



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
“INTEGRADOR DISTRIBUIDO DE TRANSACCIONES ORIENTADO
A SISTEMAS DE INFORMACIÓN GERENCIAL”

EXAMEN COMPLEXIVO

Previa a la obtención del Título de:
INGENIERO EN CIENCIAS COMPUTACIONALES
ESPECIALIZACIÓN EN SISTEMAS DE INFORMACIÓN / TECNOLOGICOS

Presentado por:
LUIS ALBERTO YCAZA GARCÍA
JAVIER ENRIQUE RIVERA RAMÓN

GUAYAQUIL – ECUADOR

2015

AGRADECIMIENTO

Queremos expresar nuestro agradecimiento

Al Director de Tesis, Ing. Fabricio Echeverría por su generosidad al brindarnos la oportunidad de recurrir a su capacidad y experiencia científica en un marco de confianza, afecto y amistad, fundamentales para la concreción de este trabajo.

A nuestros compañeros de la Facultad de Electricidad y Computación por su continuo y afectuoso aliento.

A nuestras familias por su cariño, comprensión y constante estímulo.

A nuestros padres y hermanos por brindarnos un hogar cálido y enseñarnos que la perseverancia y el esfuerzo son el camino para lograr objetivos.

DEDICATORIA

A Dios.

A nuestras familias.

A nuestros profesores.

TRIBUNAL DE SUSTENTACIÓN

ING. FABRICIO ECHEVERRÍA

Evaluador

ING. LENÍN FREIRE

Evaluador

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este Informe de Proyecto de Graduación, me corresponde exclusivamente; y el patrimonio intelectual de la misma a la **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**”

(Reglamento de exámenes y títulos profesionales de la ESPOL)

LUIS YCAZA GARCÍA

JAVIER RIVERA RAMÓN

RESUMEN

La situación actual de las empresas que han adquirido SIG para el manejo de la relación con sus clientes (distribuidores), es que tienen que mediar con el problema de falta de información actualizada en un repositorio local con formato requerido para ser cargado a SIG desde sus diferentes puntos de venta y terminales de servicio.

Esto impide a los altos ejecutivos tomar decisiones inmediatas a disminuciones en las ventas y explotar las oportunidades de nuevos negocios, debido a que la información transaccional de los distribuidores se encuentra en diferentes plataformas y diversos orígenes de datos.

El presente trabajo muestra una solución tecnológica mediante el uso de servicios web, lo que va a permitir eliminar el problema de acceso a los datos en distancias geográficas y la dificultad de conectividad que existe entre la empresa y los distribuidores. Para esto, el sistema tiene incluido la utilidad para automatizar la captura de las transacciones en línea desde los diferentes puntos de venta, independientemente de la plataforma y la ubicación geográfica donde se encuentre esta información, para su procesamiento y conversión al formato requerido por SIG.

INDICE GENERAL

1. RESUMEN	v
2. INDICE GENERAL	vi
3. REFERENCIA DE ILUSTRACIONES Y TABLAS.....	¡Error! Marcador no definido.
4. INTRODUCCIÓN	xvi
1. CAPÍTULO 1: Antecedentes y Justificación	1
1.1. Antecedentes	1
1.2. Justificación.....	2
1.3. Planteamiento del problema a resolver	3
1.4. Objetivos generales de la solución del problema	6
1.5. Objetivos específicos de la solución del problema	7
2. CAPÍTULO 2: Análisis	8
2.1. Situación actual.....	8
2.2. Problemas de conectividad.....	9
2.2.1. Evolución de las Redes GSM	10
2.3. Problemas de disponibilidad de datos en tiempo real	11
2.4. Problemas de integridad de datos.....	12
2.5. Limitaciones geográficas.....	14
2.6. Problemas con la diversidad de orígenes de datos	14
2.7. Análisis de los orígenes de datos que soportará el IDT	15
2.8. Análisis de las plataformas y dispositivos sobre los que funcionará el IDT.	16
2.9. Comunicaciones	17
3. CAPÍTULO 3: Marco Teórico.....	19
3.1. Definición del Integrador Distribuido de Transacciones (IDT).....	19

3.2.	Alcance funcional del IDT	20
3.2.1.	Administración de Usuarios:	20
3.2.2.	Administración de Clientes	21
3.2.3.	Administración de Distribuidores	21
3.2.4.	Administración de Sistemas de Información	21
3.2.5.	Administración de Transacciones	21
3.2.6.	Administración de Parámetros de Conexión a orígenes de datos	22
3.2.7.	Administración de Formato de salida de la información	22
3.2.8.	Administración de campos del origen de datos	22
3.2.9.	Administración de campos de salida según formato SIG	23
3.2.10.	Extracción de datos del distribuidor	23
3.2.11.	Conversión y Transformación de datos	23
3.3.	Definición de la arquitectura del IDT	24
3.3.1.	Páginas Estáticas	24
3.3.2.	Páginas Dinámicas del lado del Cliente	25
3.3.3.	Páginas Dinámicas del lado del Servidor	26
3.4.	Definición y concepto del protocolo HTTPS	26
3.5.	Definición de Tecnologías aplicadas	28
3.5.1.	Tecnologías para páginas Estáticas	28
3.5.2.	Tecnologías para páginas Dinámicas del lado del cliente:	29
3.5.3.	Tecnologías para páginas Dinámicas del lado del servidor:	31
3.6.	Herramientas de Software	34
3.6.1.	MyEclipse	34
3.6.2.	Microsoft Visual Studio	35

3.6.3.	Microsoft SQL Server	36
3.6.4.	ExtJS	36
3.6.5.	Java JDK.....	37
3.7.	Web Services y su funcionamiento	37
4.	CAPÍTULO 4: Diseño del sistema	40
4.1.	Modelo lógico de datos.....	41
4.2.	Servicios que brinda el IDT.....	43
4.2.1.	Tecnología empleada	43
4.2.2.	Web Services.....	44
4.2.3.	Estándares Empleados	46
4.2.4.	XML	47
4.2.5.	WSDL.....	53
4.2.6.	Descripción de un Webservice	54
4.2.7.	Significado de la descripción de un servicio.....	57
4.2.8.	Interfaces	61
4.2.9.	Objetivo.....	61
4.2.10.	Proceso.....	61
4.3.	Seguridades a implementar en el IDT (Hypertext Transfer ProtocolSecure).....	62
4.4.	Diseño de las pantallas del IDT	62
4.5.	Diseño de pruebas	80
5.	CAPÍTULO 5: Implementación y pruebas	81
5.1.	Selección de las plataformas de desarrollo del sistema	81
	La plataforma de desarrollo ASP – ASPX.....	81
5.2.	Estructura y parametrización de la base de datos.....	89

5.3.	Implementación de los módulos del sistema	103
5.3.1.	Servidor	103
5.3.2.	Cliente PC	104
5.3.3.	Cliente Pocket PC	105
5.3.4.	Administración	106
5.3.5.	Conexión	107
5.3.6.	Seguridad	107
5.3.7.	Conversión	109
5.4.	Pruebas realizadas	111
5.4.1.	Pruebas de Unidad	111
5.4.2.	Pruebas de Integración	111
5.4.3.	Pruebas de Validación	112
5.5.	Análisis y resultados	112
5.6.	Integración de los módulos	113
5.7.	Instalación del sistema	114
6.	CAPÍTULO 6: Definición del producto	116
6.1.	Costos del producto	116
6.1.1.	Costo Total de Implementación (CTI)	116
6.1.2.	Costo Total Administrativo (CTA)	120
6.1.3.	Costo Total de Capacitación (CTC)	123
6.2.	Licenciamiento	124
6.3.	Segmento del Mercado	125
6.4.	Matriz FODA del producto	127
6.5.	Servicios del producto	128

CONCLUSIONES Y RECOMENDACIONES.....	130
ANEXOS	132
BIBLIOGRAFÍA.....	135

INDICE DE FIGURAS

<i>Figura 2.1: Evolución de las tecnologías de radio y de transporte en el RAM (Nakamura, Narvaez, & Ramos, 2009)</i>	11
<i>Figura 2.2: Integridad referencial</i>	13
<i>Figura 3.1: Diseño de Páginas Estáticas</i>	25
<i>Figura 3.2: Diseño de Páginas Dinámicas del lado del cliente</i>	25
<i>Figura 3.3 Diseño de Páginas Dinámicas del lado del Servidor</i>	26
<i>Figura 4.1: Diseño del Sistema</i>	41
<i>Figura 4.2: Modelo Lógico de Datos</i>	42
<i>Figura 4.3: Tecnología Empleada</i>	44
<i>Figura 4.4: Estándares Utilizados</i>	47
<i>Figura 4.5: Tecnología basada en Servicios Web</i>	60
<i>Figura 4.6: Formulario de Acceso al Sistema</i>	63
<i>Figura 4.7: Pantalla Principal del sistema</i>	63
<i>Figura 4.8: Descripción de Pantallas</i>	65
<i>Figura 4.9Mantenimiento Sistemas de Información</i>	66
<i>Figura 4.10: Mantenimiento de Clientes</i>	67
<i>Figura 4.11: Datos Generales del Cliente</i>	68
<i>Figura 4.12: Transacciones</i>	69
<i>Figura 4.13: Ingreso de Transacciones</i>	69
<i>Figura 4.14: Ingreso de Información del Cliente</i>	70
<i>Figura 4.15: Mantenimiento de Distribuidores</i>	71
<i>Figura 4.16: Mantenimiento de Tipos de Clientes</i>	72
<i>Figura 4.17: Mantenimiento de Tipos de Datos</i>	72
<i>Figura 4.18: Mantenimiento de Tipos de Información</i>	73

<i>Figura 4.19: Mantenimiento de Tipos de valor</i>	<i>74</i>
<i>Figura 4.20: Mantenimiento de Formatos</i>	<i>74</i>
<i>Figura 4.21: Mantenimiento de Campo Origen.....</i>	<i>75</i>
<i>Figura 4.22: Mantenimiento de Campo Destino</i>	<i>76</i>
<i>Figura 4.23: Homologación</i>	<i>76</i>
<i>Figura 4.24: Proceso de Match.....</i>	<i>77</i>
<i>Figura 4.25: Transacciones del Sistema</i>	<i>78</i>
<i>Figura 4.26: Transacciones por Cliente</i>	<i>78</i>
<i>Figura 4.27: Procesador de movimientos.....</i>	<i>79</i>
<i>Figura 5.1: Ejemplo de Modularización.....</i>	<i>83</i>
<i>Figura 5.2: Escalabilidad</i>	<i>88</i>

INDICE DE TABLAS

<i>Tabla 1: Soluciones de Negocios Empresariales SAP en Ecuador</i>	2
<i>Tabla 2: Tabla Comparativa de los Sistemas Operativos existentes</i>	4
<i>Tabla 3: Comparación de los Sistemas Administradores de Base de Datos Relacionales</i>	5
<i>Tabla 4: Orígenes de datos.....</i>	16
<i>Tabla 5: Diccionario de Datos.....</i>	90
<i>Tabla 6: Definición SITD_CAMPO_DESTINO</i>	90
<i>Tabla 7: Atributos SITD_CAMPO_DESTINO</i>	90
<i>Tabla 8: Definición SITD_CAMPO_ORIGEN</i>	91
<i>Tabla 9: Atributos SITD_CAMPO_ORIGEN.....</i>	91
<i>Tabla 10: Definición SITD_CLIENTE</i>	91
<i>Tabla 11: Atributos SITD_CLIENTE.....</i>	92
<i>Tabla 12: Definición SITD_FORMATO.....</i>	92
<i>Tabla 13: Atributos SITD_FORMATO.....</i>	93
<i>Tabla 14: Definición SITD_INFO_CLIENTE.....</i>	93
<i>Tabla 15: Atributos SITD_INFO_CLIENTE.....</i>	93
<i>Tabla 16: Definición SITD_MATCH</i>	93
<i>Tabla 17: Atributos SITD_MATCH.....</i>	94
<i>Tabla 18: Definición SITD_MODO_ACCESO.....</i>	94
<i>Tabla 19: Atributos SITD_MODO_ACCESO</i>	94
<i>Tabla 20: Definición SITD OPCIONES.....</i>	95
<i>Tabla 21: Atributos SITD OPCIONES</i>	95
<i>Tabla 22: Definición SITD_SIG</i>	95
<i>Tabla 23: Atributos SITD_SIG</i>	96
<i>Tabla 24: Definición SITD_SIG_CLIENTE</i>	96

<i>Tabla 25: Atributos SITD_SIG_CLIENTE</i>	96
<i>Tabla 26: Definición SITD_TIPO_CLIENTE</i>	96
<i>Tabla 27: Atributos SITD_TIPO_CLIENTE</i>	97
<i>Tabla 28: Definición SITD_TIPO_DATO</i>	97
<i>Tabla 29: Atributos SITD_TIPO_DATO</i>	97
<i>Tabla 30: Definición SITD_TIPO_INFO</i>	98
<i>Tabla 31: Atributos SITD_TIPO_INFO</i>	98
<i>Tabla 32: Definición SITD_TIPO_VALOR</i>	98
<i>Tabla 33: Atributos SITD_TIPO_VALOR</i>	99
<i>Tabla 34: Definición SITD_TPV</i>	99
<i>Tabla 35: Atributos SITD_TPV</i>	99
<i>Tabla 36: Definición SITD_TPV_TRANS_CLIENTE</i>	100
<i>Tabla 37: Atributos SITD_TPV_TRANS_CLIENTE</i>	100
<i>Tabla 38: Definición SITD_TRANSACCION</i>	100
<i>Tabla 39: Atributos SITD_TRANSACCION</i>	101
<i>Tabla 40: Definición SITD_TRANSACCION_X_CLIENTE</i>	101
<i>Tabla 41: Atributos SITD_TRANSACCION_X_CLIENTE</i>	101
<i>Tabla 42: Definición SITD_USUARIO</i>	101
<i>Tabla 43: Atributos SITD_USUARIO</i>	102
<i>Tabla 44: Definición SITD_HOMOLOGACION</i>	102
<i>Tabla 45: Atributos SITD_HOMOLOGACION</i>	102
<i>Tabla 46: Definición SITD OPCIONES_X_USUARIO</i>	103
<i>Tabla 47: Atributos SITD OPCIONES_X_USUARIO</i>	103
<i>Tabla 48: Costos de Desarrollo</i>	118
<i>Tabla 49: Costos de Herramientas Utilizadas</i>	119

<i>Tabla 50: Costos de Instalación, Configuración y Adaptación.....</i>	<i>119</i>
<i>Tabla 51: Matriz FODA.....</i>	<i>127</i>

INTRODUCCIÓN

El primer capítulo indica a quienes va dirigido esta implementación junto con las ventajas que se obtienen del uso de una herramienta tecnológica como ésta, que maneje transacciones distribuidas geográficamente, permitiendo el acceso a información actualizada en el menor tiempo posible y considerando como medio de interacción con el usuario el internet.

El segundo capítulo, nos muestra el análisis del problema inicial que se pretende resolver con la solución propuesta, definiendo para esto, los objetivos del trabajo a realizar y estableciendo el alcance funcional del sistema a desarrollar, junto a toda la arquitectura tecnológica de comunicaciones y seguridad que van a ser requeridos.

El tercer capítulo abarca la descripción del Modelo de Datos seleccionado para este proyecto, la disponibilidad de servicios con los que contará el sistema y una presentación detallada de las pantallas con las que interactuará el usuario.

El cuarto capítulo explica el proceso de implementación y de pruebas del sistema, respectivamente.

El quinto y último capítulo contiene el Análisis de Costos y de FODA del producto o sistema final desarrollado.

CAPÍTULO 1

Antecedentes y Justificación

1.1. Antecedentes

Antes de empezar con la descripción del problema a resolver, se procederá a realizar una breve descripción de la situación en la que se encuentran las empresas que residen en el país y que han adquirido un Sistema de Información Gerencial (SIG)¹ como software de Gestión y Estrategia Empresarial.

Es importante mencionar, que a la fecha existen muchas empresas tanto nacionales como transnacionales que se encuentran como clientes de algún SIG y utilizan la herramienta para el intercambio de

¹SAP es el líder en software empresarial colaborativo.

información con sus distribuidores autorizados y terminales de punto de venta remotos.

Empresa		Sitio Web
Papelesa		www.papelesa.com
Hivimar		www.hivimar.com
Universal Sweet Industries		www.launiversal.com.ec
Deprati		www.deprati.com.ec

Tabla 1: Soluciones de Negocios Empresariales SAP en Ecuador

1.2. Justificación

La justificación para el desarrollo de este proyecto es proporcionar una solución que permita obtener información actualizada, confiable y consistente para ser cargada en CRM – SIG desde un repositorio único, utilizando para esto la extracción de datos de cualquier origen a través de la Web.

Como parte del resultado de este estudio se desarrollará una aplicación donde se podrá administrar los usuarios, clientes y orígenes de datos de manera centralizada por el administrador del proceso, de manera que siempre se pueda tener la disponibilidad de la información.

Esta aplicación facilitará la extracción, transmisión, conversión y la ubicación de las transacciones, de los diferentes puntos remotos de los clientes, en un servidor local de donde serán tomadas por CRM – SIG, para su procesamiento y/o generación de reportes gerenciales.

Cabe mencionar que con esta solución se disminuye en gran manera, la interacción humana con la información de las transacciones diarias, lo que permitirá eliminar los errores producidos por la manipulación de los datos.

1.3. Planteamiento del problema a resolver

La información que requiere el Sistema de Información CRM – SIG para poder generar los reportes que permiten hacer gestión sobre la relación que se mantiene con los clientes, necesita estar localizada en un servidor centralizado, adicionalmente de que debe tener un formato establecido para ser reconocidos por el sistema.

Este tipo de organizaciones que ha implementado SAP como ERP y CRM, en algunos casos trabajan con microempresas denominadas Distribuidores Autorizados o Terminales de Punto de Venta. Los sistemas informáticos de Facturación y Ventas en estos

puntos de intermediación, vamos a encontrar que están desarrollados sobre diversas plataformas de **Sistema Operativo** y **Base de Datos**. (Tabla 2 y 3)

Sistema operativo	Creador	Año de primera distribución
Windows 7	Microsoft	2009
Windows Vista	Microsoft	2007
Windows XP	Microsoft	2001
Windows 2000	Microsoft	2000
Haiku	Haiku Project	2009
Mac OS X	Apple	2001
Mac OS	Apple	1984
Debian GNU/Linux	Proyecto Debian	1993
Fedora(Linux)	Proyecto Fedora	2003
SUSE Linux	SuSE	1994
Mandriva Linux	Mandriva (empresa)	1998
OpenBSD	Theo de Raadt	1996
Solaris	Sun	1992
Plan 9	Bell Labs	1993

Tabla 2: Tabla Comparativa de los Sistemas Operativos existentes(Garcia, 2013)

BASE DE DATOS	Creador	Fecha de la primera versión pública	Última versión estable	<u>Licencia de software</u>
Adaptive Server Anywhere	Sybase/iAnywhere	1992	10	Propietario
Adaptive Server Enterprise	SybaseInc	1987	15	Propietario

ANTs Data Server	ANTs Software	1999	3.6	Propietario
DB2	IBM	1982	9	Propietario
Firebird	FirebirdFoundation	2000	2.1	Licencia Pública InterBase
Informix	Informix Software	1985	10	Propietario
HSQldb	Hsqldb.Org	2001	1.9	Licencia BSD
Ingres	Berkeley University, ComputerAssociates	1980	2006	CA-TOSL
InterBase	Borland	1985	7.5.1	Propietario
SapDB	SAP AG	Desconocido	7.4	GPL con drivers LGPL
MaxDB	MySQL AB, SAP AG	Desconocido	7.7	GPL o propietario
Microsoft SQL Server	Microsoft	1989	2008	Propietario
MySQL	MySQL AB	1996	5	GPL o propietario
Oracle	Oracle Corporation	1977	11g Release 2	Propietario
PostgreSQL	PostgreSQL Global DevelopmentGroup	1989	9	Licencia BSD
SmallSQL	SmallSQL	2005	0.12	LGPL
SQLite	D. Richard Hipp	2000	3.6.16	Dominio público

Tabla 3: Comparación de los Sistemas Administradores de Base de Datos Relacionales

Por lo anteriormente expuesto, vemos que es necesaria la creación de un sistema automatizado y parametrizable que permita extraer la información, sin importar la plataforma de Sistema Operativo o Base de Datos, de tal manera que se pueda concentrar todo el conjunto de datos en un servidor central que permitirá hacer la carga de datos hacia los reportes gerenciales de control y seguimiento.

1.4. Objetivos generales de la solución del problema

- Elaborar un Sistema de Integración Transaccional que permita la comunicación entre los clientes de CRM – SIG y sus diferentes puntos remotos de distribución autorizados, sin utilizar personas como intermediarios y dentro de un ambiente transaccional distribuido.
- Reducir los costos de una organización de con un modelo de negocios en que incluyan puntos remotos de distribución, en cuanto a la contratación de empleados que vayan a realizar físicamente la captación de la información y transformación de la misma en el formato requerido por CRM – SIG.

- Mantener la compatibilidad de la información extraída versus la transformada, refiriéndonos a los tipos de datos equivalentes que serán utilizados en el proceso.

1.5. Objetivos específicos de la solución del problema

- Facilitar a los usuarios de CRM – SIG la extracción de transacciones específicas, según la cadena de valor de las organizaciones, independiente de la plataforma en la que se encuentre el origen de datos.
- Controlar las parametrizaciones de las diversas transacciones definidas en el sistema, para que así el usuario pueda tener el control absoluto de las mismas, obteniendo de esta manera una transmisión de los datos amigable y eficiente.

CAPÍTULO 2

Análisis

2.1. Situación actual

En el Ecuador no se encuentra actualmente ningún partner² de SAP, por tal motivo, las empresas que necesiten algún asesoramiento o soporte lo tienen que realizar con SAP – Colombia que es el más cercano geográficamente hablando. Se puede encontrar un listado de los contactos de SAP en el mundo en <http://www.sap.com/contactsap/countries/index.epx>.

