFIA

- **SERVICIOS WEB:**

1. Desarrollo de Servicios Web en .Net:

<http://es.scribd.com/doc/502746/Sesion-08-Servicios-Web>

[2010/09/18]

2. Protocolos de Servicios Web:

<http://repositorio.utn.edu.ec/bitstream/123456789/624/4/CAPITULO>

[%20V.pdf](#)

[2010/09/19]

- **ADO .NET**

3. Fundamentos de la serialización de .NET Framework

<http://msdn.microsoft.com/es-es/library/ms233836%28v=vs.80%29.aspx>

[2011/06/20]

4. Cómo serializar un objeto a XML mediante Visual Basic .NET o Visual Basic 2005

<http://support.microsoft.com/kb/315703/es>

[2011/06/21]

5. Ejemplos de serialización XML

<http://msdn.microsoft.com/es-es/library/58a18dwa%28v=vs.80%29.aspx>

[2011/06/15]

6. La Clase DataSet

<http://msdn.microsoft.com/es-es/library/system.data.dataset>

[%28v=vs.80%29.aspx](#)

[2011/06/23]

7. Ejemplo de DataSet a XML

<http://support.microsoft.com/kb/309183/es>

[2011/07/02]

8. Asignar los tipos de datos del proveedor de datos para los tipos de datos de .NET Framework

<http://msdn.microsoft.com/es-es/library/4e5xt97a%28v=vs.80%29.aspx>

[2011/07/05]

9. Utilizar XML en un DataSet

<http://msdn.microsoft.com/es-es/library/84sxtbxh%28v=vs.80%29.aspx>

[2011/07/16]

10. Tecnologías de acceso a datos

<http://senaaccesodatos.blogspot.com/2008/04/tecnologia-de-acceso-datos.html>

[2011/07/18]

11. Introducción al modelo independiente del proveedor

<http://msdn.microsoft.com/es-es/library/cy5kfe3c%28v=vs.80%29>

[2011/07/19]

• **REPLICA DE DATOS:**

12. Seleccionar el tipo de réplica apropiado

<http://technet.microsoft.com/es-es/library/ms152565%28SQL.90%29.aspx>

[2011/07/21]

13. Información general sobre los agentes de réplica

<http://technet.microsoft.com/es-es/library/ms152501%28SQL.90%29.aspx>

[2011/07/21]

14. Consideraciones acerca de la réplica de mezcla

<http://msdn.microsoft.com/es-es/library/ms151206%28v=sql.90%29.aspx>

[2011/07/22]