²Socio de negocios.

Un factor a tomar en cuenta es la generación de informes que provee CRM – SAP, y que requieren de información actualizada, en el servidor centralizado. Esto incluye la explotación oportuna de la información de preferencias de los clientes, sectores de mercados no aprovechados, estrategias de Marketing, etc., que ayudan al mejoramiento del servicio que se da al cliente tanto interno como externo.

2.2. Problemas de conectividad

Los problemas de conectividad representan una gran pérdida de productividad para las empresas, por esta razón, se debe tener en consideración lo siguiente:

- En el caso de las aplicaciones stand-alone, el ancho de banda dedicado juega un papel importante para mantener la plataforma en línea, en el caso nuestro, para que no llegue a perderse ninguna transacción enviada hacia el servidor.
- Para el caso de las aplicaciones móviles, el servicio de transmisión de datos contratado con la operadora móvil de preferencia, será la que determine la velocidad de transmisión (GPRS, EDGE, 3G ó 4G), así como también la que permita en términos de cobertura estar conectado desde cualquier parte del territorio nacional.

A continuación un detalle de la evolución de las redes móviles, de solo voz a redes de transmisión de datos de alta velocidad.

2.2.1. Evolución de las Redes GSM

Las redes basadas en GSM³ han evolucionado rápidamente a redes que ya no solamente transmiten voz, sino que se han convertido en verdaderas redes de transmisión de datos de alta velocidad. (JARA, 2004)

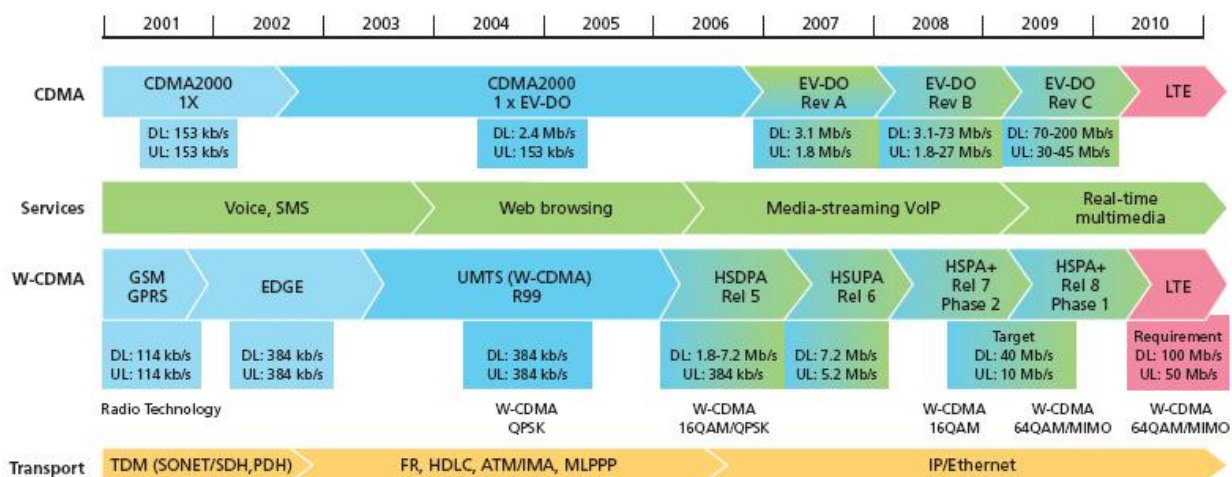
2.2.1.1. GPRS (General Packet Radio Service)

Según sus siglas, GPRS tiene su traducción al español como Servicios Generales de Paquetes por Radio y a menudo es identificada como un protocolo de transmisión de datos “2.5 G”, es decir, una tecnología entre la segunda (2G) y tercera (3G) generación de tecnología móvil digital.

Este protocolo permite realizar videoconferencias, enviar mensajes instantáneos, esté donde esté. GPRS, es la evolución de GSM orientado a paquetes y puede transmitir/enviar datos a una velocidad de hasta 114 Kbps.(Polanco, 2013)(JARA, 2004)

³ GSM: En la actualidad significa Global Systemfor Mobile Commmunications, sin embargo, en sus inicios sus siglas identificaban a un grupo de trabajo encargado de especificar un sistema de comunicaciones móviles común, para Europa en la banda de 900 Mhz (GroupeSpecial Mobile)

Figure 2. Evolution of mobile radio technologies and associated RAN transport



2.2.1.2. EDGE (Enhanced Data rates for GSM Evolution)

Esta tecnología es conocida también como EGPRS (Enhanced GPRS), pues actúa como puente entre las redes 2G y 3G. Es considerada como una evolución de GPRS. (Montero, 2013)

Figura 0.1: Evolución de las tecnologías de radio y de transporte en el RAM (Nakamura, Narvaez, & Ramos, 2009)

2.3. Problemas de disponibilidad de datos en tiempo real

La disponibilidad de datos en tiempo real, permite a las empresas tomar decisiones y esto se vuelve un problema cuando la información se captura en orígenes de datos dispersos donde no se cuenta con un enlace de datos para el envío de los datos.

El uso de Servicios Web permite la comunicación entre el servidor y clientes de captura de la información, mediante el uso de las redes disponibles de celulares en cualquier punto geográfico donde se cuente con cobertura.

El IDT, permitirá la comunicación vía protocolo seguro HTTPS⁴), con los clientes donde se encuentre la información a transmitir, por medio de tecnología basada en Servicios Web. Esto nos proveerá de una interconexión con el Servidor para que la información esté disponible en línea en el mismo momento de su captura y transmisión.

2.4. Problemas de integridad de datos

El término integridad de datos se refiere a la corrección y completitud de los datos en una base de datos. Cuando los contenidos se modifican con sentencias INSERT, DELETE o UPDATE, la integridad de los datos almacenados puede perderse de muchas maneras diferentes. Pueden añadirse datos no válidos a la base de datos, tales como un pedido que especifica un producto no existente.

⁴Hyper Text Transfer ProtocolSecure (en español: 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, es decir, es la versión segura de HTTP.

Pueden modificarse datos existentes tomando un valor incorrecto, como por ejemplo si se reasigna un vendedor a una oficina no existente. Los cambios en la base de datos pueden perderse debido a un error del sistema o a un fallo en el suministro de energía. Los cambios pueden ser aplicados parcialmente, como por ejemplo si se añade un pedido de un producto sin ajustar la cantidad disponible para vender.

El IDT, implementa integridad a nivel del motor de base de datos SQL Server 2005, para que guarde la debida integridad referencial entre las diferentes entidades y objetos de la estructura de Base de datos.

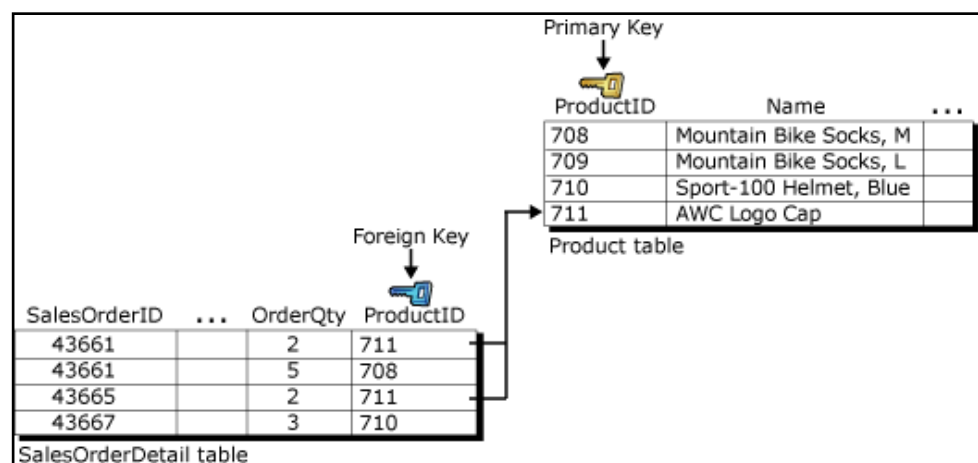


Figura 0.2: Integridad referencial

2.5. Limitaciones geográficas

Los sistemas distribuidos actualmente tienen las limitaciones de acceso a redes de Internet y de datos, según el punto geográfico donde se encuentren. Los enlaces de última milla de los proveedores de redes de transmisión de datos y los operadores celulares tienen una cobertura que no alcanza a todo el territorio nacional e internacional.

Para resolver este problema, el IDT, utiliza dentro de sus terminales una base de datos local, que permitirá la retransmisión de los datos una vez que se tenga red de datos disponible.

La retransmisión de datos será realizada como una sincronización manual, la cual la realizará el usuario cuando detecte cobertura idónea de comunicación desde el terminal al servidor del IDT.

Con esta operación se mantiene la integridad de la información que se encuentra actualizada en el servidor.

2.6. Problemas con la diversidad de orígenes de datos

Actualmente, tenemos a nuestra disposición de una cantidad de opciones para la administración de nuestra información, debido a la

diversidad de motores de bases de datos existentes, cada una con características agregadas sobre las demás.

También, tenemos diversidad en lo que respecta a lo que son dispositivos móviles de almacenamiento y procesamiento (PDAs, Pocket PCs, PDT's, Tablets, etc), los cuales pueden almacenar una cantidad considerable de información y transmitirla por medio de sincronización local o remota.

Por lo antes expuesto, el IDT cuenta con la posibilidad de realizar sincronización y extracción de información mediante Servicios WEB, que pueden ser consumidos desde cualquier aplicativo cliente instalado en un dispositivo móvil, sobre cualquier plataforma. Así mismo, el IDT puede ser configurado para poder conectarse a los distintos motores de bases de datos, con lo cual la integración de cualquier tipo de transacción en el origen que este sea está asegurada.

2.7. Análisis de los orígenes de datos que soportará el IDT

El IDT utiliza la Conectividad abierta de bases de datos (ODBC, Open DatabaseConnectivity) de Java, más conocida como

“JDBC”⁵. Esto permite que el sistema pueda conectarse a los orígenes de datos compatibles con este tipo de tecnología(Java - Oracle, 2014)(DEITEL & DEITEL, 2008). Entre los orígenes de datos soportados se encuentran:

- Bases de datos
- Hojas de cálculo
- Archivos de datos secuenciales; o
- Directorios de correo electrónico.

En la siguiente tabla encontraremos algunos de los orígenes de datos a los cuales puede acceder el IDT:

Microsoft Access
SQL Server
Oracle

Tabla 4: Orígenes de datos

2.8. Análisis de las plataformas y dispositivos sobre los que funcionará el IDT.

El IDT es un sistema desarrollado en su parte medular sobre plataforma Java, lo cual permite que funcione sin problema sobre los distintos sistemas operativos: Windows, GNU/Linux, Mac OS X.

⁵Java DatabaseConnectivity, más conocida por sus siglas JDBC1 2, es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

Adicionalmente se encuentra basado en tecnología Web 2.0⁶, que respeta los estándares en cuanto a el diseño y publicación de servicios web, lo que hace que pueda ejecutarse correctamente sobre la mayoría de los navegadores de internet, tales como: Internet Explorer, Mozilla Firefox, Opera, Safari, entre otros.

Entre los dispositivos sobre los cuales puede ejecutarse el sistema, se encuentran los smartphones, tabletsy todos aquellos equipos móviles con capacidad de conectarse a internet. Esto incluye el soporte sobre plataformas Windows Mobile – Compact Edition, Android, Symbian y iOS.

2.9. Comunicaciones

El sistema IDT funciona completamente a través del canal de información abierto como es el internet por medio de protocolo HTTPS. Esto es, cada punto de distribución intercambia datos con el servidor central de integración mediante una vía segura, mediante un agente que previamente es instalado en los terminales.

Un enlace de banda ancha, de preferencia, es requerido para mantener enlazados agentes-servidor continuamente y así evitar la

⁶El término Web 2.0 comprende aquellos sitios web que facilitan el compartir información, la interoperabilidad, el diseño centrado en el usuario¹ y la colaboración en la World Wide Web.

pérdida de datos por cortes o pérdidas de paquetes en el medio.(Barreno, 2010)(Herrera, 2011)

CAPÍTULO 3

Marco Teórico

3.1. Definición del Integrador Distribuido de Transacciones (IDT).

Un Sistema de Integración Transaccional es un coordinador de transacciones distribuidas entre sistemas heterogéneos, que permite gestionar la integración de distintas aplicaciones agregando una capa de procesos y validación middleware, término utilizado para referirse a los componentes de software que funcionan como intermediarios entre otros componentes y que generalmente se encuentran en el contexto de la interacción cliente/servidor (Elissalde, 2014), al intercambio de información entre los clientes de CRM – SIG con sus distintos puntos remotos autorizados para la distribución.

El IDT contempla un conjunto de aplicativos e infraestructura que permite soportar toda la exigencia y eficiencia en la integración de las distintas plataformas, entre los clientes de un producto o servicio y los que ofrecen el mismo, soportando para esto mecanismos mínimos de seguridad de la información transportada junto con la continuidad (disponibilidad) de los servicios a transacciones ofrecidos.

La extracción y conversión de las transacciones en el formato requerido es realizado en un determinado instante, definido por cada usuario, conforme al modelo de negocio y necesidad del cliente que contrata el servicio. (JARA, 2004),

3.2. Alcance funcional del IDT

En la aplicación demostrativa del Proyecto de Graduación, el Sistema de Integración Transaccional Distribuido (IDT) es un portal transaccional que cuenta con las siguientes funcionalidades:

3.2.1. Administración de Usuarios:El administrador del sistema podrá consultar, registrar, eliminar y modificar datos de los usuarios con acceso a la aplicación.

- 3.2.2. Administración de Clientes:** El administrador podrá consultar, ingresar, eliminar y modificar la información de los clientes que utilizan los servicios que provee el IDT. Así mismo, dentro de esta funcionalidad se podrá administrar los datos de homologación de campos.
- 3.2.3. Administración de Distribuidores:** Los clientes podrán consultar, ingresar, eliminar y modificar la información de sus distribuidores autorizados con quienes van a establecer una conexión, para la extracción de los datos de transacciones definidas. Dentro de esta funcionalidad se incluye la facilidad de establecer los parámetros de conexión hacia los distintos orígenes de datos.
- 3.2.4. Administración de Sistemas de Información:** Los clientes pueden consultar, ingresar, eliminar y modificar las descripciones de los Sistemas de Información a los cuales van a alimentar por medio del IDT. Dentro de esta funcionalidad se incluye la facilidad de poder administrar el formato de los campos que requiere SIG, para que sea colocado en el repositorio central.
- 3.2.5. Administración de Transacciones:** Los clientes pueden consultar, ingresar, eliminar y modificar las transacciones que mantienen integradas con los respectivos

distribuidores autorizados. Cabe indicar que una transacción guarda la relación entre cliente, distribuidor, sistema de información.

3.2.6. Administración de Parámetros de Conexión a orígenes

de datos: Los administradores y clientes podrán consultar, ingresar, eliminar y modificar los parámetros generales de conexión para acceder a cualquier origen de datos. Estos parámetros permitirán acceder a la información en las distintas plataformas que operen los diferentes distribuidores.

3.2.7. Administración de Formato de salida de la

información: Los clientes pueden consultar, ingresar, eliminar y modificar los formatos de campos con los cuales deben ser retornados por el Sistema de Información, en nuestro caso algún SIG. En esta opción, se establece el tipo de dato además de la máscara, los cuales serán utilizados para la salida del archivo final.

3.2.8. Administración de campos del origen de datos:

Los clientes y administradores pueden consultar, ingresar, eliminar y modificar los campos de origen que van a ser extraídos en cada distribuidor del cliente. Cabe indicar que cada campo origen tendrá un formato y pertenecerá a un

distribuidor, cliente, transacción y si es el caso, consulta SQL para el motor de base de datos correspondiente.

3.2.9. Administración de campos de salida según formato

SIG: Los clientes pueden consultar, ingresar, eliminar y modificar los campos de salida en el archivo con formato del Sistema de Información SIG. En esta opción se puede establecer el tipo de dato, longitud y máscara de salida de cada campo a transformar.

3.2.10. Extracción de datos del distribuidor:

Los distribuidores pueden realizar la extracción y envío de la información de las transacciones parametrizadas, hasta el servidor web donde se realizará la conversión y envío de los datos hasta el cliente. Esta operación puede ser realizada de manera manual o automática, lo cual es programado de lado del distribuidor, ya que se instala un aplicativo agente que consume los servicios de extracción y envío publicados en la web.

3.2.11. Conversión y Transformación de datos:

El administrador podrá realizar de manera manual o automática la conversión y transformación de los datos ya extraídos y transmitidos desde los distribuidores hasta el servidor. Cabe indicar que este proceso será en tipo batch donde se

generarán todos los archivos formateados y serán ubicados en los servidores FTP de los clientes.

3.3. Definición de la arquitectura del IDT

El proyecto de graduación incluye el análisis de algunos aspectos importantes como son: el manejo de tecnologías web, diseño de interfaces amigables del lado del cliente, desarrollo de servicios de conversión y autenticación a publicar en la web y conexión a orígenes de datos diversos.

En cuanto a las tecnologías web se refiere, tenemos que estas se clasifican en tres grandes grupos que son:

- Tecnologías para páginas estáticas
- Tecnologías para páginas dinámicas del lado del cliente
- Tecnologías para páginas dinámicas del lado del servidor

Para la mejor comprensión de estas tecnologías, vamos a realizar algunas definiciones de las mismas:

3.3.1. Páginas Estáticas: Son aquellas que se almacenan en el servidor en un archivo con extensión htm o html. No pueden ser personalizadas (Vázquez, CREACIÓN DE SITIOS WEB, 2006.), como se muestra en la siguiente ilustración:

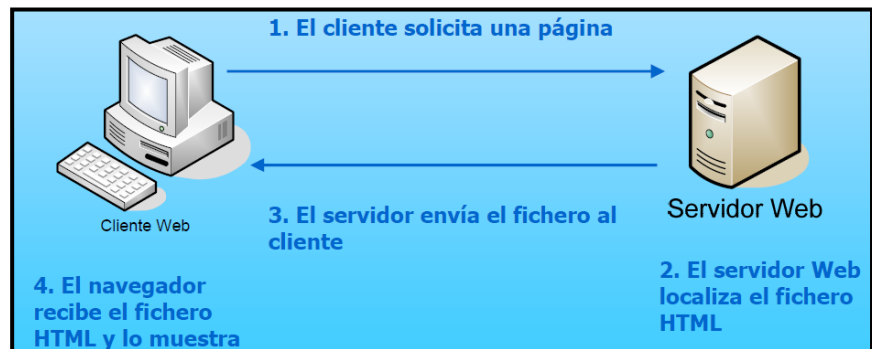


Figura 0.1: Diseño de Páginas Estáticas

3.3.2. Páginas Dinámicas del lado del Cliente: Son aquellas en las que la propia página contiene implementado código para la interacción con el usuario. Por esta razón, se suele decir que dicha interactividad se realiza del lado del (PC) cliente.

Se implementa con Lenguajes de Script (Ver sección 3.5.1), como se muestra en la siguiente ilustración:

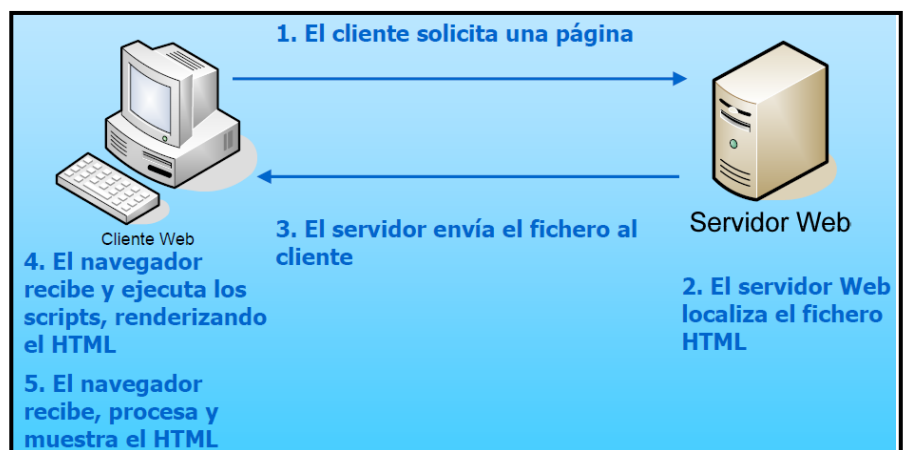


Figura 0.2: Diseño de Páginas Dinámicas del lado del cliente

3.3.3. Páginas Dinámicas del lado del Servidor: Son aquellas que son generadas por una aplicación web, de tal manera que la información contenida en ellas puede haber sido personalizada por medio del envío de parámetros de parte del usuario. La interactividad se realiza del lado del servidor.

Se implementa con diversas tecnologías, como se indica en la siguiente ilustración:

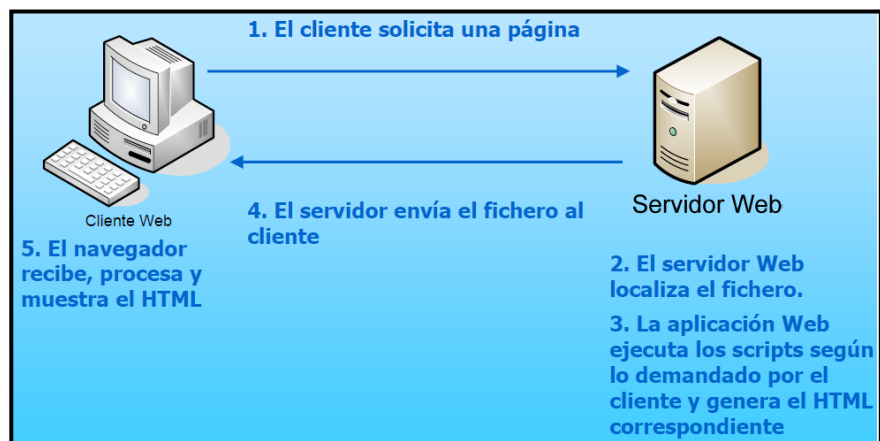


Figura 0.3 Diseño de Páginas Dinámicas del lado del Servidor

3.4. Definición y concepto del protocolo HTTPS

Más conocido por sus siglas HTTPS, es un protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto, es decir, es la versión segura de HTTP.

Es utilizado principalmente por entidades bancarias, tiendas en línea, y cualquier tipo de servicio que requiera el envío de datos personales o contraseñas.

La idea principal de https es la de crear un canal seguro sobre una red insegura. Esto proporciona una protección razonable contra ataques eavesdropping y man-in-the-middle, siempre que se empleen métodos de cifrado adecuados y que el certificado del servidor sea verificado y resulte de confianza.

La confianza inherente en HTTPS está basada en una Autoridad de certificación superior que viene preinstalada en el software del navegador (Es el equivalente a decir "Confío en la autoridad de certificación (p.e. VeriSign/Microsoft/etc.) para decirme en quien debería confiar"). Sin embargo una conexión HTTPS a un website puede ser validada si y solo si todo lo siguiente es verdad:

1. El usuario confía en la Autoridad de certificación para dar fe solo para websites legítimos sin nombres engañosos.
2. El website proporciona un certificado válido (y un certificado inválido muestra una alerta en la mayoría de los navegadores), lo que significa que está firmado por una autoridad confiable.

3. El certificado identifica correctamente al website (p.e. visitando `https://algunsitio` y recibiendo un certificado para "AlgúnSitio S.A." y no "AlgúnZitio S.A.").
4. Cada uno de los nodos involucrados en internet son dignos de confianza, o que el usuario confíe en que la capa de cifrado del protocolo (TLS o SSL) es inquebrantable por un eavesdropper.

HTTP es inseguro y está sujeto a ataques man-in-the-middle y eavesdropping que pueden permitir al atacante obtener acceso a cuentas de un sitio web e información confidencial. HTTPS está diseñado para resistir esos ataques y ser menos inseguro. (Barreno, 2010) (Eivind, 2008)

3.5. Definición de Tecnologías aplicadas

3.5.1. Tecnologías para páginas Estáticas:

- 3.5.1.1. HTML:** Lenguaje de marcado que está basado en etiquetas las cuales representan ciertos elementos. Permiten mostrar el texto. (Vázquez, 2006) (Eguiluz, Introducción a XHTML)

3.5.1.2. CSS: Hojas de estilo que permiten maquetar a las páginas HTML. (Eguiluz)

3.5.2. Tecnologías para páginas Dinámicas del lado del cliente:

3.5.2.1. Lenguajes de scripts

- **VBScript:** Lenguaje basado en Visual Basic, competidor de JavaScript. Es interpretado solamente en navegadores de Microsoft.
- **JavaScript:** Lenguaje derivado de LiveScript de Netscape. Basado en la familia de los lenguajes C. Tienen muchos elementos de Java pero no es Java propiamente. (Eguiluz)

3.5.2.2. Aplicaciones para ejecución local

- **Java Applets:** Aplicación gráfica Java que se queda embebida en una página Web. Necesita de una Máquina Virtual de Java, por tanto, es multiplataforma. (DEITEL & DEITEL, 2008)
- **Active X Controls:** Aplicación realizada en VB o en C++, basada en tecnologías Microsoft que se embebe en HTML. Esta tecnología está en desuso, ya que sólo funcionan sobre el navegador Internet Explorer.

- **Animaciones Flash:** Objetos realizados en Adobe Flash y embebidos en la página Web. (Vázquez, 2006.)
- **AJAX:** El término AJAX es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "JavaScript asíncrono + XML".

“Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes” (James , 2005)

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.(Eguiluz, 2007)

3.5.3. Tecnologías para páginas Dinámicas del lado del servidor:

Ejecutan programas o aplicaciones en el servidor que generan dinámicamente como resultado código HTML. Esto permite que el navegador sea un cliente neutro. Es posible la ejecución distribuida, accediendo a distintos recursos distribuidos como bases de datos.

Estas tecnologías se dividen en 2:

3.5.3.1. Independiente de la arquitectura de la página.

CGI: Tecnología usada durante mucho tiempo en los servidores que adolecían de problemas de rendimientos (lanzaba una instancia de la aplicación por cada petición del cliente,

independiente del servidor web). Aplicaciones escritas en C, C++ o Perl para un tratamiento adecuado de las cadenas de caracteres. Esta tecnología es muy difundida y utilizada a pesar de la complejidad para aprenderla.

3.5.3.2. Dependiente de la Arquitectura

3.5.3.2.1. Active Server Pages (ASP):

Construida usando VBScript o JavaScript. Acceden a los mismos servicios que una aplicación Windows de escritorio, incluyendo ADO, SMTP y COM. Son scripts interpretados cada vez que son solicitados, razón por la cual son lentas.

Utilizan como servidor web el Internet Information Server (IIS) y otros servidores con addons⁷.

3.5.3.2.2. Java Server Pages:

Son como las página ASP, pero implementadas en Java. Se destaca el concepto de Servlet⁸. Es la plataforma más difundida y actualizada, aunque adolece una serie de problemas. Pueden usar EJBs⁹ (propietario) y Servicios Web.

⁷Un **add-on** es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. También se lo conoce como plug-in(del inglés “enchufable”), complemento o extensión.

⁸Los **servlets**, son objetos que corren dentro del contexto de un contenedor de servlets (ej: Tomcat) y extienden su funcionalidad.

⁹Los EJB proporcionan un modelo de componentes distribuido estándar del lado del servidor.

Los servidores utilizados para este tipo de páginas son: Tomcat, Apache, JBoss y derivados además del IIS.(DEITEL & DEITEL, 2008)

3.5.3.2.3. PHP HypertextPreprocessor:Tecnología similar a ASP, pero usando C y Perl. Es código libre y funciona sobre servidores como Apache y derivados, IIS.(DEITEL & DEITEL, 2008)

3.5.3.2.4. ASP.NET: Evolución de la tecnología ASP que permite usar cualquiera de los lenguajes .NET (VB.NET, C#, C++, etc.). Resuelve muchos de los problemas de rendimiento de ASP al ser compilado. Permite utilizar Servicios Web.

El servidor que utiliza esta tecnología es IIS.

En lo que respecta al desarrollo de este proyecto de graduación, se ha escogido una combinación de las tecnologías de diseño de páginas web dinámicas, pues en lo que respecta a la capa de presentación se ha desarrollado sobre una plataforma que se ejecuta del lado del cliente y está basada en JavaScript, como lo es EXTJS¹⁰. Mientras que para la comunicación con el servidor de aplicaciones y base de datos, se eligió el

¹⁰Ext JS es una librería Javascript ligera y de alto rendimiento, compatible con la mayoría de navegadores para crear páginas web y aplicaciones dinámicas.

desarrollo sobre páginas dinámicas JSP dependientes de la arquitectura, Servlets, EJB y AJAX ejecutado desde JavaScript.

Por otro lado, en lo que respecta a la publicación de los servicios de conversión con los que cuenta el IDT, se escogió la tecnología de Servicios Web¹¹ disponibles con autenticación.

Finalmente, en cuanto a la conexión a orígenes de datos diversos se ha optado por la carga dinámica de controladores JDBC¹², pues este API¹³ permite la conexión a la mayoría de base de datos y contenedores de información.(Giardina, 2011)

3.6. Herramientas de Software

3.6.1. MyEclipse

MyEclipse es el más completo IDE (Ambiente de desarrollo Integrado) Java EE/J2EE para la plataforma open

¹¹Un 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.

¹²Java DatabaseConnectivity, más conocida por sus siglas JDBC, es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

¹³Una interfaz de programación de aplicaciones o API (del inglés ApplicationProgramming Interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

sourceEclipse. Esta herramienta no es más que un plugin para la aplicación Eclipse, es decir, es un añadido a este programa que nos facilita el desarrollo de aplicaciones en Java, JSP, JSF, Struts, y otros tantos lenguajes de programación.

(My Eclipse, 2014)

3.6.2. Microsoft Visual Studio

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta múltiples lenguajes de programación tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby, PHP; al igual que entornos de desarrollo web como ASP.NET MVC, Django, etc, a lo cual sumarle las nuevas capacidades online bajo Windows Azure en forma del editor Monaco.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos

embebidos, consolas (la xbox 360 y xboxone), etc.(Visual Studio, 2014)

3.6.3. Microsoft SQL Server

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, PostgreSQL o MySQL.(Microsoft, 2014)

3.6.4. ExtJS

ExtJS es una biblioteca de JavaScript para el desarrollo de aplicaciones web interactivas usando tecnologías como AJAX, DHTML y DOM. Fue desarrollada por Sencha.

Originalmente construida como una extensión de la biblioteca YUI por Jack Slocum, en la actualidad puede usarse como extensión para la biblioteca jQuery y Prototype. Desde la versión 1.1 puede ejecutarse como una aplicación independiente.(Sencha, 2014)

3.6.5. Java JDK

Java Development Kit o (JDK), es un software que provee herramientas de desarrollo para la creación de programas en Java. Puede instalarse en una computadora local o en una unidad de red.

En la unidad de red se pueden tener las herramientas distribuidas en varias computadoras y trabajar como una sola aplicación.

El JDK de Java es un software multiplataforma que puede ser instalado en varios sistemas operativos y poder crear aplicaciones Java tanto Web como escritorio debido a la multitud de librerías (.jar) que se encuentran disponibles.(DEITEL & DEITEL, 2008, págs. 3,10)

3.7. WebServices y su funcionamiento

Un Web Servicees una colección de protocolos y estándares que sirve para intercambiar datos entre aplicaciones.

Distintas aplicaciones de software desarrolladas en lenguajes de programación diferente y ejecutada sobre cualquier plataforma pueden utilizar Web

Services para intercambiar mensajes (servicios) en redes como Internet.

La interoperabilidad se consigue mediante la adopción de estándares abiertos. OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los Web Services.

Los Web Services, garantizando el uso de estándares en la tecnología subyacente utilizada, permiten el gran crecimiento del concepto SOA (Arquitectura Orientada a Servicios) en la actualidad. Ya que los Web Services permiten independizarse de la plataforma y de los lenguajes de programación a ser utilizados.

Los Web Services, no son por lo tanto aplicaciones con una interfaz gráfica con la que las personas puedan interactuar, sino que son un conjunto de especificaciones de software accesible en Internet (o en redes privadas que usen tecnologías Internet) por otras aplicaciones. De esta forma podemos desarrollar aplicaciones que hagan uso de otras aplicaciones que estén disponibles en Internet.

Un ejemplo común podría ser un Web Service al que se le pudiese preguntar por una empresa y que nos retornase en

tiempo real el valor al que están cotizando las acciones de dicha compañía.

De esta forma cualquier aplicación (ya sea Web o de escritorio) que quiera mostrar esta información sólo tendría que solicitarla a través de Internet al Web Service cuando la necesitase.(Besteiro & Rodriguez)

CAPÍTULO 4

Diseño del Sistema

El SISTEMA DE INTEGRACIÓN TRANSACCIONAL DISTRIBUIDO (IDT) está diseñado para la conectividad de dispositivos que pueden estar localizados en distintos puntos geográficos, y procesar sus transacciones para que sus datos sean compatibles a formato de SIG.

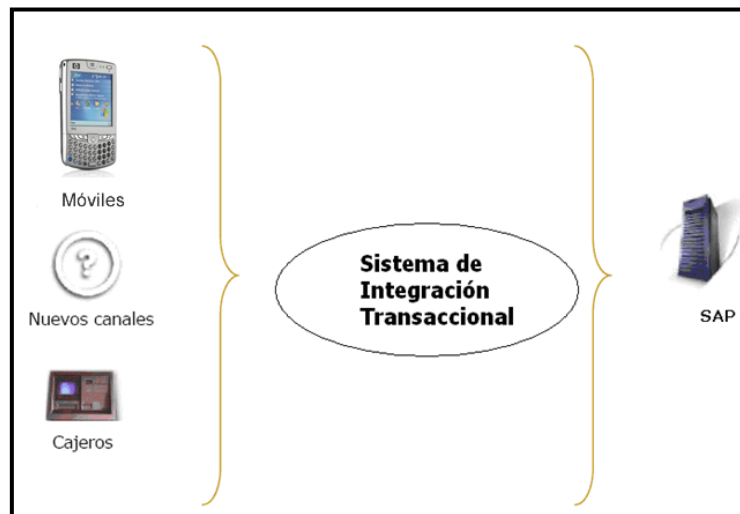


Figura 0.1: Diseño del Sistema

Para esto necesitamos de los siguientes puntos que se describen a continuación:

4.1. Modelo lógico de datos

Para la definición del modelo lógico de nuestro proyecto de graduación, hemos optado por el motor de base de datos SQL Server 2005 ya que por sus características y rendimiento, hacen de éste motor una muy buena alternativa para este tipo de sistemas.

Como podemos notar, el IDT consta de tablas y relaciones suficientes para ser un sistema de conversión y transformación de datos que ayuden a SIG a procesarlos y posteriormente llegar a ser información útil tanto para auditorías como para la toma de decisiones dentro de la empresa.

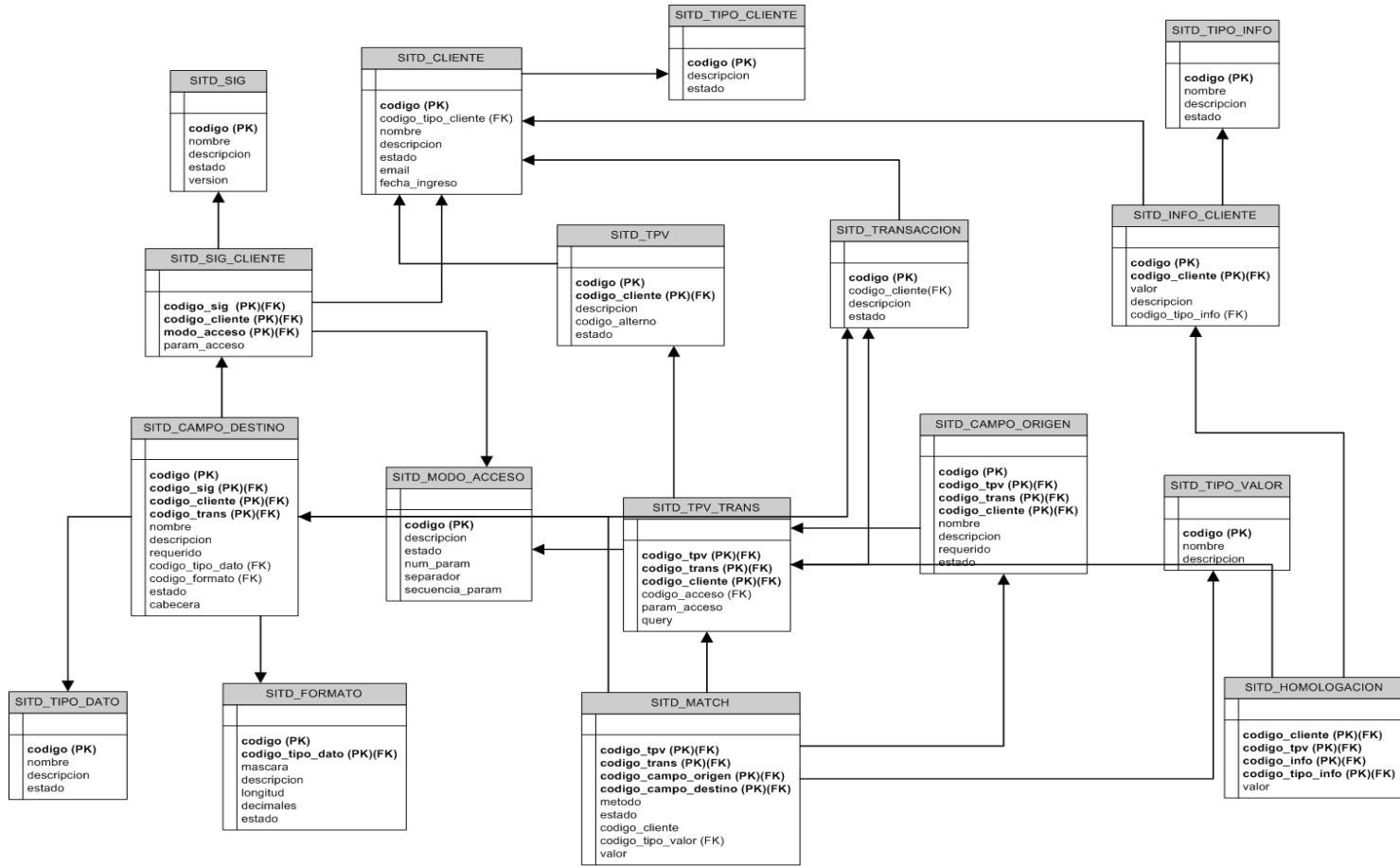


Figura 0.2: Modelo Lógico de Datos

4.2. Servicios que brinda el IDT

Debido a la interoperabilidad que tendrá el sistema, la misma que se debe a la administración y el procesamiento de las distintas transacciones en un ambiente distribuido, es necesario disponer de soluciones que permitan integrar a todas las partes, proveyendo de métodos y mecanismos seguros para el transporte de dicha información

4.2.1. Tecnología empleada

La tecnología por sí misma no podría modificar absolutamente nada dentro de la sociedad de la información. Esta es solamente el soporte, la infraestructura necesaria para que las comunidades que la utilizan construyan a partir de ella.

En este sentido las comunidades internacionales están dando ejemplo que la construcción cooperativa de sistemas es posible.

Claramente el ejemplo más contundente lo ha dado la misma comunidad que desarrolla y promueve los estándares mencionados anteriormente.

Otro ejemplo contundente y además basado en el concepto de componentes distribuidos lo ha dado la comunidad Java a través de la especificación JEE (Java Enterprise Edition) y sin duda el esfuerzo mayor lo han dado las mismas Empresas de tecnología mediante la voluntad de crear en Internet estándares que permitan la integración de cualquier sistema independientemente de su plataforma mediante la utilización de XML y Web Services.

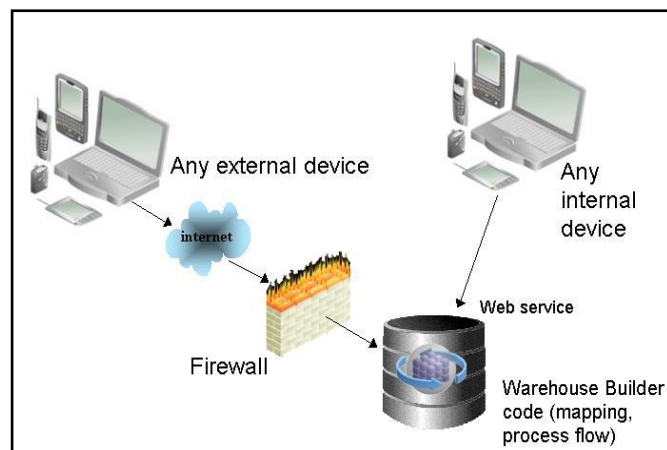


Figura 0.3: Tecnología Empleada

4.2.2. Web Services

Un WebServicees una colección de protocolos y estándares que sirve para intercambiar datos entre aplicaciones.

Distintas aplicaciones de software desarrolladas en lenguajes de programación diferente y ejecutada sobre cualquier plataforma pueden utilizar Web Servicespara

intercambiar mensajes (servicios) en redes como Internet.

La interoperabilidad se consigue mediante la adopción de estándares abiertos. OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los Web Services.

Los Web Services, garantizando el uso de estándares en la tecnología subyacente utilizada, permiten el gran crecimiento del concepto SOA (Arquitectura Orientada a Servicios) en la actualidad. Ya que los Web Services permiten independizarse de la plataforma y de los lenguajes de programación a ser utilizados.

Los Web Services, no son por lo tanto aplicaciones con una interfaz gráfica con la que las personas puedan interactuar, sino que son un conjunto de especificaciones de software accesible en Internet (o en redes privadas que usen tecnologías Internet) por otras aplicaciones. De esta forma podemos desarrollar aplicaciones que hagan uso de otras aplicaciones que estén disponibles en Internet.

Un ejemplo común podría ser un Web Service al que se le pudiese preguntar por una empresa y que nos retornase en tiempo real el valor al que están cotizando las acciones de dicha compañía.

De esta forma cualquier aplicación (ya sea Web o de escritorio) que quiera mostrar esta información sólo tendría que solicitarla a través de Internet al Web Service cuando la necesitase.(IBM, 2011)

4.2.3. Estándares Empleados

- Web ServicesProtocolStack: Así se denomina al conjunto de servicios y protocolos de los servicios Web.
- XML (eXtensibleMarkupLanguage): Es el formato estándar para los datos que se vayan a intercambiar.
- Otros protocolos: los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos normales como HTTP, FTP, o SMTP.
- WSDL (Web ServiceDefinitionLanguage): Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web.

- UDDI (Universal Description, Discovery, and Integration): Protocolo para publicar la información de los servicios Web. Permite a las aplicaciones comprobar qué servicios web están disponibles.
- WS-Security: Protocolo de seguridad aceptado como estándar por OASIS. Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados.

(Galante, 2009)

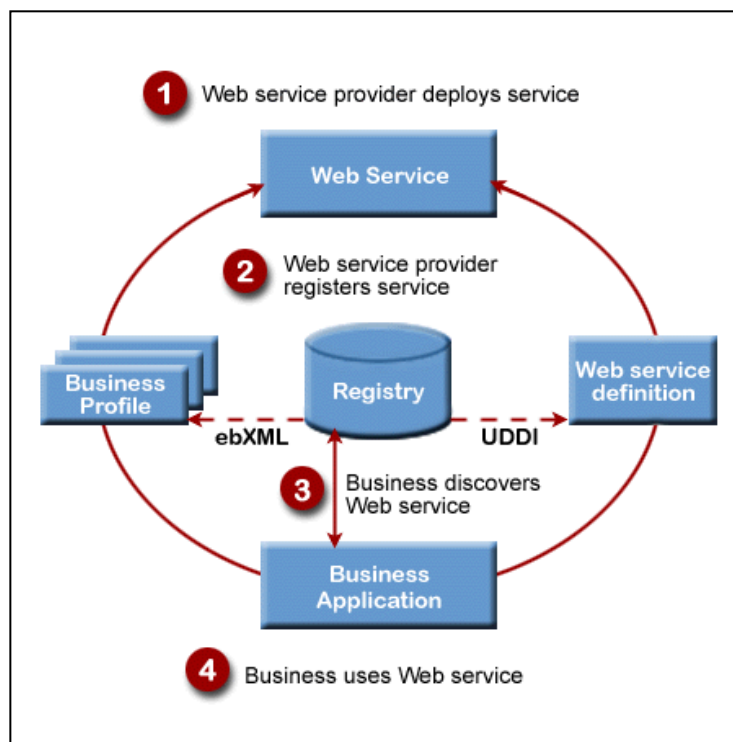


Figura 0.4: Estándares Utilizados

4.2.4. XML

XML (Extensible MarkupLanguage) es un lenguaje extensible de etiquetas, desarrollado por el World Wide Web Consortium (W3C).

Se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Permite compartir la información de una manera segura, fiable y sencilla.(W3C, 2004)

Es un subconjunto de SGML (Standard GeneralizedMarkupLanguage, lenguaje de marcado generalizado estándar), y su objetivo es permitir el uso de SGML en Internet, de modo que un documento escrito en XML sea servido, recibido y procesado de la misma manera que un documento HTML. Si bien es una forma restringida de SGML, preserva una buena parte de su potencial y riqueza, e incluye sus características más utilizadas. Fue diseñado para que fuera fácil de implementar, y para interoperar tanto con SGML como con HTML.

El SGML es un lenguaje para describir lenguajes de marcado, particularmente aquellos utilizados en intercambio, manejo y publicación de documentos

electrónicos. Surgió a mediados de los '80 y se ha mantenido bastante estable.

Gran parte de su estabilidad se debe a que es un lenguaje rico y flexible. Esta flexibilidad, sin embargo, hace que tenga un alto nivel de complejidad, lo cual impide que pueda utilizarse en varios entornos, incluyendo la Web.

Este lenguaje permite definir la gramática de lenguajes específicos para diferentes necesidades.

Los objetivos al diseñar XML fueron:

1. Que fuera directamente utilizable en Internet.
2. Que soportara una amplia variedad de aplicaciones.
3. Que fuera compatible con SGML.
4. Que fuera sencillo escribir programas que procesaran documentos XML.
5. Que la cantidad de características optativas fuera mínima, en lo posible cero.
6. Los documentos XML debían ser legibles a nivel humano y razonablemente claro.
7. El diseño de XML debía prepararse rápidamente, ser formal y conciso.

8. Los documentos XML debían ser fáciles de crear.

9. Lograr etiquetas concisas sería lo menos importante.(Álvarez, 2001)

4.2.4.1. Ventajas

- Comunicación de datos. Permite la transferencia de información en texto plano, con formato XML, entre aplicaciones diferentes.
- Migración de datos. Facilita un lenguaje intermedio para escribir los datos de una aplicación a otra diferente, por ejemplo para intercambio entre diferentes sistemas de bases de datos.
- Aplicaciones Web. Una sola aplicación maneja los datos y cada navegador aplica el estilo adecuado para mostrarlos.

HTML fue pensado como un lenguaje de publicación e intercambio de documentos técnicos y científicos. Resolvía el problema de la complejidad de SGML especificando un pequeño conjunto de etiquetas para edición de documentos relativamente simples. Además

de simplificar la estructura de los documentos, HTML agregó soporte para hipertexto.

Más tarde se agregarían capacidades multimedia. En un breve lapso de tiempo, HTML se hizo muy popular, y rápidamente superó su propósito original. Desde su creación, se han agregado nuevos elementos para incluir en el estándar, y así adaptarlo a mercados más especializados. Esta gran variedad de nuevos elementos trajo problemas para operar con documentos entre diferentes plataformas.

En XML es relativamente sencillo introducir nuevos elementos o atributos de elementos. La familia XHTML fue diseñada para acomodar esas extensiones por medio de módulos XHTML y técnicas para desarrollar nuevos módulos compatibles con XHTML.

Tales módulos permitirán la combinación de nuevas características junto con las existentes. Constantemente se introducen nuevas formas de acceso a Internet. La

familia XHTML fue diseñada pensando en la interoperabilidad de agentes de usuario generales. Mediante un nuevo agente de usuario y un mecanismo de perfilado de documentos, distintos servidores, proxies y agentes de usuario podrán intercambiar contenidos. Finalmente, será posible desarrollar contenido en formato XHTML utilizable por cualquier agente de usuario XHTML.

XHTML (Lenguaje de Marcado de Hipertexto Extensible) es una versión más estricta y limpia de HTML. Surge con el objetivo de reemplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML.

XHTML extiende HTML 4.0, combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos.

XHTML surge como el lenguaje cuyo etiquetado, más estricto que HTML, va a permitir una correcta interpretación de la información, independientemente

del dispositivo desde el que se accede a ella. XHTML puede incluir otros lenguajes como MathML, SMIL o SVG.(W3C, 1999)

4.2.4.2. Beneficios

- Los documentos XHTML son compatibles con XML. Como tales, pueden verse, editarse y validarse con herramientas estándares de XML.
- Los documentos escritos en XHTML son compatibles con los agentes de usuario (ejemplo, navegadores) para HTML 4.
- Los documentos XHTML pueden utilizar aplicaciones (por ejemplo scripts, o applets) basados en el modelo de objetos de documento de HTML o en el modelo de objetos de documento (DOM) de XML.
- A medida que la familia XHTML evolucione, los documentos basados en XHTML 1.0 podrán operar en y entre varios entornos XHTML.

4.2.5. WSDL

A medida que la estandarización de los protocolos de comunicación se extiende, la posibilidad de encarar exitosamente una organización estructurada de las comunicaciones se convierte en una realidad. WSDL (del inglés Web ServicesDescriptionLanguage) es un estándar que permite encarar este objetivo ofreciendo descripciones de servicios.

WSDL permite describir la interfaz de un WebService, y cómo debe ser ligado con una dirección de red.

Generalmente se utiliza en combinación con SOAP para proveer servicios a través de Internet. Cualquier programa cliente que se conecte al Servicio Web puede determinar que funcionalidad está disponible a partir del documento WSDL asociado.

El estándar WSDL está siendo desarrollado por el W3C.(W3C, 2007)

4.2.6. Descripción de un WebService

La descripción de la interfaz del servicio por parte de WSDL se puede dividir en:

1. Definiciones
2. Operaciones
3. ServiceBindings(Ligaduras de Servicios)

Las definiciones son expresadas, generalmente, en XML. Se encargan de definir los tipos de datos y mensajes utilizados. El vocabulario XML empleado para las definiciones puede basarse en uno previamente acordado mediante algún tipo de definición estándar, o usarse uno especialmente definido si se planea un desarrollo para uso in-house.

Otros lenguajes pueden ser utilizados, como el IDL, aunque se opta por XML por ser el más común.

Para asegurar un uso único de nombres de elementos, definiciones, operaciones, etc., se utiliza un XML Namespace¹⁴ (Espacio de Nombres).

¹⁴Un espacio de nombres es un contenedor abstracto en el que un grupo de uno o más identificadores únicos pueden existir.

Mediante las operaciones se describe las acciones para los mensajes soportadas por el Servicio Web. Existen cuatro tipos de operaciones:

- De una vía (one-way): mensajes que no necesitan una respuesta.
- Pedido/Respuesta (request-response): el emisor envía un mensaje, y el receptor responde al recibirlo.
- Solicitud de respuesta (solicit response): un pedido de respuesta (la definición específica aún está pendiente).
- Notificación (notificación): envío de un mensaje a varios destinatarios (la definición específica aún está pendiente).

De esta forma las operaciones indican un patrón de intercambio de uno o más mensajes, indicando la cardinalidad, tanto de mensajes enviados como recibidos, y quienes los enviaron y/o recibieron.

Finalmente, las operaciones son agrupadas en porttypes o interfaces. Las interfaces definen, entonces, el conjunto de operaciones soportados por el Web Service. Cabe notar que

hasta el momento se ha llevado a cabo una definición de funcionalidad en un sentido abstracto, sin referirnos a ningún tipo de formato de transporte o plataforma subyacente. WSDL separa las anteriores definiciones de los detalles concretos de su implementación permitiendo la portabilidad y la reutilización de estas especificaciones.

El `Servicebindings` especifica la forma de conectar una interfase con un puerto o endpoint, definiendo una asociación con una dirección de red (por ejemplo, una URL) con dicha interfase. Describe cómo y dónde acceder al Servicio Web, por lo que se trata de una definición a un nivel concreto. El servicio queda definido entonces a partir de una colección de endpoints. El binding generalmente es creado utilizando SOAP, aunque otros protocolos para el pase de mensajes pueden ser utilizados, como por ejemplo CORBA IIOP, DCOM, etc.

4.2.7. Significado de la descripción de un servicio.

Una descripción de un `WebService` utilizando WSDL indica como se espera que potenciales clientes interactúen con el servicio descrito. Es una declaración de que el servicio Web implementa y adhiere a lo descrito el documento WSDL.

La interfaz definida, entonces, describe interacciones potenciales, no requeridas: la interacción descrita no es necesario que ocurra en absoluto, pero de llegar a ser iniciada, las operaciones describen como debe ocurrir.

UDDI, acrónimo de Universal Description, Discovery and Integration, define un registro para negocios o entidades que quieran dar a conocer en Internet sus servicios disponibles. Así, UDDI es un Servicio Web que maneja información sobre proveedores, implementaciones de servicios y datos adicionales sobre ellos. Está basado en XML y es independiente de cualquier plataforma. UDDI es otro de los estándares mantenidos por OASIS¹⁵.

UDDI permite confeccionar un catálogo de negocios, permitiendo publicar listados de servicios y descubrir otros.

15 OASIS, acrónimo de OrganizationfortheAdvancement of StructuredInformationStandards, es un consorcio internacional sin fines de lucro que orienta el desarrollo, la convergencia y la adopción de los estándares de comercio electrónico y servicios web.

También permite definir cómo los servicios o aplicaciones deben interactuar a través de Internet. De esta manera los proveedores de servicios Web pueden darse a conocer, y a los potenciales clientes realizar búsquedas por servicios que satisfagan sus necesidades.

La versión actual de UDDI es la V2.0 aprobada como un Estándar OASIS, y ya existen numerosos productos y servicios que la implementan. La versión V3.0 está actualmente bajo desarrollo. Asimismo, UDDI es complementario con otros proyectos de OASIS y el W3C, como ser SOAP, WSDL, WS-Security, etc.

La información que guarda un registro UDDI se puede dividir en un conjunto de categorías de datos simples:

- ¿Quién? (Who?) – Información sobre el negocio o institución. Nombre, identificación, dirección de contacto, etc.
- ¿Qué? (What?) – Información de categorización (por ejemplo, códigos industriales o clasificaciones

de productos) e información descriptiva de los servicios disponibles.

- ¿Dónde? (Where?) – Información de registro sobre la URL, dirección de correo electrónico u otro medio a través de la cual se accede al servicio.
- ¿Cómo? (How?) – Información de cómo funciona una interfase en particular del Servicio Web registrado.

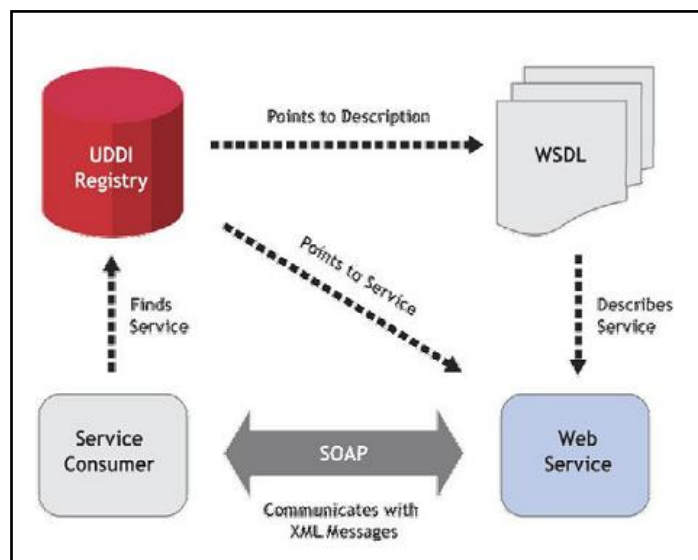


Figura 0.5: Tecnología basada en Servicios Web

UDDI está diseñada para interactuar con SOAP y WSDL. Por medio de un mensaje SOAP se puede interrogar a un servidor UDDI para que este responda con información de acceso a un Servicio Web particular, por medio de un documento WSDL.

4.2.8. Interfaces

El desarrollo de las interfaces del IDT, al estar basado en tecnología Web orientado al cliente, presenta formularios que simulan la ejecución de aplicaciones de escritorio sobre el navegador de internet.

4.2.9. Objetivo

El objetivo del “Sistema De Integración Transaccional Distribuido” es generar finalmente un archivo con la información de las transacciones de los puntos de venta o distribuidores de los clientes a los sistemas comerciales utilizados por cada uno, esto se logra mediante configuraciones que se establecen para determinar la manera como se va a obtener y generar la información.

4.2.10. Proceso

En este punto hacemos referencia al procesamiento de todos los movimientos que disponemos en la base de datos principal del sistema provenientes de cada uno de los puntos de venta, y de acuerdo a cada una de las configuraciones de los mismos en lo que respecta a la conversión de datos y parametrizaciones, procedemos con

creación del archivo que finalmente se generará para el Sistema de Información Gerencial.

4.3. Seguridades a implementar en el IDT (Hypertext Transfer ProtocolSecure)

Para efecto de nuestro sistema, la seguridad que vamos a emplear es la utilización del protocolo HTTPS (HypertextTransferProtocolSecure), tal como lo hemos detallado en el capítulo 3.4.

4.4. Diseño de las pantallas del IDT

De manera general se puede indicar que para el desarrollo del sistema y su interfaz grafica se utilizaron herramientas como:

MyEclipse: Herramienta utilizada para la edición de la interfaz

ExtJs: Utilizada para el diseño de las pantallas en ambiente Web.

En esta sección se presenta el diseño de las páginas web utilizado en la realización de este sistema, a continuación el detalle de las pantallas que lo componen.

❖ Ingreso al Sistema

Esta página muestra el ingreso al IDT, mediante la cual los usuarios podrán acceder a él y configurar la información que se

requiere para la correcta generación de los archivos que finalmente serán ingresados en su sistema comercial.

Figura 0.6: Formulario de Acceso al Sistema

La página de ingreso es igual para cualquier tipo de usuario que ingrese al sistema.

❖ Pantalla principal del sistema

La siguiente pantalla será la que se le muestre al usuario cada vez que ingrese al sistema en donde el podrá seleccionar a que opción del menú desea ingresar.

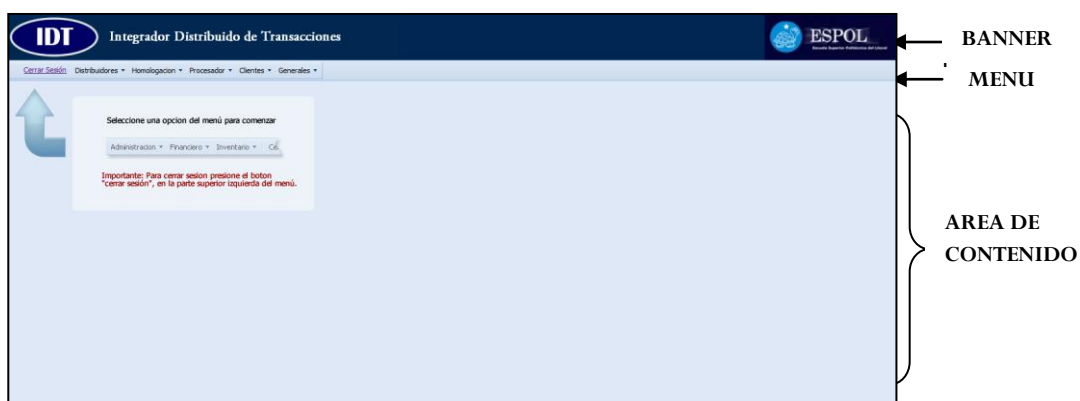


Figura 0.7: Pantalla Principal del sistema

La pantalla principal del sistema consta de 3 partes:

- ✓ Banner
- ✓ Menú
- ✓ Área de contenido

❖ **Descripción de las pantallas del sistema**

Antes de comenzar a describir cada una de las pantallas es necesario revisar el estándar utilizado en el sistema.

El área de contenido se divide en:

- Filtros: Utilizado para obtener información seleccionada dentro de la consulta de datos.
- Consulta: Consulta de información que ha sido ingresada al sistema.
- Botonería: Utilizada para realizar una acción dentro del mantenimiento o proceso del sistema.
- Paginación: Sección utilizada para mostrar la información que es consultada, se puede mediante ella indicar el número de registros que se va a cargar por cada página a visualizar, indicando adicionalmente cual es el total de registros encontrados en la consulta.

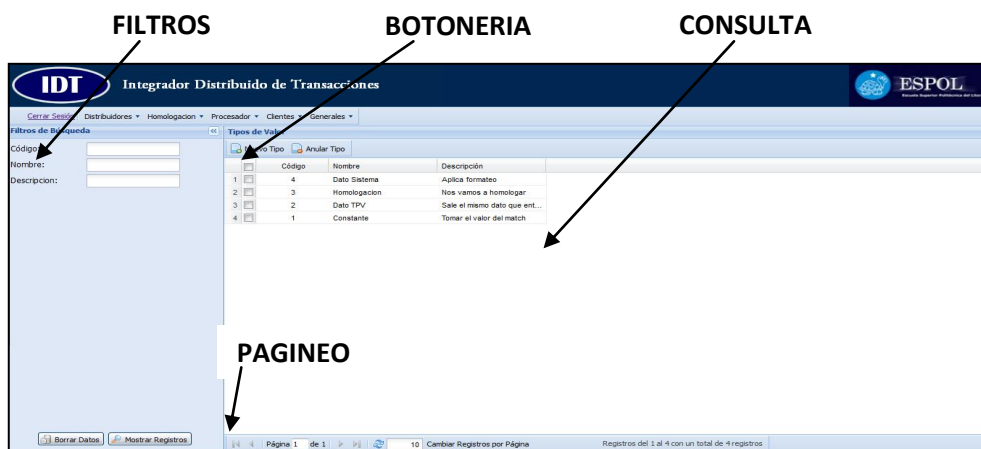


Figura 0.8: Descripción de Pantallas

❖ Mantenimientos

Los mantenimientos muestran las diferentes configuraciones que se pueden realizar en el sistema para lograr la integración de la información. Mediante estos se puede ingresar, modificar o eliminar los datos del sistema.

❖ Sistemas de Información

Son los diferentes sistemas que van a ser utilizados por los clientes, al ingresar a esta opción se presentara la siguiente pantalla que es el estándar explicado anteriormente.

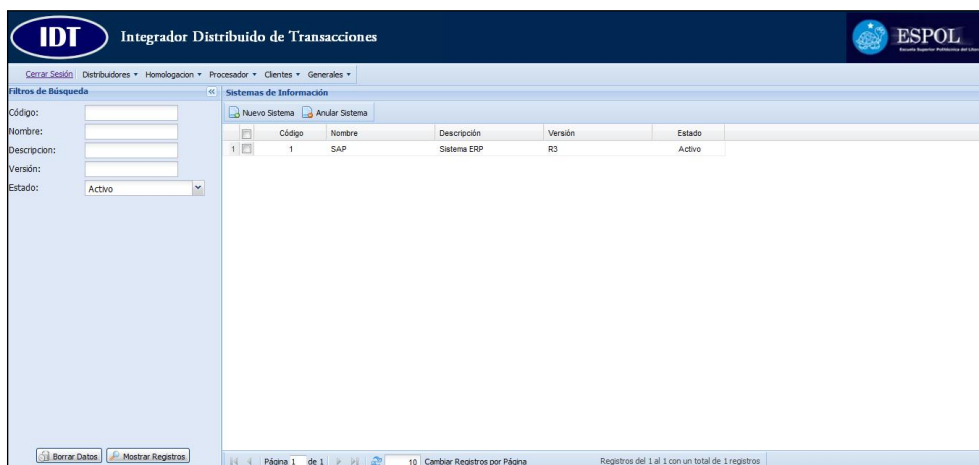


Figura 0.9Mantenimiento Sistemas de Información

Al dar clic en el botón Nuevo Sistema se puede ir agregando cada uno de los sistemas que van a ser utilizados. Así también dando doble clic se ejecuta el modo consulta del registro en caso de necesitar modificar sus datos.

❖ Clientes

Mediante esta pantalla se puede ir ingresando los diferentes clientes que van a utilizar el sistema de integración transaccional distribuido, estos clientes podrán copiar la información de sus transacciones realizadas en cada punto de venta a cada sistema comercial que haya sido definido.

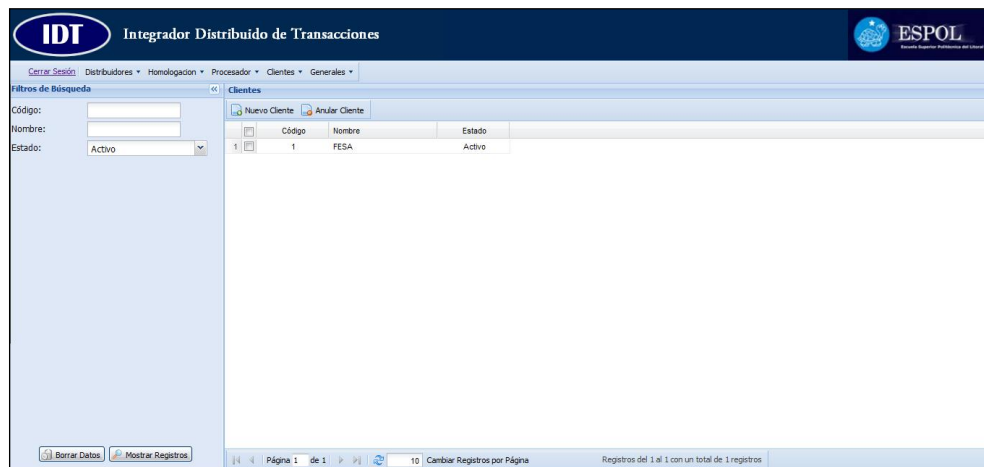


Figura 0.10: Mantenimiento de Clientes

Cada uno de estos clientes podrá acceder al sistema y procesar sus datos.

Esta opción le corresponde al administrador, que será el encargado de determinar cuáles serán los clientes que podrán utilizar el sistema integrado, este administrador podrá crear o eliminar los clientes que considere. A continuación se muestran las pantallas de creación o modificación de datos de los clientes.

The screenshot shows a software window titled "Consulta del Cliente" with three tabs: "Datos Generales", "Transacciones", and "Información Cliente". The "Datos Generales" tab is active. It contains two main sections: "General" and "SIG".

General Section:

- Tipo Cliente: Corporativo (dropdown)
- Nombre: Banco Bolivariano (text input)
- Descripción: El banco con vision (text input)
- Estado: Activo (dropdown)

SIG Section:

- SIG: SIG_1 (dropdown)
- Modo de Acceso: Web Service (dropdown)
- Parámetros: xxxxxxxx (text input)

At the bottom of the window, there are three buttons: "Guardar Cambios", "Anular", and "Cancelar".

Figura 0.11: Datos Generales del Cliente

En esta pestaña se ingresan los datos principales del cliente y se determina el tipo de comunicación que utilizara para integrarse con su sistema comercial. El sistema de integración transaccional "IDT" trabaja con el modo de acceso Servicio Web o Web Service por ser una tecnología de comunicación más segura, sin embargo se deja abierta la posibilidad de que se determinen más modos de acceso para que en un futuro se pueda ampliar esta funcionalidad.

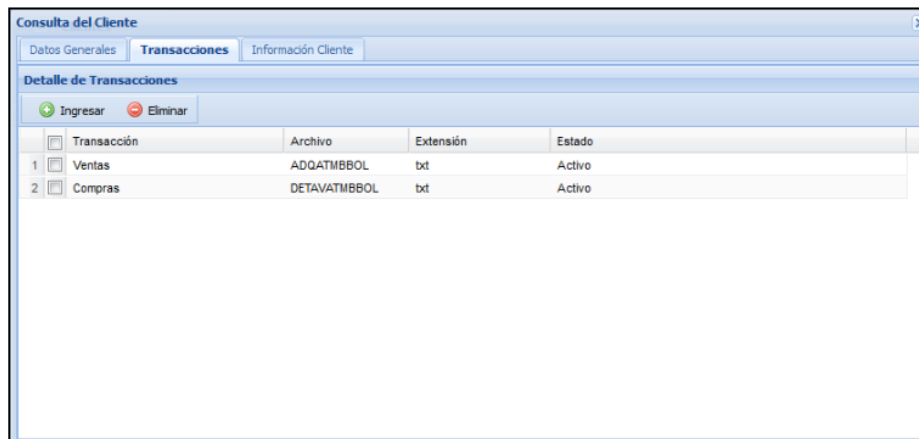


Figura 0.12: Transacciones

En esta pestaña se pueden indicar las transacciones que va a manejar el cliente de manera general, estas transacciones son las que se obtienen de los puntos de venta y que alimentaran al sistema de información definido.

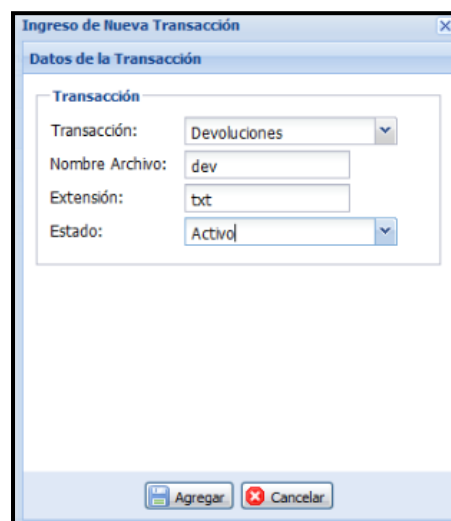
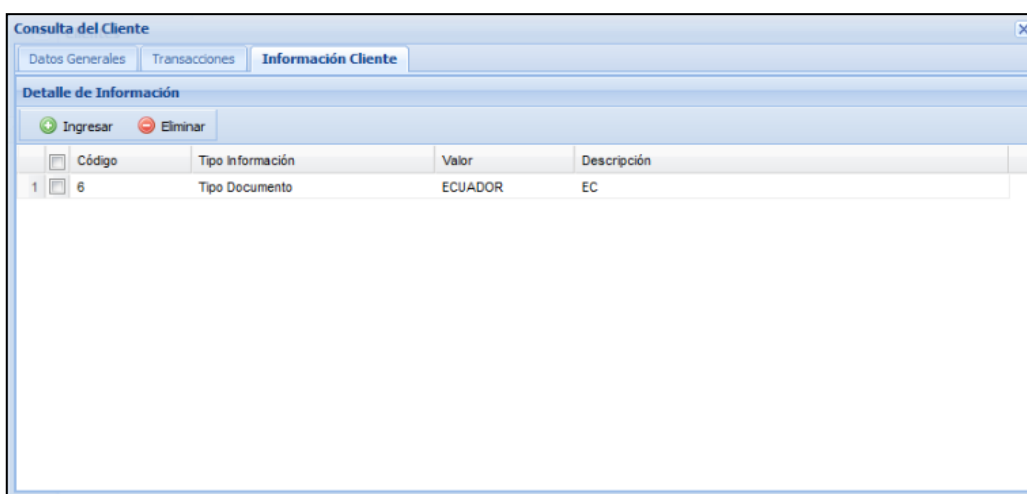


Figura 0.13: Ingreso de Transacciones

Debido a que aquí se define lo relacionado a la información de las transacciones que se van a extraer de cada terminal o punto de venta, se debe determinar el nombre del archivo que será subido al sistema y en que extensión vendrá, dependiendo de cada tipo de transacción indicada.



	Código	Tipo Información	Valor	Descripción
1	6	Tipo Documento	ECUADOR	EC

Figura 0.14: Ingreso de Información del Cliente

❖ Distribuidores

Son las diferentes sucursales correspondientes a los clientes, centros de distribución o agencias que utilizan algún sistema comercial (sistema de información) y de donde se va a obtener los datos para ser subidos a SIG de la manera definida.

La pantalla conserva el estándar de los mantenimientos y al igual que los demás permite modificar, agregar o eliminar distribuidores al sistema.

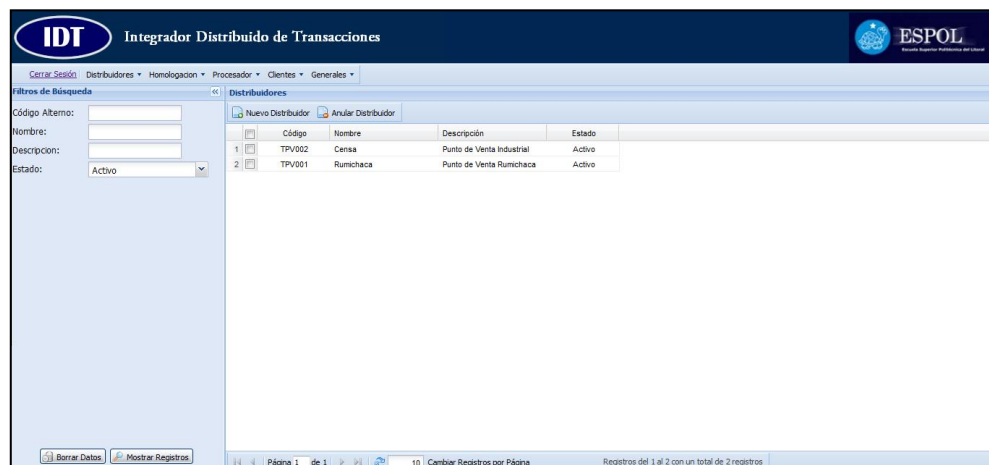


Figura 0.15: Mantenimiento de Distribuidores

❖ Tipos de clientes

Permite clasificar a los clientes por tipo, es una manera de organizar la información de cada uno.

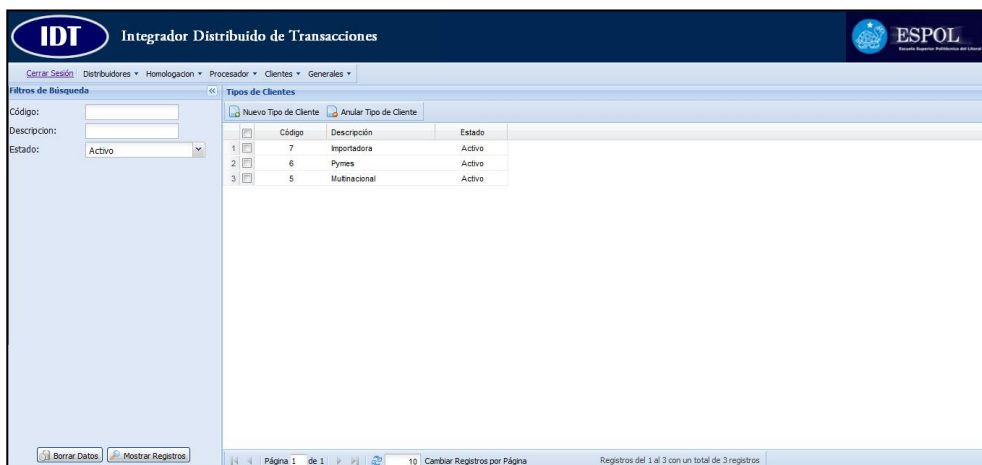


Figura 0.16: Mantenimiento de Tipos de Clientes

❖ Tipo de Dato

Este mantenimiento permite indicar todos los tipos de datos que se van a utilizar en el sistema, estos tipos de datos corresponden a cada campo que ha sido definido en el sistema. Aquí se pueden ingresar otros tipos de datos reconocidos si es que se necesitan para el desarrollo de la aplicación.

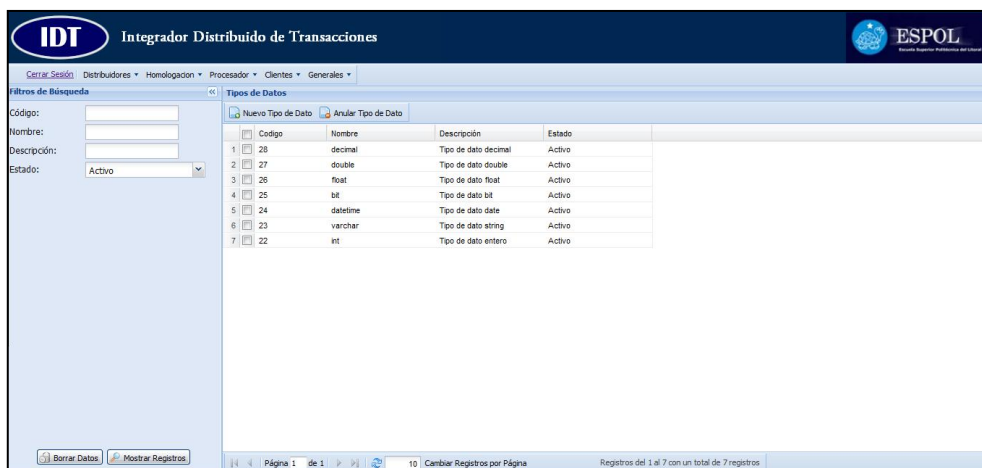


Figura 0.17: Mantenimiento de Tipos de Datos

❖ Tipos de Información

Permite clasificar la información de los clientes por tipos, de esta manera tendremos una mejor organización de los mismos.

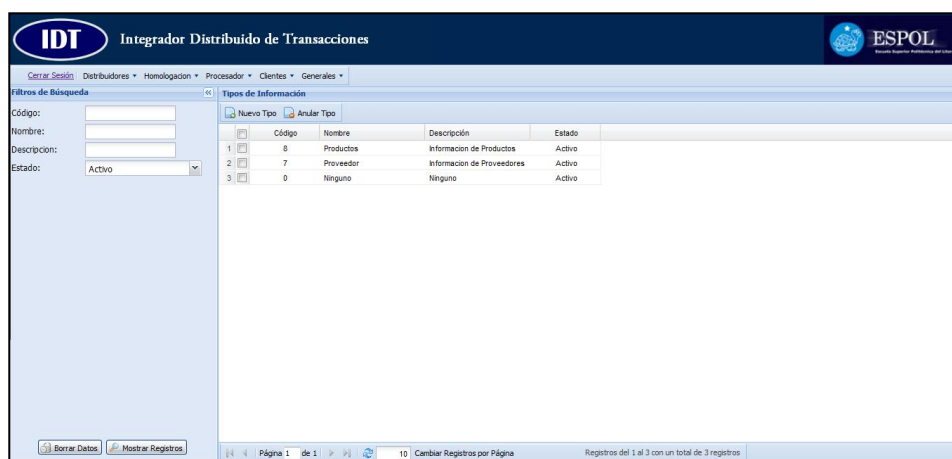


Figura 0.18: Mantenimiento de Tipos de Información

❖ Tipo de Valor

En esta pantalla se encuentran los tipos de valor que tiene el sistema, y que a su vez éstos serán utilizados para la conversión de datos de los distintos campos.

Estos tipos de valor solo se ingresarán una sola vez en el sistema y no se podrán dar mantenimiento. Los valores que pueden tener estos tipos se muestran en la pantalla a continuación.

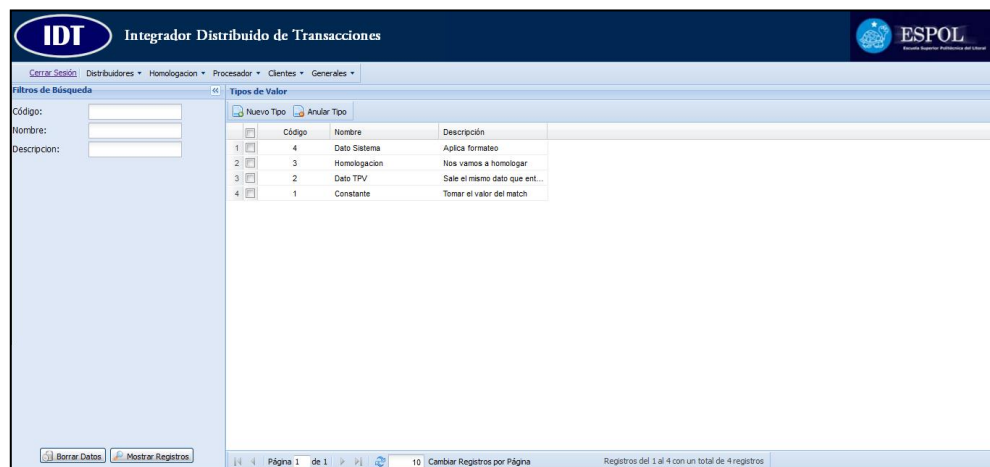


Figura 0.19: Mantenimiento de Tipos de valor

❖ Formatos

En esta pantalla se da mantenimiento a los diferentes formatos que son utilizados para la conversión del campo origen (distribuidores) hacia el destino (Ejm: SIG).

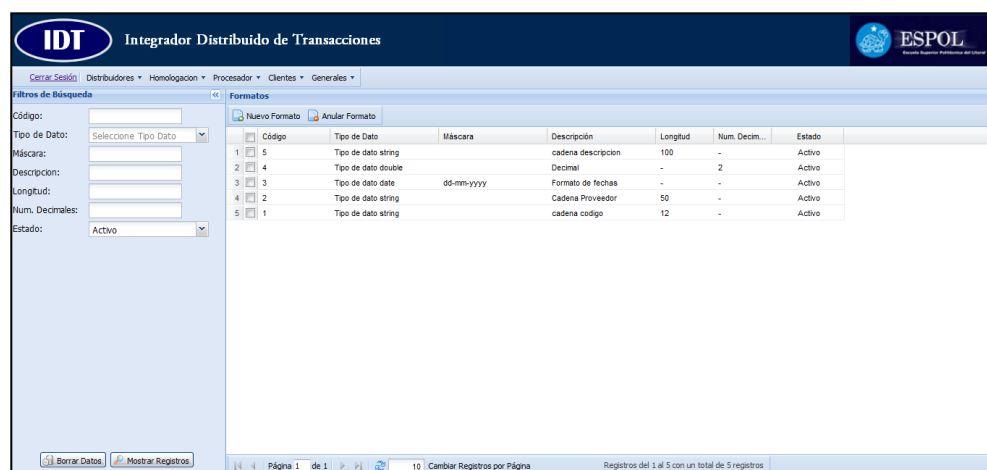


Figura 0.20: Mantenimiento de Formatos

❖ Campo Origen

En esta pantalla podemos registrar o dar mantenimiento a cada uno de los campos que intervienen en cada transacción perteneciente al punto de venta y ésta a su vez a un cliente en particular.

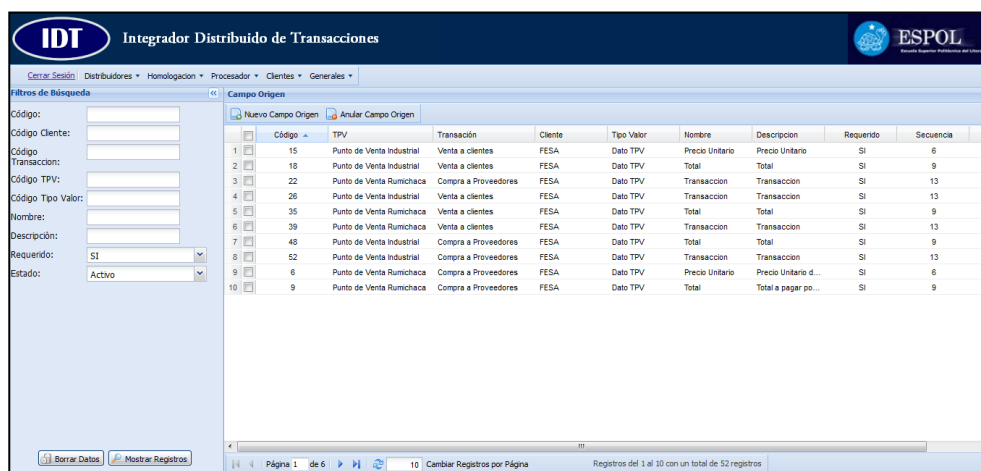


Figura 0.21: Mantenimiento de Campo Origen

❖ Campo Destino

En esta pantalla podemos registrar o dar mantenimiento a cada uno de los campos destino, es decir los campos que intervienen en cada transacción perteneciente al sistema de información gerencial. Aquí también indicamos el formato y el tipo de dato que debe tener el campo.

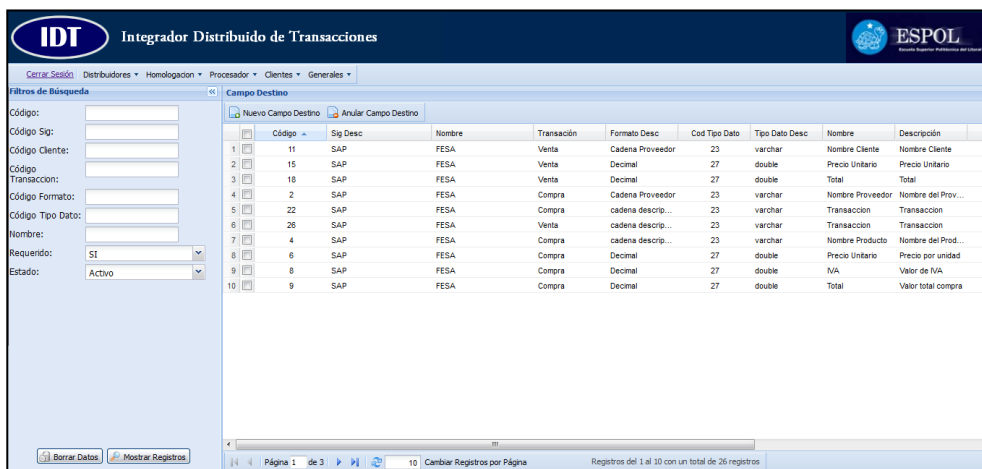


Figura 0.22: Mantenimiento de Campo Destino

❖ Homologación

Aquí describimos el valor que debe tomar el campo origen para llegar al destino, cuando dicho campo se encuentre registrado el tipo valor como homologación.

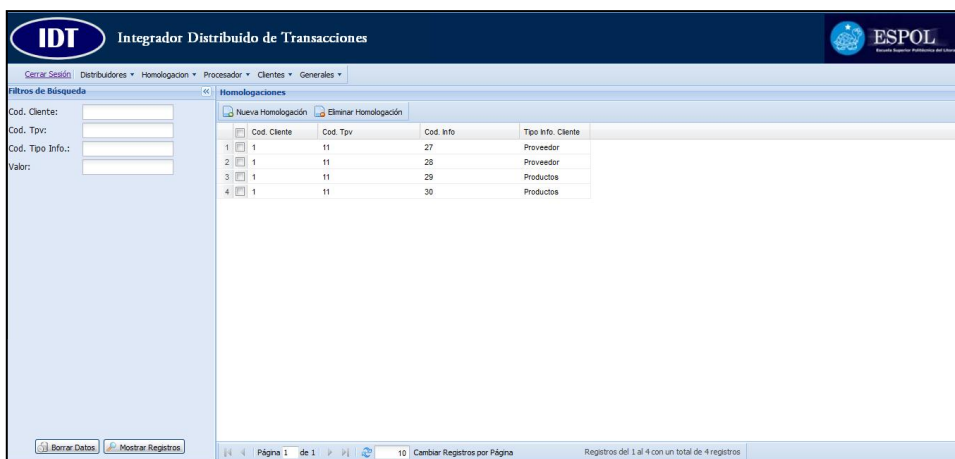


Figura 0.23: Homologación

❖ Match

En esta pantalla enlazamos cada campo origen de las transacciones del punto de venta, con cada campo destino de las transacciones en el sistema de información gerencial.

Tomar en cuenta también que tanto la secuencia del campo origen como la secuencia del campo destino, deben ser las mismas como para poder realizar el match entre ambas.

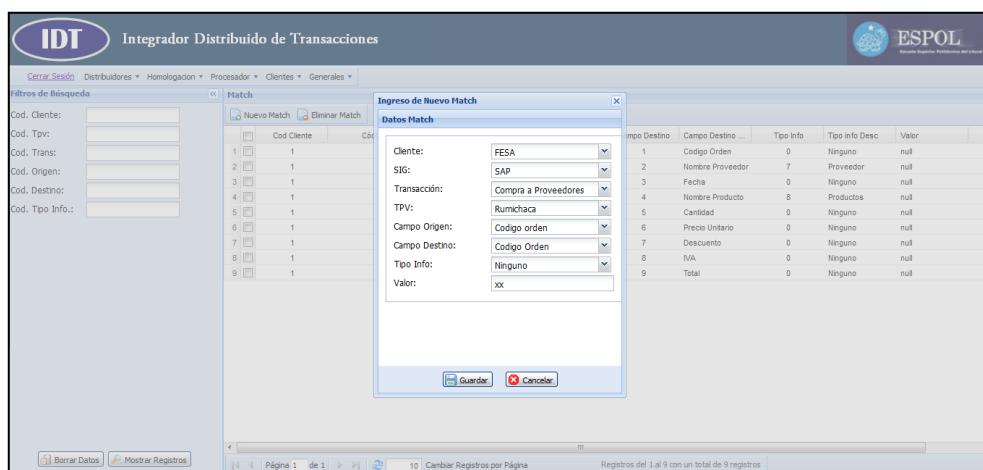


Figura 0.24: Proceso de Match

❖ Transacciones

Por medio de este mantenimiento podemos agregar, modificar o eliminar transacciones, las mismas que luego serán asociadas a un cliente en particular, y que a su vez luego serán relacionadas a los distintos puntos de venta configurados a dicho cliente.

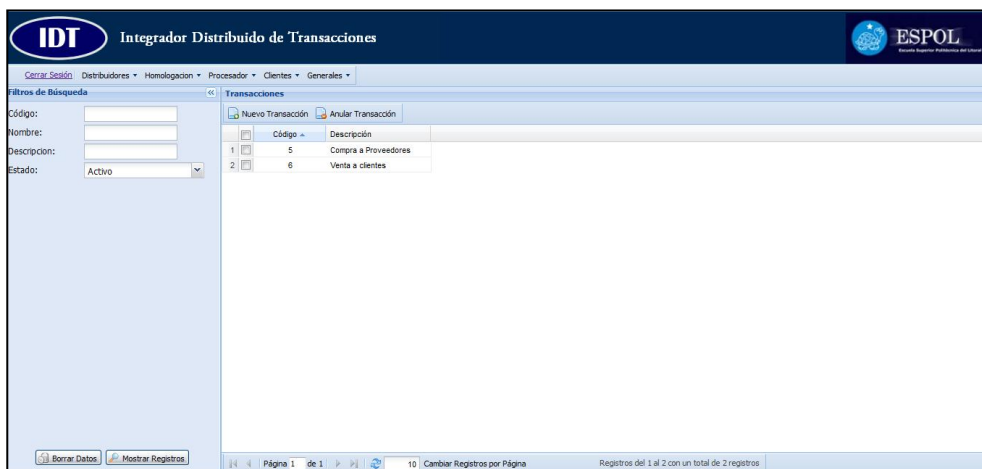


Figura 0.25: Transacciones del Sistema

❖ Transacciones por Cliente

En esta pantalla es donde ya procedemos a asociar las transacciones anteriormente creadas con los clientes.

A su vez también podemos relacionarlas con los puntos de venta o distribuidores configurados.

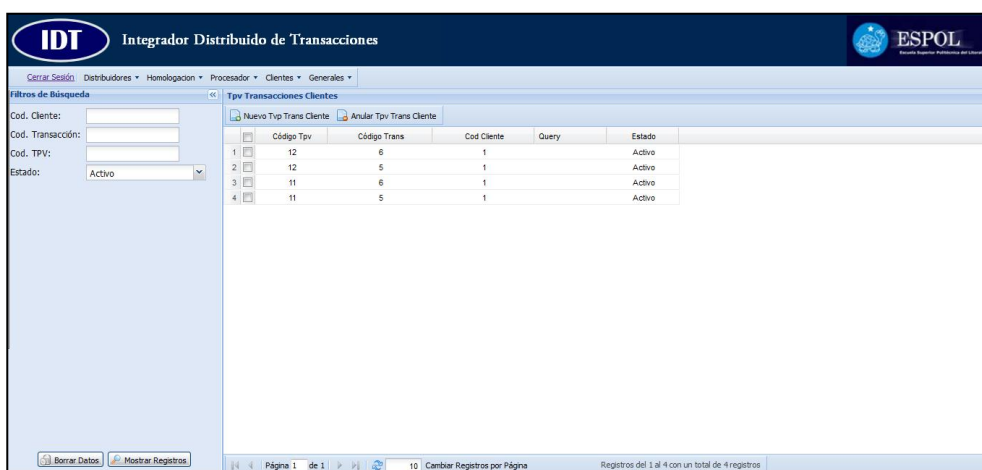


Figura 0.26: Transacciones por Cliente

❖ Procesamiento de movimientos

Por medio de esta opción procedemos a procesar todos los movimientos que los puntos de venta han llegado a transmitir al servidor.

Al ejecutar el procesador se iniciará la conversión respectiva de todos los campos originarios de los puntos de venta en cada una de las transacciones e acuerdo a las configuraciones establecidas en el servidor.

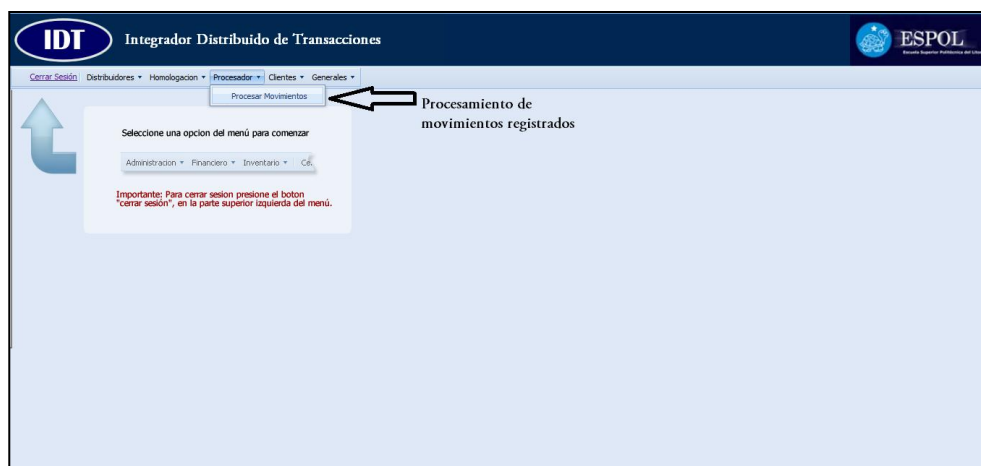


Figura 0.27: Procesador de movimientos

4.5. Diseño de pruebas

Las pruebas que se harán dentro del IDT como verificación de que el sistema cumpla con lo que fue diseñado, contendrán las siguientes fases:

- Pruebas de Unidad
- Pruebas de Integración
- Pruebas de Validación
- Análisis de resultados

Cada una de estas fases se analizará más a detalle en el siguiente capítulo.

CAPÍTULO 5

Implementación y Pruebas

5.1. Selección de las plataformas de desarrollo del sistema

Para el desarrollo del IDT, se evaluaron tres plataformas que son calificadas como las más potentes en cuanto a creación de portales web robustos y seguros:

- Java
- PHP
- ASP .Net

Cada de una de estas plataformas ya fueron descritas anteriormente en el capítulo 3.5.3.2.

La plataforma de desarrollo ASP – ASPX, fue descartada al tener limitaciones en cuanto a que sólo puede ser implementada sobre

tecnología Microsoft, lo que implicaba tener que instalar el servidor de Aplicaciones Internet Information Services y a su vez a tener que incurrir en los costos de licenciamiento.

En el análisis acerca de seleccionar cual es la mejor tecnología para el desarrollo web, se lo realizó tomando en cuenta los siguientes criterios:

- **Modularización**

Consideramos el término Modularización como la separación en capas definidas en un modelo MVC (Modelo Vista-Controlador).

La modularidad de un sistema tiene vital importancia en el aspecto de la consistencia, robustez, mantenibilidad y demás aspectos que detallaremos más adelante.

La tecnología Java usada en cualquier portal web posee una estructura claramente diferenciada, pudiendo diferenciar con facilidad el modelo MVC con sus diferentes módulos:

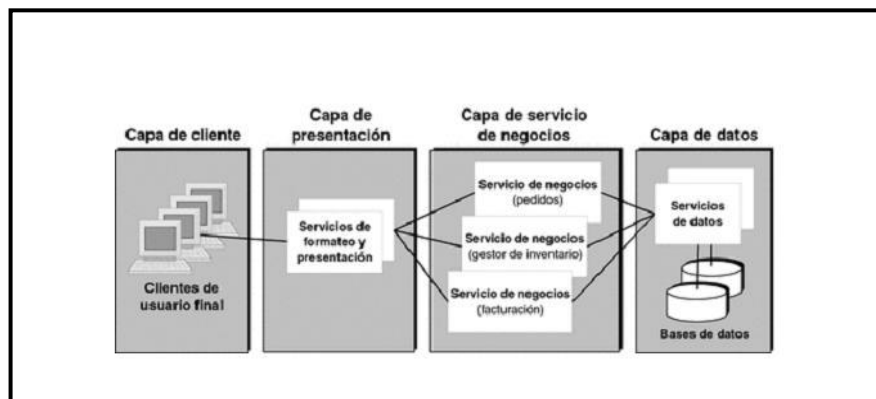


Figura 0.1: Ejemplo de Modularización

En cuanto a PHP, podemos decir que perdemos un tanto la pista de la modularidad que hemos destacado en java puesto que todas las capas lógicas son implementadas en un mismo archivo .php.

▪ **Mantenibilidad**

La mantenibilidad del sistema es una parte fundamental en el ciclo de vida de cualquier proyecto que estemos tratando, y está estrechamente relacionada con la tecnología que hayamos elegido en la etapa de diseño.

Para realizar el análisis de las técnicas que estamos tratando debemos remontarnos de nuevo al punto anterior para conseguir sacar una conclusión firme. Un sistema en el que exista una estructura clara de sus componentes será más

facilmente mantenible en un futuro ya que será necesario el seguimiento de una metodología ya definida, lo que evitará un empobrecimiento de su código y por tanto de su rendimiento.

- **Crecimiento del Sistema**

Cuando se realiza el diseño de un proyecto y se elige una tecnología a usar, durante la etapa de desarrollo se debe orientar el proceso al pensamiento de que una vez implementado y puesto en funcionamiento, serán necesarias nuevas mejoras, que no deben suponer un coste demasiado elevado ni que mucho menos produzca un empobrecimiento del sistema actual.

Por experiencia sabemos que una vez finalizado el desarrollo, puede ocurrir que las mejoras del sistema no sean implementadas por su programador original y sí por otra persona o empresa externa. El hecho de que PHP sea un lenguaje de programación poco estructurado y que el desarrollador no sea pleno conocedor de la estrategia seguida en la programación original, puede dar origen a un empobrecimiento del código, repercutiendo normalmente a su rendimiento.

- **Costo de Desarrollo**

El costo estimado en un proyecto PHP siempre será menor que en un proyecto Java. Mientras que la programación de un sistema PHP es mucho más directa con resultados inmediatos, el uso de Java supone el montaje de la estructura mencionada en los puntos anteriores, lo que alarga el tiempo de desarrollo y con esto, su coste.

Toda inversión en un proyecto debe ser estimada de una manera realista, teniendo en cuenta la magnitud del mismo y echando un vistazo a largo plazo. El sistema Java está más orientada a grandes proyectos ya que, como hablaremos más adelante proporciona una mayor escalabilidad que PHP.

- **Formación**

Está claro que para lograr el éxito en un sistema, este ha debido ser diseñado y desarrollado por los mejores y en su mejor versión. A los costes de desarrollo mencionados anteriormente debemos sumar otros, que si bien pueden no ser económicos, también debemos tener en cuenta cuando diseñamos un proyecto Software.

Como llevamos viendo desde el inicio del documento, está demostrado que Java es una tecnología mucho más amplia y

desarrollada que PHP, lo que nos llevará a un coste de formación mayor.

- **Integración externa**

Nos referimos a “Integración externa” como el uso de herramientas, métodos y funcionalidades desarrollados por otros programadores y que son integrables fácilmente en nuestro sistema.

El mundo Java es muy amplio y variado. Por un lado existen diferentes Frameworks que facilitarán la tarea de los programadores, pudiendo obtener los mismos resultados (o incluso mejores) que sin ellos en un tiempo más breve. Por otro lado, existe lo que podemos llamar “módulos” ya desarrollados y de libre distribución que podemos usar en nuestro proyecto.

Si realizamos la comparación con PHP, existe una gran diferencia, ya que este último como vimos al principio es mucho menos estructurado (podemos llamarlo “all-in-one”) y es mucho más complicado encontrar un módulo complejo completo integrable con facilidad.

- **Seguridad**

El aspecto de la seguridad siempre ha sido un punto a tener muy en cuenta en cualquier sistema informático y un portal web es especialmente vulnerable por estar expuesto a todo el público en internet.

Java implementa en diferentes niveles un sistema seguro de validación que en PHP echamos en falta.

Como sistemas de seguridad usados en proyectos Java cabe destacar los que se implementan a nivel de Servidor de aplicaciones (como “JAAS”) y los que están incluidos en Frameworks externos (como por ejemplo “Spring Security” o “ACEGI”), ambos eficaces y transparentes a usuarios y programadores. En el desarrollo de un portal web con PHP debemos controlar la seguridad de acceso a nuestro sistema de una forma mucho más manual, realizando comprobaciones minuciosas de los diferentes ataques que podemos recibir (como por ejemplo SQL Injection).

- **Rendimiento**

Quizás una de las ventajas de PHP frente a Java sea en cuestión de rendimiento ya que el primero es mucho menos

pesado, lo que produce una sensación al usuario de rapidez y mayor usabilidad.

▪ Escalabilidad

Uno de los temas que siempre ha seguido al debate de PHP o Java ha sido el tema de la escalabilidad del sistema, es decir, propiedad por la cual un sistema no empeora su rendimiento y funcionalidad ante un número creciente de usuarios.

Desde hace mucho tiempo siempre se ha dicho que la tecnología Java es mucho más escalable que PHP, demostrándose la pérdida de rendimiento de este último ante un aumento considerable de usuarios concurrentes. Con PHP5, los desarrolladores y seguidores de esta tecnología apoyan que dicha deficiencia ha sido solucionada aunque sigue sin demostrarse de manera real.

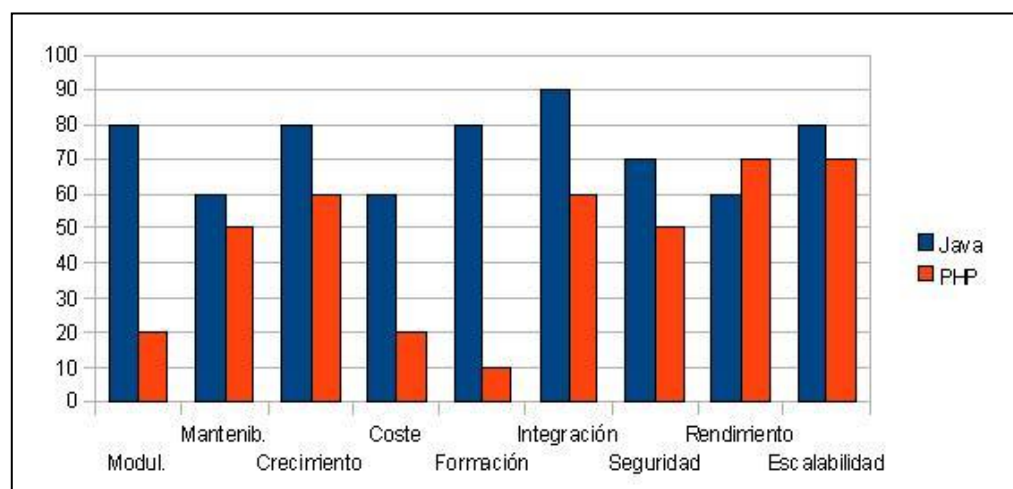


Figura 0.2: Escalabilidad

el análisis anterior y el cuadro comparativo pudimos determinar que la mejor plataforma de desarrollo para proyectos grandes y que requieren de seguridad por el tipo de información que se va a transmitir, es JAVA.

5.2. Estructura y parametrización de la base de datos

Para efecto de poder cumplir con el objetivo y para lo cual fue diseñado el IDT, el sistema consta de un modelo lógico que será descrito en el siguiente diccionario de datos:

Nombre	Número
SITD_SIG	1
SITD_CLIENTE	2
SITD_TIPO_CLIENTE	3
SITD_TIPO_INFO	4
SITD_MODO_ACCESO	5
SITD_TIPO_VALOR	6
SITD_TIPO_DATO	7
SITD_SIG_CLIENTE	8
SITD_TPV	9
SITD_TRANSACCION	10
SITD_TPV_TRANS_CLIENTE	11
SITD_CAMPO_DESTINO	12

SITD_FORMATO	13
SITD_CAMPO_ORIGEN	14
SITD_MATCH	15
SITD_INFO_CLIENTE	16
SITD_TRANSACCION_X_CLIENTE	17
SITD OPCIONES	18
SITD_USUARIO	19

Tabla 5: Diccionario de Datos

SITD_CAMPO_DESTINO

Nombre	SITD_CAMPO_DESTINO
Descripción	Tabla que contiene la parametrización para la salida de la información.
Secuencia	1

Tabla 6: Definición SITD_CAMPO_DESTINO

Atributos

Nombre	Tipo de Dato	Requerido
codigo	Variable multibyte (10)	X
nombre	Variable multibyte (100)	
descripcion	Variable multibyte (255)	
requerido	Boolean	
estado	Boolean	
cabecera	Boolean	

Tabla 7: Atributos SITD_CAMPO_DESTINO

SITD_CAMPO_ORIGEN

Nombre	SITD_CAMPO_ORIGEN
Descripción	Tabla que registra el detalle de la información a ser extraída del origen.
Secuencia	2

Tabla 8: Definición SITD_CAMPO_ORIGEN

Atributos

Nombre	Tipo de Dato	Requerido
codigo	Variable multibyte (10)	X
nombre	Variable multibyte (100)	
descripcion	Variable multibyte (255)	
requerido	Boolean	
estado	Boolean	

Tabla 9: Atributos SITD_CAMPO_ORIGEN

SITD_CLIENTE

Nombre	SITD_CLIENTE
Descripción	Tabla para almacenar todos los clientes usuarios del SITD
Secuencia	3

Tabla 10: Definición SITD_CLIENTE

Atributos

Nombre	Tipo de Dato	Requerido
codigo	Variable multibyte (10)	X
nombre	Variable multibyte (100)	
descripcion	Variable multibyte (100)	
estado	Boolean	
fecha_ingreso	Date & Time	

Tabla 11: Atributos SITD_CLIENTE

SITD_FORMATO

Nombre	SITD_FORMATO
Descripción	Tabla de definición de formato de los datos
Secuencia	4

Tabla 12: Definición SITD_FORMATO

Atributos

Nombre	Tipo de Dato	Requerido
codigo	Variable multibyte (10)	X
mascara	Variable multibyte (255)	
descripcion	Variable multibyte (255)	
longitud	Integer	
num_decimales	Integer	

estado	Boolean	
--------	---------	--

Tabla 13: Atributos SITD_FORMATO

SITD_INFO_CLIENTE

Nombre	SITD_INFO_CLIENTE
Descripción	Tabla de ingreso de datos de información propia del cliente para el archivo de salida.
Secuencia	5

Tabla 14: Definición SITD_INFO_CLIENTE

Atributos

Nombre	Tipo de Dato	Requerido
codigo	Variable multibyte (10)	X
valor	Variable multibyte (100)	
descripcion	Variable multibyte (255)	

Tabla 15: Atributos SITD_INFO_CLIENTE

SITD_MATCH

Nombre	SITD_MATCH
Descripción	Tabla de parámetros de transformación y homologación de datos.
Secuencia	6

Tabla 16: Definición SITD_MATCH

Atributos

Nombre	Tipo de Dato	Requerido
metodo	Variable multibyte (1)	
estado	Boolean	
valor	Variable multibyte (100)	

Tabla 17: Atributos SITD_MATCH

SITD_MODO_ACCESO

Nombre	SITD_MODO_ACCESO
Descripción	Tabla para almacenar los diferentes modos de acceso a la información
Secuencia	7

Tabla 18: Definición SITD_MODO_ACCESO

Atributos

Nombre	Tipo de Dato	Requerido
codigo	Variable multibyte (10)	X
descripcion	Variable multibyte (200)	
estado	Boolean	
num_param	Integer	
separador	Variable multibyte (1)	
secuencia_param	Variable multibyte (300)	

Tabla 19: Atributos SITD_MODO_ACCESO

SITD_OPCIONES

Nombre	SITD_OPCIONES
Descripción	Tabla donde se registra las opciones que tiene el sistema
Secuencia	8

Tabla 20: Definición SITD_OPCIONES

Atributos

Nombre	Tipo de Dato	Requerido
codigo	Variable multibyte (10)	X
nombre	Variable multibyte (100)	
descripcion	Variable multibyte (255)	
url	Variable multibyte (100)	
estado	Boolean	

Tabla 21: Atributos SITD_OPCIONES

SITD_SIG

Nombre	SITD_SIG
Descripción	Tabla para los diferentes sistemas de información a los que alimentará el SITD.
Secuencia	9

Tabla 22: Definición SITD_SIG

Atributos

Nombre	Tipo de Dato	Requerido
codigo	Variable multibyte (10)	X

nombre	Variable multibyte (100)	
descripcion	Variable multibyte (200)	
estado	Boolean	
version	Variable multibyte (10)	

Tabla 23: Atributos SITD_SIG

SITD_SIG_CLIENTE

Nombre	SITD_SIG_CLIENTE
Descripción	Tabla que registra los sistemas de información que maneja el cliente.
Secuencia	10

Tabla 24: Definición SITD_SIG_CLIENTE

Atributos

Nombre	Tipo de Dato	Requerido
param_acceso	Variable multibyte (255)	

Tabla 25: Atributos SITD_SIG_CLIENTE

SITD_TIPO_CLIENTE

Nombre	SITD_TIPO_CLIENTE
Descripción	Tabla genérica para almacenar los diferentes tipos de clientes que maneja el SITD
Secuencia	11

Tabla 26: Definición SITD_TIPO_CLIENTE

Atributos

Nombre	Tipo de Dato	Requerido
codigo	Variable multibyte (10)	X
descripcion	Variable multibyte (100)	
estado	Boolean	

Tabla 27: Atributos SITD_TIPO_CLIENTE

SITD_TIPO_DATO

Nombre	SITD_TIPO_DATO
Descripción	Tabla para almacenar los diferentes tipos de datos que se van a manejar en el SITD
Secuencia	12

Tabla 28: Definición SITD_TIPO_DATO

Atributos

Nombre	Tipo de Dato	Requerido
codigo	Variable multibyte (10)	X
nombre	Variable multibyte (100)	
descripcion	Variable multibyte (200)	
estado	Boolean	

Tabla 29: Atributos SITD_TIPO_DATO

SITD_TIPO_INFO

Nombre	SITD_TIPO_INFO
Descripción	Tabla para almacenar los diferentes tipos de información que el cliente desea transformar
Secuencia	13

Tabla 30: Definición SITD_TIPO_INFO

Atributos

Nombre	Tipo de Dato	Requerido
codigo	Variable multibyte (10)	X
nombre	Variable multibyte (100)	
descripcion	Variable multibyte (100)	
estado	Boolean	

Tabla 31: Atributos SITD_TIPO_INFO

SITD_TIPO_VALOR

Nombre	SITD_TIPO_VALOR
Descripción	Tabla para almacenar los diferentes tipos de valores de los campos de la transacción. Ejm: Si es constante o transformacionindicando su método
Secuencia	14

Tabla 32: Definición SITD_TIPO_VALOR

Atributos

Nombre	Tipo de Dato	Requerido
--------	--------------	-----------

codigo	Variable multibyte (10)	X
nombre	Variable multibyte (100)	
descripcion	Variable multibyte (200)	

Tabla 33: Atributos SITD_TIPO_VALOR

SITD_TPV

Nombre	SITD_TPV
Descripción	Tabla para registrar los datos principales del Distribuidor (Punto de origen de Transacciones)
Secuencia	15

Tabla 34: Definición SITD_TPV

Atributos

Nombre	Tipo de Dato	Requerido
codigo	Integer	X
nombre	Variable multibyte (30)	
descripcion	Variable multibyte (255)	
codigo_alterno	Variable multibyte (10)	
estado	Boolean	
param_acceso	Variable multibyte (255)	
url	Variable multibyte (100)	

Tabla 35: Atributos SITD_TPV

SITD_TPV_TRANS_CLIENTE

Nombre	SITD_TPV_TRANS_CLIENTE
Descripción	Tabla relacional que asocia las transacciones del cliente a los diferentes puntos donde se encuentran los datos.
Secuencia	16

Tabla 36: Definición SITD_TPV_TRANS_CLIENTE

Atributos

Nombre	Tipo de Dato	Requerido
query	Variable multibyte (4000)	
estado	Boolean	

Tabla 37: Atributos SITD_TPV_TRANS_CLIENTE

SITD_TRANSACCION

Nombre	SITD_TRANSACCION
Descripción	Tabla general de las líneas de negocio que el cliente integra en el SITD.
Secuencia	17

Tabla 38: Definición SITD_TRANSACCION

Atributos

Nombre	Tipo de Dato	Requerido
codigo	Variable multibyte (10)	X

descripcion	Variable multibyte (100)	
estado	Boolean	

Tabla 39: Atributos SITD_TRANSACCION

SITD_TRANSACCION_X_CLIENTE

Nombre	SITD_TRANSACCION_X_CLIENTE
Descripción	Tabla que relaciona las transacciones parametrizadas en el sistema con el cliente.
Secuencia	18

Tabla 40: Definición SITD_TRANSACCION_X_CLIENTE

Atributos

Nombre	Tipo de Dato	Requerido
estado	Boolean	
nombre_archivo	Variable multibyte (100)	
extension	Variable multibyte (5)	

Tabla 41: Atributos SITD_TRANSACCION_X_CLIENTE

SITD_USUARIO

Nombre	SITD_USUARIO
Descripción	Tabla que administra los usuarios que tienen acceso al SITD
Secuencia	19

Tabla 42: Definición SITD_USUARIO

Atributos

Nombre	Tipo de Dato	Requerido
codigo	Variable multibyte (10)	X
nombres	Variable multibyte (100)	
apellidos	Variable multibyte (100)	
user	Variable multibyte (10)	
password	Variable multibyte (10)	

Tabla 43: Atributos SITD_USUARIO

SITD_HOMOLOGACION

Nombre	SITD_HOMOLOGACION
Descripción	Si el codigo del tipo_info es homologación, venir a esta tabla y tomar el valor. Parto del Match - Info_Cliente–Homologación
Secuencia	20

Tabla 44: Definición SITD_HOMOLOGACION

Atributos

Nombre	Tipo de Dato	Requerido
valor	Variable multibyte (100)	

Tabla 45: Atributos SITD_HOMOLOGACION

SITD OPCIONES_X_USUARIO

Nombre	SITD OPCIONES_X_USUARIO
Descripción	Tabla donde se relaciona las opciones del sistema a cada uno de los usuarios.
Secuencia	21

Tabla 46: Definición SITD OPCIONES_X_USUARIO

Atributos

Nombre	Tipo de Dato	Requerido
codigo_padre	Variable multibyte (10)	
nivel	Integer	
estado	Boolean	

Tabla 47: Atributos SITD OPCIONES_X_USUARIO

5.3. Implementación de los módulos del sistema

5.3.1. Servidor

Este módulo contiene las funciones que permiten la extracción, validación y transformación de la información extraída desde los puntos de distribución (origen), hasta los repositorios definidos de los clientes donde se ubicará la información lista para la carga y procesamiento en SIG (destino).

Dentro de la operación de extracción se encuentra la comunicación segura cliente - servidor para intercambiar los datos de conexión y sentencias de consulta que se van a ejecutar en el origen.

El procedimiento de validación se realiza en el servidor y permite verificar que la información extraída se encuentre en el formato y tipo de dato definido para cada campo de la transacción.

La transformación permite que cierta información que ha sido extraída se convierta al formato parametrizado por el cliente ó sea homologado según los criterios especificados por él en las tablas de información a homologar según la codificación particular del cliente.

5.3.2. Cliente PC

Este módulo se requiere instalar en cada punto donde se encuentra un origen de información a ser extraída.

Contiene la lógica necesaria para el intercambio de datos con el servidor de manera segura.

Entre las funcionalidades de este módulo se encuentra la conexión al origen de datos con los parámetros consultados al servidor, la extracción de la información utilizando las

sentencias de consulta parametrizadas con anterioridad en el servidor para cada punto de distribución y el envío seguro de los datos extraídos desde el origen hasta el servidor, para su posterior procesamiento y alojamiento en el repositorio definido por el cliente.

5.3.3. Cliente Pocket PC

Este módulo es una aplicación desarrollada sobre plataforma móvil para realizar la extracción de información desde una Base de Datos que internamente posea el dispositivo.

Las funcionalidades de este módulo son básicamente las mismas que tiene el módulo del punto anterior, como son comunicación con el servidor para conseguir los datos de conexión a la Base de Datos móvil embebida en el dispositivo, extracción de los datos con las sentencias de consulta enviadas por el servidor y transmisión de la información hacia al servidor central para su procesamiento. Es importante indicar que este módulo requiere de conexión a internet en el dispositivo, sea ésta provista por una red inalámbrica dentro de una red LAN o WAN como la que

proporcionan los proveedores de telefonía celular GPRS, EDGE, 3G, HSPA+, etc.

5.3.4. Administración

El módulo administrativo es el principal que tiene el sistema, el cual permite el mantenimiento de las tablas generales que necesita el IDT para su funcionamiento.

Entre las tablas que permite administrar este módulo encontramos: Clientes, Distribuidores, Tipo de Clientes, Sistemas de Información, Tipo de Datos, Formato, etc.

Así mismo permite realizar la parametrización en cuanto a las conexiones, homologaciones y validaciones que se deben realizar a los datos transaccionales por cada tipo de cliente.

Las tablas de parámetros que posee el IDT, son las siguientes: Transacción, Homologación, Match, Campo Origen, Campo Destino, Información del Cliente, etc.

Finalmente, podemos indicar que este módulo permite administrar los usuarios, roles y opciones con la finalidad de restringir el acceso al IDT, creando los perfiles según el nivel de permisos que defina el cliente o el administrador de la plataforma.

5.3.5. Conexión

Para la comunicación entre los nodos y IDT utilizamos servicios web, los mismos que nos permiten el intercambio de información entre la aplicación principal IDT y el nodo del cliente, sin que este tenga algún tipo de restricción en lo referente al lenguaje en la que fue desarrollado o algún inconveniente de compatibilidad, ya que pueden interactuar mediante estos servicios aplicaciones desarrolladas en lenguajes de programación diferentes y lo más importante que pueden ser ejecutadas sobre cualquier plataforma, esto gracias a la adopción de estándares abiertos con las que trabajan estos servicios.

Adicional en el IDT se configuran los parámetros generales para acceder a cualquier origen de datos. Estos parámetros permitirán acceder a la información en las distintas plataformas que operen los diferentes distribuidores, dando flexibilidad total en cuanto a las herramientas que manejen estos.

5.3.6. Seguridad

El sistema corre sobre HTTPS, el mismo que utiliza un cifrado basado en SSL/TLS para crear un canal cifrado más

apropiado para el tráfico de información sensible que el protocolo HTTP. De este modo se consigue que la información sensible (usuario y claves de paso normalmente) no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de descifrar..

El módulo de Seguridad permite estructurar los permisos que se pueden asignar a los usuarios. Al momento de acceder al sistema, internamente lee una tabla donde se encuentran almacenados los permisos de todos los usuarios, que previamente fueron ingresados, y mediante la información obtenida de la misma presenta el menú al cual tiene acceso el usuario que está ingresando, por lo que únicamente se le presentaran las opciones en el menú a las que tiene acceso el usuario. Adicional a esto cada página tiene un control de verificación de sesión, con esto se procede a denegar cualquier intento de acceso directo a cualquier página si se intentara burlar la seguridad.

Con este esquema es posible redefinir los permisos a cada funcionalidad a un usuario determinado en cualquier

momento y se reflejara los nuevos accesos o restricciones en el sistema.

5.3.7. Conversión

El IDT posee el módulo de conversión, siendo este uno de los módulos pilares del sistema, donde intervienen las siguientes funcionalidades.

- Campo Destino.-Funcionalidad donde se puede realizar acciones sobre los formatos de salida de la información, los mismos que son campos con los cuales deben ser retornados por el Sistema de Información (SIG). En esta opción, se establece el tipo de dato además de la máscara, la cual será utilizada para la salida del archivo final.
- Campo Origen.-Funcionalidad donde se realizan acciones sobre los campos de origen que van a ser extraídos en cada distribuidor del cliente. Cabe indicar que cada campo origen tendrá un formato y pertenecerá a un distribuidor, cliente, transacción y si es el caso, consulta sql para el motor de base de datos correspondiente.
- Formato de salida.-Funcionalidad donde se puede realizar acciones sobre los campos de salida en el

archivo con formato del Sistema de Información SIG tales como tipo de dato, longitud y máscara de salida de cada campo a transformar.

- Extracción.-Funcionalidad donde se puede realizar la extracción y envío de la información de las transacciones parametrizadas, hasta el servidor web donde se realizará la conversión y envío de los datos hasta el cliente. Esta operación puede ser realizada de manera manual o automáticamente, lo cual es programado de lado del distribuidor.
- Conversión.-Funcionalidad principal en este módulo es la Conversión y Transformación de datos donde el administrador podrá realizar de manera manual o automática la conversión y transformación de los datos ya extraídos y transmitidos desde los distribuidores hasta el servidor, este proceso será en tipo batch donde se generarán todos los archivos formateados y serán ubicados en los servidores FTP de los clientes.

5.4. Pruebas realizadas

En este punto se describe las estrategias del plan de pruebas, realizadas sobre el Sistema Web demostrativo para este Proyecto de Graduación:

5.4.1. Pruebas de Unidad

Dentro de este tipo de pruebas, en las cuales se realiza el recorrido por cada módulo del sistema, por separado, para verificar que lo implementado se encuentre retornando los resultados esperados, debe ser considerado lo siguiente:

Deben ser probadas todas las posibilidades de ejecución dentro de las sentencias de control que han sido codificadas dentro de cada módulo.

Probar todos los escenarios dentro de los cuales podrían darse errores de ejecución dentro de cada módulo.

Como actividad complementaria, se debe hacer las pruebas que permitan detectar posibles errores a nivel de inicializaciones de variables, inconsistencias de tipos de datos, comparaciones dentro de las sentencias de control (bucles).

5.4.2. Pruebas de Integración

Se ha definido utilizar el método denominado integración descendente, el cual consiste en ir realizando las pruebas

desde el módulo principal del sistema, en donde se inicia la sesión e ir en secuencia incorporando los módulos que permiten ingresos, eliminaciones, modificaciones y consultas.

5.4.3. Pruebas de Validación

Lo que se va a conseguir con estas pruebas es comprobar que el sistema está cumpliendo con los requerimientos funcionales, seguridad informática, confiabilidad, facilidad de mantenimiento y documentación de código fuente.

Para estas validaciones se requiere de la participación del usuario final, pero para este proyecto de graduación se ha obviado la participación del mismo y hemos realizado nosotros mismos las pruebas.

La metodología utilizada para realizar esta tarea se denomina prueba – error, que consiste en depurar los errores encontrados en los diferentes tipos de pruebas, evaluar los resultados obtenidos versus los esperados y una vez corregidos volver a probar.

5.5. Análisis y resultados

Dentro de las pruebas de unidad se pudieron identificar los siguientes errores:

- En las anotaciones de los servicios web que permiten establecer el estilo del protocolo SOAP, el servicio de conversión tenía el estilo “document”, lo que provocaba un comportamiento impredecible del sistema. Este error fue corregido estableciendo el estilo “rpc”, que permite el manejo adecuado de tipo de datos complejos en el paso de mensajes al momento de consumir Servicios Web.
- Existieron errores de inicialización de variables, lo cual producía resultados no esperados del sistema.
- Se encontraron problemas con los tipos de datos de fechas y moneda, por las diferentes configuraciones regionales que poseen cada usuario y que definen el formato y separador de Fecha/Hora y Miles/decimales.
- Dentro de las pruebas modulares se hallaron errores en las sentencias de control, donde se presentó que al recorrer una lista de datos el índice salía del rango, debido a los límites definidos en el bucle.

5.6. Integración de los módulos

En las pruebas de integración, se realizó un recorrido por todos los módulos del sistema, evaluando todos los escenarios posibles que

podieron dar errores. Una vez que ya no fueron encontrados errores las pruebas fueron finalizadas.

Por último fue comprobado que el sistema cumplía con todos los requerimientos funcionales y de seguridad informática definidos para al desarrollo de este proyecto de graduación.

5.7. Instalación del sistema

- **Administrador:** Los elementos de instalación para el software de administración IDT, es necesario tener lo siguiente:
 - 1) JBOSS 4.2
 - 2) Java JDK 1.6
 - 3) SQL Server 2005

Una vez instalados estos componentes, se procede a colocar el .war en la carpeta deploy del servidor de aplicaciones; para luego iniciar el servicio.

La base de datos debe estar creada y lista para poder ser usada desde el sistema y proceder con las configuraciones necesarias para el buen desempeño del sistema y podamos obtener los resultados esperados.

- **Batch:** La aplicación batch es un archivo .jar, por lo tanto lo que se necesita es tener el Java JDK 1.6 para que pueda correr con normalidad.
- **Aplicaciones Cliente:** Son aplicaciones externas al IDT pero que a la vez estarán enlazadas por medio de las distintas configuraciones que se realicen en la pantalla administrativa. Para efecto de pruebas, hemos seleccionado como herramienta de desarrollo a Microsoft Visual Studio 2010, por lo tanto necesitamos tener la versión actualizada del framework 4.0. Con esto ya podríamos ejecutar las aplicaciones clientes, tanto móviles como de escritorio. Para las aplicaciones móviles hemos seleccionado como plataforma Windows Phone.

CAPÍTULO6

Definición del Producto

6.1. Costos del producto

Para obtener el costo total de la solución vamos a considerar tres componentes:

$$CTS = CTI + CTA + CTC$$

Dónde:

- CTI: Costo Total de Implementación
- CTA: Costo Total Administrativo
- CTC: Costo Total de Capacitación

Para el cálculo de costos CTA y CTC se considerará por lo menos 3 años de funcionamiento de la solución.

6.1.1. Costo Total de Implementación (CTI)

Es el costo total de rubros y actividades necesarios para poner a funcionar la solución. Se incluye adquisición de equipos, licencias y recurso humano puntual para la implementación. El CTI se calcula de la siguiente forma:

$$CTI = CD + CI + CADH + CADS$$

Dónde:

- CD: Costos de desarrollo del software
- CI: Costos de instalación, configuración y adaptación (si fuera el caso)
- CADH: Costos adicionales de hardware e infraestructura
- CADS: Costos adicionales de software

6.1.1.1. Costos de desarrollo del software (CD)

En este punto se considera las horas de trabajo dedicadas y las herramientas utilizadas en la misma.

$$CD = CDP + CHU$$

Dónde:

- CDP: Costos de desarrollo de Programador
- CHU: Costos de Herramientas Utilizadas

$$\text{Costo total de desarrollo } 1.596,00 + 12000 = 12596$$

6.1.1.1.1. Costos de desarrollo de Programador

Número de Programadores: 3

Costo Hora Programador: US\$ 3,50
(Considerando sueldo de US\$ 800 mensuales).

Detalle de las funciones del sistema con un estimado de horas necesarias para su desarrollo y el costo final en dólares.

Funcionalidad	Horas Desarrollo	Costo Final (US\$)
Administración de Usuarios	24	\$ 84,00
Administración de Clientes	24	\$ 84,00
Administración de Distribuidores	24	\$ 84,00
Administración de Sistemas de Información	24	\$ 84,00
Administración de Transacciones	40	\$ 140,00
Administración de Parámetros de Conexión y Orígenes de Datos	40	\$ 140,00
Administración de Formato de Salida de la Información	40	\$ 140,00
Administración de Campos del Origen de Datos	40	\$ 140,00
Administración de Campos de Salida según formato SIG	40	\$ 140,00
Extracción de datos del distribuidor	40	\$ 140,00
Conversión y Transformación	80	\$ 280,00
Revisiones y Pruebas	40	\$ 140,00
TOTALES	456	\$ 1.596,00

Tabla 48: Costos de Desarrollo

6.1.1.1.2. Costo de Herramientas Utilizadas (CHU)

Herramientas Utilizadas	Cantidad	Valor Unitario	Total
Servidor de Desarrollo	1	9000	9000
Computadores personales con licencias para desarrollo	3	1000	3000
		Total	12000

Tabla 49: Costos de Herramientas Utilizadas

6.1.1.1.3. Costos de instalación, configuración y adaptación (CI).

Para este punto se considera los valores correspondientes a los recursos responsables de la instalación, configuración y adaptación del sistema.

A continuación se presenta una tabla con el detalle de los recursos con los valores correspondientes a costos para cada uno de ellos y un aproximado en tiempo.

Recurso	Cantidad	Tiempo (horas)	Costo por Hora	Total
DBA	1	120	12,5	1500
Técnico	1	40	10	400

Tabla 50: Costos de Instalación, Configuración y Adaptación

6.1.1.2. Costos adicionales de hardware e infraestructura (CADH)

Los costos adicionales en hardware e infraestructura para el desarrollo del sistema fueron tomados en cuenta en el punto 5.1.1.1 (Costos de desarrollo de Software) en caso de la necesidad de hardware e infraestructura adicional, ejemplo dispositivos móviles en puntos de venta, este será un valor adicional que se agregara al total del costo del producto.

6.1.1.3. Costos adicionales de software (CADS)

Aquí no se considera ningún software que se utilizó en el desarrollo e implementación del sistema en IDT en la consola administrativa, el mismo que fue considerado en el punto 5.1.1.1, este rubro hace referencia a todo software en el nodo cliente que interactúe con la capa de servicios del IDT, como por ejemplo una aplicación para dispositivo móvil basado en sistema operativo Android, por lo que los costos de dichas aplicaciones dependerán de la plataforma y complejidad de la misma, valor que será adicional al costo final del producto.

6.1.2. Costo Total Administrativo (CTA)

Es el costo total promedio anual de rubros y actividades necesarios para garantizar la disponibilidad, capacidad y

continuidad de la solución implantada. Incluye el costo total promedio anual del recurso humano empleado en estas actividades. El CTA se calcula de la siguiente forma:

$$\text{CTA} = \text{CMH} + \text{CASS} + \text{CRH}$$

Dónde:

- CMH: Costos de actualización y mantenimiento del hardware e infraestructura.
- CASS: Costos de actualización y soporte del software.
- CRH: Costos del Recurso Humano.

6.1.2.1. Costos de actualización y mantenimiento del hardware e infraestructura (CMH)

6.1.2.2. Costos de actualización y soporte del software (CASS).

Los CASS representan los costos de actualización de licencias o nuevas versiones y el soporte del software. Este rubro se lo calcula de forma anual y se calcula un aproximado de US\$ 25000, como resultado por los 3 años calculamos US\$75000.

6.1.2.3. Costos del Recurso Humano

Los CRH se calculan de la siguiente forma:

$$\text{CRH} = \text{CA} + \text{CO} + \text{CS}$$

Dónde:

- CA: Costo de personal de la organización para administración de la solución con el fin de garantizar la disponibilidad y correcto funcionamiento. Este costo se estima de la siguiente forma:

CA = costo promedio anual de un ingeniero administrador * número de ingenieros * porcentaje de tiempo dedicado a la administración de la solución * número de años de funcionamiento de la solución

$$\mathbf{CA = 18,000 * 1 * 50\% * 3 = US\$ 27,000}$$

- CO: Costo de personal de la organización para operación de la solución con el fin de garantizar la continuidad de la misma. Este costo se estima de la siguiente forma:

CO = costo promedio anual de un ingeniero operador * número de ingenieros * porcentaje de tiempo dedicado a la operación de la solución * número de años de funcionamiento de la solución.

$$\mathbf{CO = 12000 * 2 * 100\% * 3 = US\$ 72,000}$$

- CS: Costo de personal de la organización para soporte en la solución de incidentes y problemas detectados. Este costo se calcula de la siguiente forma:

CS = costo promedio anual de un ingeniero de soporte * número de ingenieros * porcentaje de tiempo dedicado al soporte de la solución * número de años de funcionamiento de la solución.

$$\text{CS} = 9,600 * 2 * 50\% * 3 = \text{US\$ } 28,800$$

6.1.3. Costo Total de Capacitación (CTC)

Es el costo promedio anual para la capacitación continua del personal (técnico y usuarios) en la operación y explotación de la solución. El CTC se calcula de la siguiente forma:

$$\text{CTC} = \text{CT} + \text{CU}$$

Dónde:

- CT = Costo hora capacitación técnica * número de técnicos * número de horas
- CT = 30 * 1 * 20 = US\$ 600
- CU = Costo hora capacitación usuario * número de usuarios * número de horas
- CU = 20 * 1 * 50 = US\$ 1,000

6.2. Licenciamiento

Para licenciar nuestro sistema debemos completar el formulario correspondiente, para programas de ordenador, que se encuentra publicado en el Portal Web del IEPI (Formulario para el registro de programas de ordenador), www.iepi.gob.ec.

El formulario lleva las siguientes secciones, las mismas que deberán ser llenados (Anexo 1):

1. Número de solicitud y fecha, van en blanco;
2. Datos del autor: Al ser nuestro caso un sistema desarrollado por 3 personas, llenamos las opciones a), b) y c)
3. Datos de la obra: se indica el título de manera precisa, y otros datos referente a la obra, los mismos que tendrán que ser seleccionados de varias opciones que se presentan en el formulario.
4. Datos del titular: Se indica quien ostenta los derechos patrimoniales sobre la obra, en este caso la Escuela Politécnica del Litoral
5. Datos del solicitante y firma
6. La firma del abogado en la misma no es obligatorio pero es recomendable.

Como requisitos adjuntos, se necesita:

- Un ejemplar del programa de ordenador;

- Copia de la cédula, pasaporte o cualquier documento de identidad de los autores.
- Para el titular es necesario adjuntar una copia simple del documento de creación.

6.3. Segmento del Mercado

A continuación se describe algunos de los clientes al que está enfocado el Sistema de Integración Transaccional Distribuido (IDT):

HIVIMAR: Es una empresa proveedora de soluciones integrales en el área de componentes industriales y automotrices. Posee una distribución de los productos a través de su oficina matriz en Guayaquil, Quito y Cuenca, cuenta con 4000 subdistribuidores cubriendo en el mercado Nacional.

Las solicitudes de distribución se realizan a través de televentas (pedidos telefónicos), quienes atienden las necesidades de los clientes recogiendo sus pedidos en todo el país.

PAPELESA: es una empresa especializada en la fabricación de productos a base de papel y en la actualidad es la industria más grande del país por su infraestructura y recurso humano.

La distribución de su producto se realiza a través de los agentes vendedores, quienes atienden las necesidades de los clientes recogiendo sus pedidos en todo el país.

COLGATE: es una empresa líder en productos de consumo de limpieza personal, reconocida por la calidad de su producto.

La distribución del producto se realiza a través de un canal indirecto, por esta razón el producto se vende en establecimientos minoristas como mayorista.

Este sistema de distribución es el adecuado para este tipo de productos de consumo.

El Sistema de Integración Transaccional Distribuido (IDT), está dirigido a este tipo de empresas transnacionales que han adquirido SAP en el país, y que cuentan con la necesidad de obtener la información actualizada un depósito local con formato requerido para ser cargado a SAP desde sus diferentes puntos de venta y terminales de servicio de todos sus distribuidores y que pueda estar disponible en cualquier momento.

6.4. Matriz FODA del producto

ASPECTOS ANALIZADOS	FACTORES INTERNOS		FACTORES EXTERNOS	
	FORTALEZAS	DEBILIDADES	OPORTUNIDADES	AMENAZAS
IDT	<ul style="list-style-type: none"> Mantiene información actualizada confiable y consistente cargada en un repositorio único Facilidad para la extracción, transmisión, conversión y ubicación de las transacciones de los puntos remotos de los clientes. Disminuye la interacción humana evitando los errores por la manipulación de datos. Total parametrización de datos de origen y de destino. 	<ul style="list-style-type: none"> Incertidumbre de cómo van a recibir nuestros o posibles mercados potenciales nuestro sistema Falta de cobertura de los canales de comunicación. Posible complejidad en el uso del servicio para niveles socioeconómicos de bajos recursos 	<ul style="list-style-type: none"> Servir a grupos de clientes adicionales o abrirse hacia nuevos mercados geográficos o segmentos del producto. Alianzas o empresas conjuntas que amplíen la cobertura del mercado. Oportunidades de mercado para ampliar la marca registrada del sistema a nuevas áreas geográficas. 	<ul style="list-style-type: none"> Competidores potenciales. Perdida de ventas debido a productos sustitutos.

Tabla 51: Matriz FODA

6.5. Servicios del producto

A continuación presentamos los servicios que ofrece el IDT para satisfacción del cliente una vez entregado y/o implementado el software.

Soporte técnico: Este soporte comprende una personalización o configuración adicional en el sistema que es requerida por el cliente.

Aplicación de garantía: Servicio ofrecido para para tranquilidad del cliente, en el que existe el compromiso de solucionar de manera urgente todo error de funcionamiento comprobado en el sistema mediante los ingenieros de soportes.

Cursos de capacitación: Al momento de la entrega del sistema, se ofrecerá obligatoriamente dos tipos de capacitaciones, técnica y a usuarios, sobre el funcionamiento del mismo a las personas que estarán directamente involucradas en el producto. Se incluyen 20 horas de capacitación técnica y 50 horas de capacitaciones a usuarios, las mismas que podrán ser tomadas a conveniencia del cliente en el transcurso del primer año.

Manejo de quejas: Comprende básicamente en que si el personal de soporte del IDT no satisface la necesidad del cliente, el mismo

tendrá la total libertad de levantar formalmente un reclamo y el personal administrativo del IDT deberá implantar las correcciones necesarias.

CONCLUSIONES

1. Sistema escalable que soporta la exigencia y eficiencia en la integración de las distintas aplicaciones entre los clientes de un servicio.
2. Provee independencia entre las aplicaciones clientes.
3. Alto grado de interoperabilidad entre los sistemas clientes y los servicios publicados para su consumo.
4. Con este esquema se asegura la disponibilidad de los Servicios Web para cuando cualquier aplicación lo requiera.
5. Analizando las bases de datos existentes y el mercado hacia el cual fue enfocado el desarrollo del sistema (SIG) se seleccionó SQL Server como herramienta a utilizar con el fin de tener una mayor integración.

RECOMENDACIONES

1. Se recomienda que los equipos en los puntos de venta tengan una buena plataforma de hardware para tener una operación eficiente de manera independiente.
2. El canal de transmisión de datos (enlace) debe ser lo suficientemente robusto para lograr una comunicación de alto rendimiento al momento de sincronizar los movimientos o transacciones con el repositorio central de datos o con el sistema central (IDT).

ANEXOS

A.1 Registro de Programa de Ordenador de Software

Es necesario realizar el registro del Software en el Instituto Ecuatoriano de la Propiedad Intelectual – IEPI – Dirección Nacional de Derecho de Autor y Derechos Conexos, a continuación se presenta el formulario que se debe llenar para completar este proceso.

REGISTRO DE PROGRAMAS DE ORDENADOR (SOFTWARE)



**INSTITUTO ECUATORIANO DE LA PROPIEDAD
INTELECTUAL -IEPI-
DIRECCIÓN NACIONAL DE DERECHO
DE AUTOR Y DERECHOS CONEXOS**

Número de solicitud: *(Para uso del IEPI)*

Fecha de presentación: *(Para uso del IEPI)*

DATOS DEL AUTOR O AUTORES

a Nombre: _____ Nacionalidad: _____
 Domicilio: _____ Ciudad: _____ País: _____
 Fecha de nacimiento: _____ Fecha de defunción: _____ Seudónimo: _____

b Nombre: _____ Nacionalidad: _____
 Domicilio: _____ Ciudad: _____ País: _____
 Fecha de nacimiento: _____ Fecha de defunción: _____ Seudónimo: _____

c Nombre: _____ Nacionalidad: _____
 Domicilio: _____ Ciudad: _____ País: _____
 Fecha de nacimiento: _____ Fecha de defunción: _____ Seudónimo: _____

DATOS DE LA OBRA

Título: _____

Fecha publicación: _____ Individual Seudónima Original Inédita
 En colaboración Póstuma Derivada Publicada
 País de origen: _____ Colectiva Por encargo
 Anónima

a Otras _____
 Breve descripción de funciones del programa: _____

TITULAR DE LA OBRA

Nombre: _____
 Domicilio: _____ Ciudad: _____ País: _____

ELEMENTOS DEL PROGRAMA DE ORDENADOR ADJUNTADOS

a. Programa de computador b. Descripción de programa c. Material auxiliar d. Otros
 Observaciones: _____

DATOS DEL PRODUCTOR

Nombre: _____
 Domicilio: _____ Ciudad: _____ País: _____

DATOS DEL SOLICITANTE

Nombre: _____
 Domicilio: _____ Ciudad: _____ Teléfono: _____
 En representación de: _____ Domicilio: _____ País: _____

 Firma de Abogado Patrocinador

 Firma solicitante

INSTRUCTIVO

Llenar la solicitud a máquina o en computadora.

DATOS DEL AUTOR O AUTORES

1. Indicar los nombres y más datos del autor o autores del programa de ordenador (software), en caso de ser aplicable. Si hay más de tres autores, adjuntar los nombres en una hoja adicional.
2. Si el programa de ordenador a registrar es póstumo, deberá mencionárselo y en este caso el registro podrá hacerse a nombre del autor o autores o de los herederos reconocidos por la ley.

DATOS DE LA OBRA

1. Indicar de manera exacta y completa el título identificativo del programa de ordenador a registrar (software).
2. Mencionar la fecha en que por primera vez la obra ha sido accesible al público en forma masiva (fecha de publicación).
3. Indicar el país donde se realizó la primera publicación (país origen).
4. Señalar con una X si el programa de ordenador (software) es individual, en colaboración, colectivo, anónimo, seudónimo, póstumo, por encargo, o señalar lo que corresponda.
5. Indicar si la obra es original, o derivada, es decir que la obra es el resultado de una transformación de una obra originaria, siempre que constituya una creación autónoma.
6. Indicar si la obra es inédita o si ya ha sido puesta al alcance del público en forma masiva (publicada).
7. Describir brevemente las funciones y características básicas y fundamentales del programa de ordenador (software).

DATOS DEL TITULAR DE LA OBRA

1. Indicar los datos del productor, esto es la persona natural o jurídica que toma la iniciativa y la responsabilidad de realizar la obra.

ELEMENTOS DEL PROGRAMA DE ORDENADOR ADJUNTADOS

1. Señalar con una X los elementos del programa de ordenador adjuntados al registro como: a) Programa de computador, b) Descripción de programa y c) Material auxiliar (manual para el usuario) y d) Otros.
2. Observaciones: Especificar la identificación del material adjunto.

DATOS DEL PRODUCTOR

Indicar los datos correspondientes de la persona natural o jurídica bajo la cual se fija, distribuye y comercializa el programa de ordenador (software).

DATOS DEL SOLICITANTE

1. Indicar el nombre y más datos de la persona que hace la solicitud, conjuntamente con la firma del Abogado Patrocinador, que es recomendable, pero obligatoria.
2. Los datos correspondientes del representado si es el caso. De ser persona jurídica, deberá acompañarse además el documento que legitime su representación.

REQUISITOS ADICIONALES

1. Entregar un ejemplar de la obra.
2. Adjuntar el recibo de pago de la tasa correspondiente.
3. Anexar la fotocopia de la cédula de ciudadanía del autor y/o autores.
4. El patrocinio del abogado es recomendado pero no obligatorio.
5. De contar la obra con más datos de los contenidos en el formulario, hágalos constar en una hoja adicional.

NOTA: La atención al público es ininterrumpida de lunes a viernes. El valor de la tasa es de \$ 20, a ser depositados en el Banco de Guayaquil. Cta.Cte N° 6265391, Código: 130199, a nombre del IEPI.

BIBLIOGRAFÍA

- [1] Álvarez, M. A. (21 de 06 de 2001). *Objetivos y usos del XML*. Recuperado el 27 de 07 de 2014, de *Objetivos y usos del XML*:
<http://www.desarrolloweb.com/articulos/460.php>
- [2] Barreno, B. (02 de 04 de 2010). *ESPOCH Escuela de Ingeniería en Sistemas*. Obtenido de *ESPOCH Escuela de Ingeniería en Sistemas*:
<http://es.slideshare.net/babp/protocolo-https-3617275>
- [3] Besteiro, M., & Rodriguez, M. (s.f.). *Web Services*. Obtenido de *Web Services*:
<http://www.ehu.es/mrodriguez/archivos/csharp/pdf/ServiciosWeb/WebServices.pdf>
- [4] DEITEL, P. J., & DEITEL, H. M. (2008). *CÓMO PROGRAMAR EN JAVA. Séptima edición*. México: PEARS ON EDUCACIÓN.
- [5] Eguiluz, J. (2007). *Introducción a AJAX*.
- [6] Eguiluz, J. *Introducción a CSS*. Creative Commons.
- [7] Eguiluz, J. *Introducción a JavaScript*.
- [8] Eguiluz, J. *Introducción a XHTML*. Creative Commons.
- [9] Eivind. (01 de 03 de 2008). *JBoss guide: How to enable SSL (HTTPS) on JBoss, as well as other "nice-to-know" configurations*. Obtenido de *JBoss guide: How to enable SSL (HTTPS) on JBoss, as well as other "nice-to-know" configurations*:
<http://roneiv.wordpress.com/2008/01/03/jboss-tutorial-how-to-enable-ssl-https-on-jboss-as-well-as-other-nice-to-know-configurations/>
- [10] Elissalde, J. C. (2014). *Wikipedia: Middleware*. Obtenido de *Wikipedia: Middleware*:
<http://es.wikipedia.org/wiki/Middleware>
- [11] Galante, J. C. (05 de 01 de 2009). *Web Services Protocol Stack*. Recuperado el 25 de 07 de 2014, de *Web Services Protocol Stack*:
<http://www.juancarlosfernandez.net/2009/01/web-services-protocol-stack.html>
- [12] Garcia, A. S. (15 de 06 de 2013). *Taringa*. Recuperado el 15 de 02 de 2014, de *Taringa*: <http://www.taringa.net/posts/info/16836103/Listado-de-algunos-sistemas-operativos-existentes.html>

- [13] Giardina, F. (2011). *Guía de ASP.NET: Desarrollo de sitios y aplicaciones web dinámicas*.
- [14] Herrera, W. (13 de 04 de 2011). *Qué es el protocolo HTTPS, cómo funciona y para que sirve?* . Obtenido de Qué es el protocolo HTTPS, cómo funciona y para que sirve? : <http://webadictos.com/2011/04/13/que-es-el-protocolo-https-y-como-funciona/>
- [15] IBM. (05 de 08 de 2011). *Comprender las especificaciones de los servicios web*. Recuperado el 25 de 07 de 2014, de Comprender las especificaciones de los servicios web:
<http://www.ibm.com/developerworks/ssa/webservices/tutorials/ws-understand-web-services1>
- [16] James , J. (2005). Ajax: A New Approach to Web Applications. En J. J. Garrett.
- [17] JARA, O. H. (29 de 10 de 2004). *Monografías: Sistemas distribuidos*. Obtenido de Monografías: Sistemas distribuidos:
<http://www.monografias.com/trabajos16/sistemas-distribuidos/sistemas-distribuidos.shtml>
- [18] Java - Oracle. (01 de 01 de 2014). *The Java Tutorials*. Recuperado el 2014, de The Java Tutorials:
<http://docs.oracle.com/javase/tutorial/jdbc/basics/gettingstarted.html#step1>
- [19] Microsoft. (2014). *Microsoft SQL Server*. Obtenido de Microsoft SQL Server:
http://www.microsoft.com/OEM/es/products/servers/Pages/sql_server.aspx#fbid=G_saYtSoVtO
- [20] Montero, C. (17 de 09 de 2013). *Wikipedia*. Recuperado el 20 de 03 de 2014, de Enhanced Data Rates for GSM Evolution:
http://es.wikipedia.org/wiki/Enhanced_Data_Rates_for_GSM_Evolution
- [21] My Eclipse. (2014). *My Eclipse*. Obtenido de My Eclipse:
<http://www.myeclipseide.com/module-htm/pages-display-pid-7.html>
- [22] Nakamura, M., Narvaez, D., & Ramos, A. (22 de 11 de 2009). Recuperado el 20 de 03 de 2014, de IP Backhaul para redes móviles:
<http://blog.pucp.edu.pe/item/79314/ip-backhaul-para-redes-moviles>
- [23] Polanco, S. L. (20 de 07 de 2013). *Wikipedia*. Recuperado el 15 de 03 de 2014, de Servicio general de paquetes vía radio. Definición. Tecnología utilizada.:
http://es.wikipedia.org/wiki/Servicio_general_de_paquetes_v%C3%ADa_radio

- [24] roneiv. (03 de 01 de 2008). *JBoss guide: How to enable SSL (HTTPS) on JBoss, as well as other "nice-to-know" configurations*. Recuperado el 28 de 07 de 2014, de JBoss guide: How to enable SSL (HTTPS) on JBoss, as well as other "nice-to-know" configurations: <http://roneiv.wordpress.com/2008/01/03/jboss-tutorial-how-to-enable-ssl-https-on-jboss-as-well-as-other-nice-to-know-configurations/>
- [25] Sencha. (2014). *Extjs-tutorial*. Obtenido de Extjs-tutorial: <http://www.extjs-tutorial.com/extjs/Introduction>
- [26] Vázquez, P. (2006.). CREACIÓN DE SITIOS WEB. En P. Vázquez, *CREACIÓN DE SITIOS WEB* (págs. 30 -43). Banfield : Manuales USERS.
- [27] Vázquez, P. (2006). HTML. En P. Vázquez, *CREACIÓN DE SITIOS WEB* (págs. 33-36). Manuales USERS.
- [28] Visual Studio. (2014). *Visual Studio*. Obtenido de Visual Studio: <http://www.visualstudio.com/>
- [29] W3C. (28 de 08 de 2007). *Semantic Annotations for WSDL and XML Schema*. Recuperado el 27 de 07 de 2014, de Semantic Annotations for WSDL and XML Schema: <http://www.w3.org/TR/sawsdl/>
- [30] W3C. (27 de 03 de 1999). *XML en 10 puntos*. Recuperado el 25 de 07 de 2014, de XML en 10 puntos: <http://www.w3.org/XML/1999/XML-in-10-points.es.html>
- [31] W3C. (04 de 02 de 2004). *XML Information Set*. Recuperado el 25 de 07 de 2014, de XML Information Set: [http://www.w3.org/TR/2004/REC-xml-infoset-20040204/\(04-02-2004\)](http://www.w3.org/TR/2004/REC-xml-infoset-20040204/(04-02-2004))